Recursive Markov Decision Processes and Recursive Stochastic Games

Kousha Etessami, University of Edinburgh
Mihalis Yannakakis, Columbia University

We introduce Recursive Markov Decision Processes (RMDPs) and Recursive Simple Stochastic Games (RSSGs), which are classes of (finitely presented) countable-state MDPs and zero-sum turn-based (perfect information) stochastic games. They extend standard finite-state MDPs and stochastic games with a recursion feature. We study the decidability and computational complexity of these games under termination objectives for the two players: one player's goal is to maximize the probability of termination at a given exit, while the other player's goal is to minimize this probability. In the *quantitative termination problems*, given an RMDP (or RSSG) and probability p, we wish to decide whether the value of such a termination game is at least p (or at most p); in the *qualitative termination problem* we wish to decide whether the value is 1. The important 1-exit subclasses of these models, 1-RMDPs and 1-RSSGs, correspond in a precise sense to controlled and game versions of classic stochastic models, including multi-type Branching Processes and Stochastic Context-Free Grammars, where the objective of the players is to maximize or minimize the probability of termination (extinction).

We provide a number of upper and lower bounds for qualitative and quantitative termination problems for RMDPs and RSSGs. We show both problems are undecidable for multi-exit RMDPs, but are decidable for 1-RMDPs and 1-RSSGs. Specifically, the quantitative termination problem is decidable in PSPACE for both 1-RMDPs and 1-RSSGs, and is at least as hard as the square root sum problem, a well-known open problem in numerical computation. We show that the qualitative termination problem for 1-RMDPs (i.e. a controlled version of branching processes) can be solved in polynomial time both for maximizing and minimizing 1-RMDPs. The qualitative problem for 1-RSSGs is in NP $\cap$ coNP, and is at least as hard as the quantitative termination problem for Condon's finite-state simple stochastic games, whose complexity remains a well known open problem. Finally, we show that even for 1-RMDPs, more general (qualitative and quantitative) model checking problems with respect to linear-time temporal properties are undecidable even for a fixed property.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumeric Algorithms and Problems; G.3 [**Probability and Statistics**]: Markov Processes

General Terms: Algorithms, Theory, Verification

Additional Key Words and Phrases: Recursive stochastic processes, Markov decision processes, stochastic games, multi-type branching processes, stochastic context-free grammars

1. INTRODUCTION

Markov Decision Processes (MDPs) are a fundamental model for stochastic dynamic optimization, with widespread applications in many fields (see, e.g., [Puterman 1994]). Stochastic games generalize MDPs with multiple players and are a basic model in

Preliminary versions of the material in this paper have appeared in two separate conference papers in 2005 and 2006: [Etessami and Yannakakis 2005; 2006]. This work is partially supported by the Royal Society, and by NSF Grants CCF-10-17955 and CCF-1320654. Authors' addresses: Kousha Etessami, LFCS, School of Informatics, University of Edinburgh, kousha@inf.ed.ac.uk; Mihalis Yannakakis, Department of Computer Science, Columbia University, mihalis@cs.columbia.edu

DOI:http://dx.doi.org/10.1145/0000000.0000000

game theory (see, e.g., [Filar and Vrieze 1997; Neyman and Sorin 2003]). In a stochastic game, each player gets to choose an action at a state, and their joint choice determines a probability distribution on the next state. Fixing an initial state, and fixing a strategy for every player, determines a probability space of runs (trajectories) of the stochastic game. In a 2-player zero-sum stochastic game, there is some objective function (which can in general be any (measurable) function of the entire trajectory) and one player's goal is to maximize (the expected value of) this objective while the other aims to minimize it. Many such objectives have been studied in the literature for many different applications. Markov decision processes are simply the 1-player (1 controller) version of such games.

In general, the state space of an MDP or a stochastic game can be finite or infinite. The computational study of MDPs and games, and analysis of their computational complexity, has been largely restricted to the finite state case. Of course, it is not possible to computationally treat arbitrary infinite-state games, because we can not handle infinite-sized objects as input. This is possible only if we have a finite representation, that is, each finite instance represents an underlying infinite-state system.

In this paper we introduce and study a class of finitely presentable, countable state, MDPs and 2-player, zero-sum, *turn-based* (perfect information) stochastic games, which arise naturally as models of probabilistic procedural programs, and which also relate to classic infinite-state automata theoretic models and stochastic models, and we study the decidability and computational complexity of basic problems related to computing the value and optimal strategies in such games. The MDPs and games that we study can be seen on the one hand as extensions of usual finite-state MDPs and games with a recursive feature. Alternatively, they can also been seen as extensions of recursive stochastic models (e.g., recursive Markov chains and branching processes) with control and game features.

The games we study are natural generalizations of finite-state Simple Stochastic Games (SSGs), introduced by Condon [1992]. These are turn-based (perfect information) stochastic games, meaning at each state only one player gets to choose an action, and the goal of one player is to maximize the probability of reaching a given terminal state, while the other player's aim is to minimize this probability. From a computational point of view, finite-state SSGs generalize several other well studied games in computer science and other fields, such as parity games ([Emerson and Jutla 1991]) and mean payoff games ([Ehrenfeucht and Mycielski 1979; Zwick and Paterson 1996]), in the sense that there are polynomial time reductions from the main decision problems associated with these other games to the quantitative termination decision problem associated with SSGs. The quantitative termination problem for SSGs (i.e., is the game value $\geq 1/2$?), already presents a major challenge: it is in NP $\cap$ coNP, but it is a well-known open problem whether there is a polynomial time algorithm for it (see [Condon 1992]). (The same open complexity status holds also for the mentioned decision problems for the "simpler" parity and mean payoff games.)

The MDPs and games we study can be viewed also as controlled and game extensions of Recursive Markov Chains, in the same way that ordinary finite-state MDPs and games generalize finite-state Markov chains. Recursive Markov Chains (RMCs), a class of finitely presented countable state Markov chains, were introduced and studied in ([Etessami and Yannakakis 2009]) as a natural model of probabilistic procedural programs and other systems that exhibit both recursive and probabilistic behavior. Informally, a RMC consists of a (finite) collection of finite state Markov chains that can call each other in a potentially recursive manner like recursive procedures. A recursive call to a component Markov chain initiates its execution at one of a set of designated entry nodes; if and when the called component reaches one of a set of designated exit nodes, the recursive call terminates and the Markov chain that initiated the call

resumes execution. RMCs define a class of denumerable Markov chains with a rich theory. Their computational problems subsume in a precise sense central questions for a number of other classic stochastic models including multi-type Branching Processes (BPs) (see, e.g., [Harris 1963]), Stochastic Context-Free Grammars (SCFGs) (see, e.g., [Manning and Schütze 1999]), and (discrete-time) Quasi-Birth Death processes (see [Neuts 1981; Latouche and Ramaswami 1999]), as well as more recently studied models like Backbutton Processes ([Fagin et al. 2000]). Both SCFGs and BPs correspond in a precise sense to 1-exit RMCs (1-RMCs): RMCs in which each component Markov chain has 1 terminating exit state where it can return control back to the location (in some component) that it was called from. General multi-exit RMCs are equivalent in a precise sense to probabilistic Pushdown Systems (pPDSs). Pushdown automata are a classic automata theoretic model, and their probabilistic versions, pPDSs, have also been studied recently in connection to verification of probabilistic programs ([Esparza et al. 2004; Brázdil et al. 2005]). See [Etessami and Yannakakis 2009] where these relationships are detailed.

In the context of probabilistic program verification, it is quite natural and useful to extend RMCs with nondeterminism (i.e., a controller), where some states are nondeterministically controlled by the system or environment while others exhibit probabilistic behavior. Indeed, finite-state MDPs have been studied extensively for verification of probabilistic systems, and verification tools already exist for them (see, e.g., [Courcoubetis and Yannakakis 1998; 1995; Vardi 1985; Hart et al. 1983; Baier and Kwiatkowska 1998; de Alfaro et al. 2000; Kwiatkowska 2003]). Simple stochastic games extend MDPs further with a second (adversarial) player. They can also be used to model and analyze the interactions between a system with both probabilistic and non-deterministic/controlled behavior, and its (adversarial) environment.

In this paper we study such optimization and game extensions of RMCs: we introduce *Recursive Markov Decision Processes* (RMDPs) and *Recursive Simple Stochastic Games* (RSSGs), which define natural classes of countable-state MDPs and SSGs, respectively. We should emphasize that these games are very different from, and should not be confused with, Everett's classic notion of *recursive games* [Everett 1957]. Everett's games are in fact a particular class of finite-state (imperfect information) zero-sum stochastic games where distinct rewards are accumulated only at specified terminal nodes at which the game halts. This clash in names with Everett's games is unfortunate, but the use of the word *recursive* is appropriate for our setting of RMDPs and RSSGs, where recursion is used in the usual computer science sense as in recursive programs.

Example 1.1.

An example of a Recursive Simple Stochastic Game (RSSG) is depicted in Figure 1.

In this example, there are two *components*, A_1 and A_2. In each component, there are three kinds of *nodes*, which are denoted by diamonds, triangles, and black dots, respectively. The nodes denoted by a diamond (namely, nodes en and w) are controlled by Player 1 (the maximizer), whereas nodes denoted by a triangle (namely, node en' and node z) are controlled by Player 2 (the minimizer). The remaining nodes, which are denoted by black dots, are probabilistic nodes.

Each component of a RSSG has *entry* nodes (where it can start execution) and *exit* nodes (where it terminates). In this RSSG, both components have only one entry node: in A_1 the only entry node is en, and in A_2 the only entry node is en'. Furthermore, in this RSSG both components have two exit nodes. In component A_1 the two exit nodes are ex_1 and ex_2. In component A_2 the two exit nodes are ex'_1 and ex'_2. In general, when we depict a RSSG, the entry nodes of each component are depicted on the left side of each component, and the exit nodes are depicted on the right side.

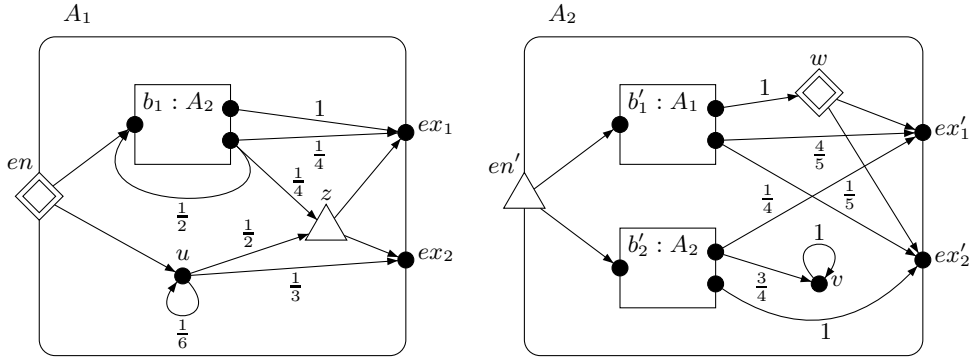

Fig. 1. *A sample Recursive Simple Stochastic Game*

In addition to nodes, the two components A_1 and A_2 both contain *boxes*, which represent "subroutine calls" to other components. In particular, A_1 contains one box, called b_1, which is "mapped" to A_2, meaning it represents a subroutine call to A_2. A_2 contains two boxes: b'_1 which is mapped to A_1, and b'_2 which is mapped to A_2.

Associated with each box in each component, there are *call port* nodes and *return port* nodes (in the example, these are denoted by black dots on the left and right side of each box). The call ports of each box are in 1-to-1 correspondence with the entries of the component the box is mapped to. Since in this example each component has only one entry, this means the unique call port of each box corresponds to the unique entry of the component the box maps to. For instance, the unique call port of box b'_1, which we can name using the pair (b'_1, en), corresponds to (maps to) the entry en of component A_1. Likewise, the return ports of each box are in 1-to-1 correspondence with the set of exits of the component that the box is mapped to. So, for instance, the two return ports of box b_1, which can be named by the pairs (b_1, ex'_1) and (b_1, ex'_2), correspond to the exits ex'_1 and ex'_2 of component A_2, respectively. (Note that distinct return ports of a box capture at an abstracted level the distinct possible values that can be returned by the procedure that is called by the box. Likewise, distinct call ports can be used to capture distinct possible parameter values that can be passed to the procedure.)

Intuitively, this finitely-specified RSSG corresponds to an infinite-state stochastic game, in the following way: imagine that we repeatedly replace all the boxes in both of the two components by a copy of the component to which each given box is mapped to, by appropriately attaching (or identifying) the call ports and return ports of each box to the corresponding entries and exits of the corresponding copy we have made of the component. This however is clearly not a terminating procedure, because in the presence of unbounded recursion there will always remain more boxes that we have to "expand". However, in the limit this procedure creates, in place of each component, an infinite-state transition graph for a simple stochastic game.

Furthermore, if we also specify a start node in some component, for example the entry node en in component A_1, and if we additionally specify a target exit node in the same component, for example ex_1, then the RSSG together with this information fully defines an infinite-state simple stochastic game, where the goal of Player 1 (maximizer) is to maximize the probability of eventually *terminating* (exiting) at ex_1, starting at en, whereas the goal of Player 2 (minimizer) is to minimize this probability.

To be more precise, the *states* of the infinite-state simple stochastic game are given by pairs of the form $\langle \beta, v \rangle$, where v is a node and $\beta = \beta_1 \dots \beta_m$ is a possibly empty string (sequence) of boxes, which denote the *call stack*. A play of the game begins in node en with an *empty* call stack, i.e., in state $\langle \epsilon, en \rangle$, where ϵ is the empty string, and the aim of

Player 1 (maximizer) is to maximize the probability that the game eventually reaches the exit node ex_1 with an empty call stack, i.e., reaches the state $\langle \epsilon, ex_1 \rangle$; whereas the aim of Player 2 (minimizer) is to minimize this probability.

It follows from very general results about stochastic games that such RSSG termination games are *determined*, meaning they have a *value*. The computational goal is to compute (or approximate) this value. □

Of course, if a given RSSG happens to only contain probabilistic nodes, then it is a Recursive Markov Chain (RMC) and it defines an infinite-state Markov chain in the same way. Likewise, it a given RSSG only contains probabilistic nodes and nodes belonging to Player 1 (or, only to Player 2, respectively), then it is a Recursive Markov Decision Process (RMDP),and it defines a infinite-state Markov decision process (MDP), where the goal of the single player (the controller) is to maximize (respectively, minimize) the specified termination probability.

In the stochastic dynamic programming literature MDPs are studied under many different reward criteria, such as average reward, discounted reward, etc. Our original motivations come from verification of probabilistic systems, and thus we study RMDPs and RSSGs under termination criteria which are central to all of the more general temporal analyses one might wish to perform on such models, such as model checking against regular properties ([Courcoubetis and Yannakakis 1998]). Specifically, we consider RMDPs where the objective of the controller is to maximize or to minimize the probability of termination at a given exit, and we consider RSSGs which are the natural two-player zero-sum extension of this, i.e., where player 1's goal is to maximize this probability and player 2's goal is to minimize it.

The central computational questions we study in this paper are:
(1) *Quantitative termination problems:* Given an RMDP (or RSSG) A and a probability p, is the associated termination game value $\geq p$ (or $\leq p$), or approximate the game value to within a desired error $\epsilon > 0$;
(2) *Qualitative termination problems*: Is the game value exactly 1?

We show that for general multi-exit RMDPs and RSSGs, these questions are all undecidable. Our positive results apply to 1-exit RMDPs (abbreviated as 1-RMDP) and 1-exit RSSGs (1-RSSGs), where each component is restricted to have only one exit. 1-RMDPs and 1-RSSGs correspond to controlled and game extensions, respectively, of both BPs and SCFGs. Branching processes (BPs) are of course 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), and continuing in the 20th century in the work of Kolmogorov and Sevastyanov [1947], Harris and others for multi-type BPs and beyond (see, e.g., [Harris 1963; Athreya and Ney 1972; Jagers 1975; Kimmel and Axelrod 2002; Haccou et al. 2005]). BPs have been used in a wide variety of applications, including in population genetics ([Haccou et al. 2005]), nuclear chain reactions ([Everett and Ulam 1948]), and biology [Jagers 1975; Kimmel and Axelrod 2002]. SCFGs are fundamental models in statistical natural language processing (see, e.g., [Manning and Schütze 1999]), and are used also in computational biology (for example for RNA modeling ([Sakakibara et al. 1994]).

The termination problems for 1-RMDPs (and 1-RSSGs) capture precisely the *extinction problems* for a controlled (respectively, 2-player game) version of BPs, which we shall call (multi-type) *Branching Markov Decision Processes*, abbreviated as *BMDPs* (and resp., (multi-type) *Branching Simple Stochastic Games*, abbreviated as BSSGs). Such extinction problems for BMDPs and BSSGs are also equivalent to termination problems for controlled/game versions of SCFGs. In more detail, in a BMDP (BSSG) there are a finite number of distinct types. The reproduction dynamics of some types is controlled by a player, whose actions affect the probability distribution of the off-

springs of the type in the next generation (in the case of BSSGs, each such controlled type is controlled by a specific one of the two players). The reproduction of other types is probabilistic, i.e., governed by a (finite support) probability distribution over multi-sets of types, as in ordinary BPs. We are interested in what is the maximum or minimum probability (respectively, game value) of eventual extinction, starting from a given single type, or more generally, starting from a given initial population (the goal of maximizing or minimizing this probability yields distinct problems for BMDPs). It is important to point out that the termination and extinction probability even for purely probabilistic 1-RMCs and BPs can be irrational, and thus we can not compute it exactly, but rather we can ask various (qualitative or quantitative) decision questions about it, or approximate it. The same applies to the extinction values in the optimization and game setting of 1-RMDPs and 1-RSSGs (i.e., BMDPs and BSSGs).

In [Etessami and Yannakakis 2009] it was shown that extinction problems for BPs can be viewed as termination problems for 1-RMCs, and vice versa, so that the two problems are polynomial time equivalent. We show that the extinction problems for BMDPs (BSSGs) can be viewed as termination problems for 1-RMDPs (1-RSSGs, respectively). Thus, our results on 1-RMDPs and 1-RSSGs apply equally to BMDPs and BSSGs.

The BMDP model is well suited for analyzing population dynamics under worst-case (or best-case) assumptions for some types and probabilistic assumptions for other types. It is a natural generalization of the classic BP model to the optimization setting. As such it has a number of potential applications. There has been some work in the Operations Research literature, going back to the 1970s, on such BMDP models (see [Pliska 1976; Rothblum and Whittle 1982]). However, the existing work typically focuses on (discounted and long-run average) reward criteria and growth rate optimization criteria, and also it does not consider algorithmic and computational complexity questions. The BMDP model under the basic extinction probability criterion, and its associated algorithmic problems, appear not to have been studied previously, despite the rich literature on branching processes, and nor have their 2-player game generalizations, BSSGs. Indeed, even classifying the computational complexity of basic qualitative and quantitative extinction problems for purely probabilistic BPs and SCFGs had received little attention prior to our predecessor work on RMCs and 1-RMCs ([Etessami and Yannakakis 2009]).

Main Results

We now outline our main results of this paper (with the key results highlighted in bold):

— We associate with every 1-RMDP and 1-RSSG, with n vertices, a system of *monotone nonlinear polynomial min/max equations*, in n variables, which has the vector form $\mathbf{x} = P(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n)$, and the equations are $x_i = P_i(\mathbf{x})$, where for $i = 1, \ldots, n$, $P_i(\mathbf{x})$ is an algebraic expression over the basis $\{+, *, \min, \max\}$ which uses only positive rational coefficients and constants. Such a system defines a monotone operator $P : \mathbb{R}^n_{\geq 0} \mapsto \mathbb{R}^n_{\geq 0}$.
[Main Result 0]: We show that the Least Fixed Point solution of this system, $q^* \in \mathbb{R}^n_{\geq 0}$, exists and that q^*_i is precisely the value of the termination game starting at vertex i of the 1-RSSG.
These equations generalize both the standard linear-min-max Bellman's equations for certain basic classes of finite-state MDPs (see, e.g., [Puterman 1994; Filar and Vrieze 1997]) and the monotone systems of nonlinear polynomial equations for RMCs and 1-RMCs that were studied in [Etessami and Yannakakis 2009]. They imme-

diately yield a natural value iteration method which converges monotonically to the game values. Namely, beginning with the vector $\mathbf{x}_0 = \mathbf{0}$, iterate $\mathbf{x}_{i+1} = P(\mathbf{x}_i)$, $i = 1, 2, \ldots$. This sequence converges, in the limit, to the game values, but is in the worst case very slow to converge to within a desired error. (Specifically, even for a fixed 1-RMC it can require 2^i iterations to converge to within i bits of precision, see [Etessami and Yannakakis 2009]. Furthermore, using examples from [Esparza et al. 2010] one can show that for a 1-RMC with encoding size $O(n)$, value iteration can require 2^{2^n} iterations, starting from $\mathbf{0}$, to converge to within a single bit of precision.)

— **[Main Result 1]:** We prove a strong *Stackless & Memoryless (SM) Determinacy* result for 1-RSSG termination games: We show that both players have optimal deterministic strategies that use neither the prior history of the game, nor the contents of the call stack (of pending recursive calls), but only the current vertex in order to choose their next move. Our proof uses a strategy improvement argument, and is established by studying subtle analytic properties of certain power series associated with these stochastic games. The technique we develop is rather general and flexible, and adaptations of it have already had applications for other classes of stochastic games. We shall describe some of these extensions in the conclusion section.
We observe on the other hand that the existence of optimal strategies fails badly even for (maximizing) 2-exit RMDPs. Namely, optimal strategies, of any kind, do not always exist for 2-exit RMDPs under the objective of maximizing termination probability, only ϵ-optimal strategies exist, and in general the SM strategies can all be the worst possible strategies. The same holds on 1-RMDPs where the goal is to maximize the probability of reaching a given vertex (not necessarily an exit) in any calling context (i.e., any call stack), rather than terminating at an exit.

— Using the nonlinear-min-max equations, we show that the quantitative termination *decision* problems for 1-RMDPs and 1-RSSGs can be decided in PSPACE by employing PSPACE decision procedures for the existential theory of reals. This matches our PSPACE upper bound for the special case of 1-exit RMCs in [Etessami and Yannakakis 2009] and, as shown there, it can not be improved substantially without resolving long standing open problems in the complexity of numerical computation, namely the square-root sum problem, as well as certain fundamental arithmetic circuit decision problems. Both these problems reduce to deciding whether the termination probability of a 1-RMC is $\geq p$, and it has been an open question whether these problems are even contained in NP.[1]

— We give a simple algorithm that determines if the value of the termination game for 1-RSSGs (and 1-RMDPs) is 0 in polynomial time.

— **[Main Result 2:]** We show that for both maximizing and minimizing 1-RMDPs, the qualitative termination problem (is the maximum/minimum termination probability equal to 1?) can be decided in polynomial time. We do this by providing criteria for almost sure termination for 1-RMDPs based on the optimal spectral radius of associated families of non-negative matrices, and using graph decomposition methods and linear programming to obtain the P-time upper bound.
It follows from this and our SM-determinacy result that the qualitative termination problem for 1-RSSGs can be decided in NP $\cap$ coNP.

— We show that the well known *quantitative* termination problem for Condon's finite-state simple stochastic games reduces (via a polynomial-time many-one reduction) to the qualitative termination problem for 1-RSSGs. We do not know a reduction in the other direction.

[1] In the conclusion section, we shall reference much more recent work together with Alistair Stewart [Etessami et al. 2012b], in which we have shown that quantitative termination *approximation* problems for 1-RMDPs (and 1-RSSGs) can be solved in P-time (in FNP, respectively).

We note that, by contrast, for finite-state SSGs the qualitative termination problem is decidable in polynomial time. We in fact show a more general result: it is decidable in polynomial time also for the restricted class of 1-RSSGs that are *linearly recursive*. For the class of linearly-recursive (maximizing or minimizing) 1-RMDPs, we can in fact compute exactly the value (the optimal termination probability) in polynomial time (it is a rational number in this case).

— **[Main Result 3]:** We show that for multi-exit RMDPs and RSSGs the situation is far worse. Quantitative termination for general (maximizing or minimizing) RMDPs is undecidable, even when the number of exits in bounded by a fixed constant, and even when the RMDP is only linearly recursive. Furthermore, even the qualitative termination problem (both in the supremum = 1 sense and in the witness sense) for (both maximizing and minimizing) multi-exit RMDPs is undecidable. It is also undecidable for any fixed $\epsilon > 0$, to distinguish whether the supremum termination probability for a maximizing multi-exit RMDP is 1 or is less than ϵ. So the optimal probabilities can not even be approximated in a strong sense for maximizing multi-exit RMDPs, with any amount of resources.
Furthermore, we show that these undecidability holds already in the setting of 1-exit RMDPs, for both the quantitative and qualitative *model checking* problems for 1-RMDPs against regular, ω-regular, or LTL properties. More specifically, we show that already for a *fixed* LTL (or regular) property, and given a labeled 1-RMDP as input, it is undecidable whether there exists a strategy for the controller under which the LTL property holds with probability 1. Moreover, we show that the optimal probability can not be computed to within any nontrivial constant (additive) factor.
Our undecidability results are derived in part from classic and more recent undecidability results for Probabilistic Finite Automata (PFA) [Paz 1971; Condon and Lipton 1989; Blondel and Canterini 2003]. We in fact show that PFAs can be viewed as essentially a special case of linearly-recursive multi-exit RMDPs.

Related work. Both MDPs and stochastic games have a vast literature, dating back to Bellman and Shapley (see, e.g., [Puterman 1994; Feinberg and Shwartz 2002; Filar and Vrieze 1997; Neyman and Sorin 2003]). In particular, there are well-known efficient algorithms for optimizing finite-state MDPs with reward-based objectives. Finite-state MDPs where the objectives are specified by desired properties of the trajectories have also been studied for a long time, in connection with the verification of finite state MDPs against temporal properties (see, e.g., [Courcoubetis and Yannakakis 1998; 1995; Vardi 1985; Hart et al. 1983]). [Courcoubetis and Yannakakis 1998] provides efficient algorithms for ω-regular model checking of finite-state MDPs.

Our earlier work [Etessami and Yannakakis 2009; 2012] developed the basic theory of RMCs and studied efficient algorithms for both their reachability analysis and model checking. We showed, among many other results, that qualitative termination (and even model checking of ω-regular properties) for 1-RMCs can be decided in polynomial time in the size of the 1-RMC, and that quantitative model checking of general RMCs can be done in PSPACE in the size of the RMC. Although countable state MDPs are studied extensively in the MDP literature (see, e.g., [Puterman 1994; Feinberg and Shwartz 2002]), the concise representations afforded by RMDPs, and its algorithmic properties, have apparently not been studied.

Our polynomial time algorithms for deciding qualitative termination problems for 1-RMDPs were partly inspired by work by Denardo and Rothblum [2005; 2006] on *Multi-Matrix Multiplicative Systems*. They study families of square nonnegative matrices, which arise from choosing each matrix row independently from a choice of rows, and they give LP characterizations of when the spectral radius of all matrices in the family

will be ≥ 1 or > 1. None of our results follow from theirs, but we use techniques similar to theirs, along with other techniques, to obtain our upper bounds.

This paper is based on two conference papers that appeared in 2005 and 2006 [Etessami and Yannakakis 2005; 2006]. Since then, a manuscript of this full paper has been made available on our web page. Since the publication of the conference papers, there have been a large number of follow-up papers by ourselves and by others, building on this work and extending it in various different directions. We shall describe some of this subsequent work in the conclusion section of this paper.

Here we want to particularly mention one of these subsequent works, which was published by ourselves in 2006 ([Etessami and Yannakakis 2008]), where we extended the models of this paper to *recursive concurrent stochastic games* (RCSGs), where the game is no longer turn-based, both players choose moves simultaneously and independently at each state, and the game is thus an imperfect information game. Finite-state concurrent stochastic games have been studied in the verification literature in computer science (see, e.g., [de Alfaro et al. 2007; de Alfaro and Majumdar 2004; Chatterjee et al. 2006]). RCSGs constitute a more general model than RSSGs, and the generalization changes the nature of the model in various ways, both in terms of the classes of strategies required for optimality, as well as for the computational complexity of relevant problems. We showed in [Etessami and Yannakakis 2008] that some complexity results which hold for 1-exit RSSGs can be suitably extended to 1-exit RCSGs, whereas other results can not be extended because of concrete complexity-theoretic reasons. The journal version of [Etessami and Yannakakis 2008] has already appeared in a special issue of invited papers selected from the conference where it was published. Because of the timing of the publication of [Etessami and Yannakakis 2008], we would like to clarify its precise relationship to this paper. Most importantly: all of the results in [Etessami and Yannakakis 2008] assume as given, and directly build upon, the results established in this paper. In particular, although some of the results in [Etessami and Yannakakis 2008] generalize some results in this paper, their proofs use, without proof, results that we prove for the first time in this paper, which were only announced in the conference versions of this paper [Etessami and Yannakakis 2005; 2006].

Whenever it is appropriate to do so throughout this paper, we will point out cases where results in [Etessami and Yannakakis 2008] are related to, or generalize, results established in this paper. We will not reprove any results here which are proved directly in [Etessami and Yannakakis 2008], but in order to make this paper self-contained, we will repeat some arguments whose generalizations appear in [Etessami and Yannakakis 2008]. We will discuss in more detail the results established in [Etessami and Yannakakis 2008], as well as in other more recent papers, in the conclusions of this paper, where it will be easier to compare them to results established here.

In the conclusion section we shall also mention some more recent and closely related joint work with Alistair Stewart [Etessami et al. 2012a; 2012b], in which we have shown that quantitative termination *approximation* problems for 1-RMDPs can be solved in polynomial time, using algorithms based on a generalization of Newton's method. (And it follows from these results that for 1-RSSGs the termination value approximation problem is in FNP.) We will elaborate on these results in the conclusion section.

Organization of the paper.
The rest of the paper is organized as follows. In Section 2 we define RMDPs and RSSGs, and define the basic problems that we will study. Sections 3-9 deal with 1-exit RMDPs and RSSGs. In Section 3 we derive a system of nonlinear min-max equations for 1-RSSGs, whose least nonnegative solution (the 'least fixed point') is the vector of game values for all the different starting vertices of the game. In Section 4 we show

that both players have deterministic stackless and memoryless optimal strategies in these games. In Section 5 we show that the quantitative termination problems for 1-RSSGs (and 1-RMDPs) can be solved in PSPACE. We also show that the value 0 question for 1-RSSGs can be solved in polynomial time. Section 6 concerns the qualitative termination (value=1) problem for 1-RMDPs. We show that the problem can be solved in polynomial time for both maximizing and minimizing 1-RMDPs. Section 7 concerns the qualitative problem for 1-RSSGs; we show that it is in NP∩coNP, and that it is at least as hard as the quantitative problem for finite-state (not recursive) simple stochastic games considered by Condon. Section 8 concerns linearly recursive 1-RMDPs and 1-RSSGs; we give polynomial-time algorithms for the quantitative problem for 1-RMDPs, and for the qualitative problem for 1-RSSGs. In Section 9 we define Branching Markov Decision Processes and Games, and establish their relation with 1-RMDPs and 1-RSSGs respectively. Section 10 shows undecidability of the termination problems for multi-exit RMDPs (and 1-RSSGs). We also briefly recall there the basic concepts related to model checking, and show that model checking of ω-regular or LTL properties for (maximizing or minimizing) 1-RMDPs is undecidable. We conclude in Section 11, where we also describe some of the more recent results that have built on and extended the results in this paper.

2. DEFINITIONS AND BACKGROUND

In this section we will give the basic definitions on the models and the problems that we will study. For intuition regarding the definition, the reader is referred back to the example RSSG described in Example 1.1, and depicted in Figure 1.

2.1. Recursive Simple Stochastic Games and Subclasses

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

— A set N_i of *nodes*, with a distinguished subset En_i of *entry* nodes and a (disjoint) subset Ex_i of *exit* nodes.
— 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) a component. To each box $b \in B_i$, we associate a set of *call ports*, $\boldsymbol{Call}_b = \{(b, en) \mid en \in En_{Y(b)}\}$, and a set of *return ports*, $\boldsymbol{Return}_b = \{(b, ex) \mid ex \in Ex_{Y(b)}\}$. Let $\boldsymbol{Call}^i = \cup_{b \in B_i} \boldsymbol{Call}_b$, $\boldsymbol{Return}^i = \cup_{b \in B_i} \boldsymbol{Return}_b$, and let $Q_i = N_i \cup Call^i \cup Return^i$ be the set of all nodes, call ports and return ports; we refer to these as the *vertices* of component A_i.
— A mapping $\mathtt{pl}_i : Q_i \mapsto \{0, 1, 2\}$ that assigns to every vertex a *player*. Player 0 represents "chance" or "nature", player 1 is called the maximizing player and player 2 the minimizing player. We assume $\mathtt{pl}_i(u) = 0$ for all $u \in Call_i \cup Ex_i$.
— 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 $u \in (N_i \setminus Ex_i) \cup \boldsymbol{Return}^i$, the destination $v \in (N_i \setminus En_i) \cup \boldsymbol{Call}^i$, and x is either (i) a real number $p_{u,v} \in (0, 1]$ (the transition probability) if $\mathtt{pl}_i(u) = 0$, or (ii) $x = \bot$ if $\mathtt{pl}_i(u) = 1$ or 2. Furthermore they must satisfy the consistency property: for every $u \in \mathtt{pl}_i^{-1}(0)$, $\sum_{\{v' \mid (u, p_{u,v'}, v') \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or exit node, neither of which have outgoing transitions, in which case by default $\sum_{v'} p_{u,v'} = 0$.

We use the symbols (N, B, Q, δ, etc.) without a subscript, to denote the union over all components. Thus, eg. $N = \cup_{i=1}^{k} N_i$ is the set of all nodes of A, $\delta = \cup_{i=1}^{k} \delta_i$ the set of all transitions, etc.

For computational purposes, we assume as usual that the transition probabilities $p_{u,v}$ are rational, and are specified by giving in binary the numerator and denominator.

The *size* of a RSSG is the space (number of bits) that is needed for its description, including all the nodes, boxes, and transitions, as well as the transition probabilities.

An RSSG A defines a global denumerable Simple Stochastic Game (SSG) $M_A = (V = V_0 \cup V_1 \cup V_2, \Delta, \mathtt{pl})$ 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. The sequence β is the stack of pending recursive calls; we will refer to it sometimes as the *context* of the state. The empty sequence is denoted by ϵ. We will sometimes write $\langle u \rangle$ instead of $\langle \epsilon, u \rangle$ for the state where the RSSG is at vertex u with empty context, i.e. no pending recursive call.

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 $u \in Q$.
(2) if $\langle \beta, u \rangle \in V$ and $(u, x, v) \in \delta$, then $\langle \beta, v \rangle \in V$ and $(\langle \beta, u \rangle, x, \langle \beta, v \rangle) \in \Delta$.
(3) if $\langle \beta, (b, en) \rangle \in V$ and $(b, en) \in \mathit{Call}_b$, then $\langle \beta b, en \rangle \in V$ and $(\langle \beta, (b, en) \rangle, 1, \langle \beta b, en \rangle) \in \Delta$.
(4) if $\langle \beta b, ex \rangle \in V$ and $(b, ex) \in \mathit{Return}_b$, 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: the RSSG starts initially at some vertex with no pending recursive calls. Item 2 corresponds to control staying within a component. Item 3 is when a new recursive call is initiated, i.e. a component is entered via a box. Item 4 is when a call terminates, control exits a box and returns to the calling component.

The player mapping is extended from the set Q of vertices of the RSSG A to the set V of states of M_A: the mapping $\mathtt{pl} : V \mapsto \{0, 1, 2\}$ is given by $\mathtt{pl}(\langle \beta, u \rangle) = \mathtt{pl}(u)$. The set of vertices V is partitioned into V_0, V_1, and V_2, where $V_i = \mathtt{pl}^{-1}(i)$.

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. If we wish to view M_A as a non-terminating SSG, we can consider the terminating states as absorbing states of M_A, with a self-loop of probability 1.

An RSSG where no vertices are assigned to the minimizing player (and hence $V_2 = \emptyset$) is called a *maximizing Recursive Markov Decision Process*; similarly, an RSSG where no vertices are assigned to the minimizing player (and hence $V_2 = \emptyset$) is called a *minimizing Recursive Markov Decision Process*. An RSSG where all the vertices are assigned to player 0 (i.e., $V_1 \cup V_2 = \emptyset$) is called a *Recursive Markov Chain* (RMC) ([Etessami and Yannakakis 2009]). An RSSG where $V_0 \cup V_2 = \emptyset$ is called a Recursive Graph or *Recursive State Machine* (RSM) ([Alur et al. 2005]).

Define *1-RSSGs* (also referred to as a 1-exit RSSG), to be those RSSGs where every component has 1 exit, and likewise define *1-RMDPs* and *1-RMCs*. The example depicted in Figure 2 is a 1-RSSG, because its only component, f, has only one exit. By contrast, the earlier example in Figure 1 is not a 1-RSSG, because both of its components have 2 exits. W.l.o.g., one can assume that every component of a RSSG has 1 entry, because multi-entry RSSGs can be transformed to equivalent 1-entry RSSGs with polynomial blowup (similar to the RSM transformations [Alur et al. 2005]). However, the analogous statement does not hold for exits, as we shall see.

We call a RSSG (RMDP, RMC, etc.) *linearly-recursive* or *linear* if there is no path of transitions in any component from any return port to a call port. Neither the example in Figure 1, nor the example in Figure 2 are linear. In particular, the RSSG in Figure 1 is not linear, because in the component A_1 there is a direct transition from one of the box-exits of box b_1 to the box-entry of the same box. If we remove this transition, then the resulting RSSG becomes linear. The RSSG in Figure 2 is not linear because there is a direct transition from the unique exit of box b_1 to the entry of box b_2.) The linearly-recursive restriction corresponds to the standard notion of linear recursion in

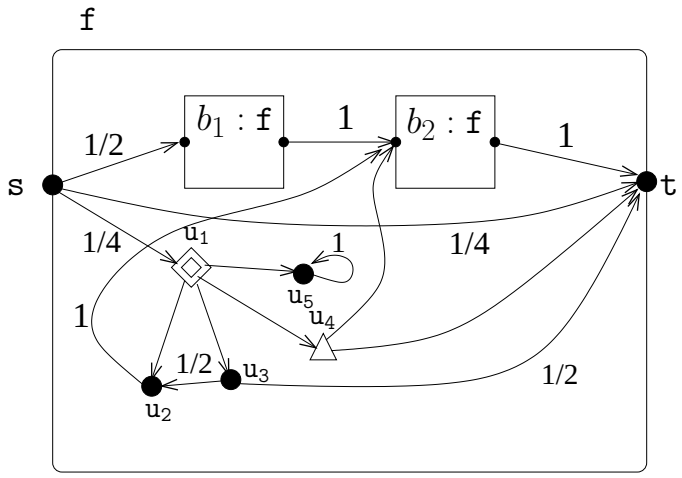

Fig. 2. Example 1-exit RSSG (1-RSSG)

programs. Linearly-recursive RMCs (without players) are much easier to analyse than general RMCs: termination and reachability probabilities (as well as probabilities of more general temporal properties) are rational and can be computed in polynomial time, with the same complexity as for finite-state Markov chains; see [Etessami and Yannakakis 2009; 2012]. We will see that linearly-recursive 1-RMDPs and 1-RSSGs also preserve all the positive features of finite state MDPs and SSGs; however multi-exit linear RMDPs and RSSGs are not at all that easy.

In our definitions of RMDPs and RSSGs we have used for convenience separate vertices to represent the probabilistic steps and the players' actions. As in the case of finite MDPs and SSGs, this is equivalent (both in terms of expressiveness and in terms of computational efficiency) to an alternative model where there are no separate probabilistic vertices, but rather at each vertex controlled by a player, the player selects an action from a set of available actions at the vertex, and then a probabilistic transition takes place where the transition probabilities depend on the vertex and the selected action.

2.2. Objectives, strategies and value of the game

Markov decisions processes and stochastic games have been studied under many different kinds of objectives. We can categorize these objectives under two types: in the first type, there is a reward (payoff) structure given for the individual vertices and actions, and the objectives of the players are to maximize or minimize the aggregated reward during the execution (e.g., the discounted total reward, or average reward per step, etc.). In the second type, we are given a certain desirable or undesirable property of the possible executions (which amounts to an event over the probability space of possible executions, or trajectories, once strategies are fixed), and the objectives of the players are to maximize or minimize the probability that the execution satisfies this property.

In this paper we will study RMDPs and RSSGs with objectives of the second type, and in particular we will study the most basic type of reachability objective, where the goal of the players is to minimize or maximize the probability that the process terminates (or that it terminates at a particular exit). We have to formalize the questions precisely, and there are subtle issues that will require careful treat-

ment. We will first require some definitions. For a finite or countable set Z, let $\mathcal{D}(Z)$ denote the set of probability distributions over Z. For a distribution $d \in \mathcal{D}(Z)$, let $\texttt{support}(d) = \{i \in Z \mid d(i) > 0\}$. We say d has finite support if $|\texttt{support}(d)| < \infty$.

We now define the notion of a *strategy* for players in an infinite-state simple stochastic game (including those arising from RSSGs). For any denumerable-state SSG $M = (V, \Delta, \texttt{pl})$, a general *strategy* σ for player i, $i \in \{1, 2\}$, is defined by a function $\sigma : V^*V_i \mapsto \mathcal{D}(V)$, 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), $\sigma(ws) \in \mathcal{D}(V)$ is a (finite support) probability distribution on the next state, where moreover it must also be the case that for all $s' \in \texttt{support}(\sigma(ws))$, we have $(s, \bot, s') \in \Delta$. In other words, the move from s to s' must be *available* to the player at that state. A strategy is *deterministic* (or *pure*) if we always have $|\texttt{support}(\sigma(ws))| = 1$, meaning with probability 1 a move is chosen to one particular new state, for all such histories ws where $s \in V_i$. Otherwise it is called a *randomized* (*mixed*) strategy. A deterministic strategy can thus more simply be viewed as a function $\sigma : V^*V_i \mapsto V$.

Let us focus again on the infinite-state SSG, M_A corresponding to a given RSSG A. Let Ψ_i denote the set of all strategies for player i. Given a start node u, a strategy $\sigma \in \Psi_1$ for player 1, and a strategy $\tau \in \Psi_2$ for player 2, we define a new Markov chain (with initial state u) $M_A^{u,\sigma,\tau} = (\mathcal{S}, \Delta')$. The states $\mathcal{S} \subseteq \langle \epsilon, u \rangle V^*$ of $M_A^{u,\sigma,\tau}$ are non-empty sequences of states of M_A, which must begin with $\langle \epsilon, u \rangle$. Inductively, if $ws \in \mathcal{S}$, then: (0) if $s \in V_0$ and $(s, p_{s,s'}, s') \in \Delta$ then $wss' \in \mathcal{S}$ and $(ws, p_{s,s'}, wss') \in \Delta'$; (1) if $s \in V_1$ and $\sigma(ws)(s') = p > 0$ (where $(s, \bot, s') \in \Delta$) then $wss' \in \mathcal{S}$ and $(ws, p, wss') \in \Delta'$; (2) if $s \in V_2$ and $\tau(ws)(s') = p > 0$ (where $(s, \bot, s') \in \Delta$) then $wss' \in \mathcal{S}$ and $(ws, p, wss') \in \Delta'$.

Let u be a given initial vertex. It follows from the definition of M_A that the only terminating states reachable from $\langle \epsilon, u \rangle$ are of the form $\langle \epsilon, ex \rangle$ where ex is an exit node in the same component as u. Given initial vertex u, and exit ex in the same component, and given strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$, for $k \geq 0$, let $q^{k,\sigma,\tau}_{(u,ex)}$ be the probability that, in $M_A^{u,\sigma,\tau}$, 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 $\langle \epsilon, ex \rangle$ in any number of steps. (Note, the limit exists: it is a monotonically non-decreasing sequence bounded by 1). Let $q^{k}_{(u,ex)} = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} q^{k,\sigma,\tau}_{(u,ex)}$ and let $q^{*}_{(u,ex)} = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} q^{*,\sigma,\tau}_{(u,ex)}$. Next, for a strategy $\sigma \in \Psi_1$, let $q^{k,\sigma}_{(u,ex)} = \inf_{\tau \in \Psi_2} q^{k,\sigma,\tau}_{(u,ex)}$, 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 $q^{k,\cdot,\tau}_{(u,ex)} = \sup_{\sigma \in \Psi_1} q^{k,\sigma,\tau}_{(u,ex)}$, and let $q^{*,\cdot,\tau}_{(u,ex)} = \sup_{\sigma \in \Psi_1} q^{*,\sigma,\tau}_{(u,ex)}$.

From very general determinacy results, namely Martin's *Blackwell determinacy* [Martin 1998] (see also [Maitra and Sudderth 1998]), which applies to all two-player zero-sum stochastic games with countable state spaces and bounded Borel measurable payoff functions, it follows that the games M_A are *determined*, meaning that

$$q^{*}_{(u,ex)} \doteq \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} q^{*,\sigma,\tau}_{(u,ex)} = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} q^{*,\sigma,\tau}_{(u,ex)}$$

We call $q^{*}_{(u,ex)}$ the *value* of the RSSG termination game with start vertex u and terminating exit ex.[2] We can define similarly the termination game where only a start node u is specified, but not a terminating exit. That is, the maximizer's goal is for the game

[2]Let us remark, in response to a question raised by a referee, that for general multi-exit RMDPs and RSSGs, we know little about the nature of the values q^* of such games. In particular, in light of the undecidability results we shall establish for computing or even approximating the value for general RMDPs and RSSGs, it is almost certainly true that even when the specification of the RMDP or RSSG involves only rational numbers, their value can be a transcendental number. On the other hand, as we shall see, it follows from

to terminate (at any exit), while the minimizer's goal is for the game to run forever. We will use the above notations without the subscript ex for these termination games where an exit is not specified. Thus, $q_u^{k,\sigma,\tau}$ denotes the probability of termination in k steps under strategies σ, τ for the two players, $q_u^{*,\sigma,\tau}$ is the probability of termination in any number of steps, and q_u^* denotes the value of the termination game starting from vertex u (determinacy holds again). In the case of 1-RMDPs and 1-RSSGs, the component of u has only one exit ex, hence q_u^* is the same as $q_{(u,ex)}^*$.

Note that, in general, determinacy does not guarantee the existence of an optimal strategy for either player. By an *optimal* strategy for the maximizer (minimizer) we mean one that *achieves* a payoff at least (respectively, at most) the value of the game. We say a strategy σ (respectively, τ) *achieves* a given value r for a maximizing (respectively, minimizing) player, if $\inf_{\tau \in \Psi_2} q_{(u,ex)}^{*,\sigma,\tau} \geq r$ (respectively, $\sup_{\sigma \in \Psi_1} q_{(u,ex)}^{*,\sigma,\tau} \leq r$). For the games we consider, determinacy does imply the existence of ϵ-optimal strategies, for all $\epsilon > 0$, meaning strategies that achieve a payoff no less than $q_{(u,ex)}^* - \epsilon$ for the maximizer (no worse than $q_{(u,ex)}^* + \epsilon$ for the minimizer). This is so because the possible payoffs are bounded, $q_{(u,ex)}^{*,\sigma,\tau} \in [0,1]$ for these games.

Finite state MDPs and SSGs with reachability objectives are trivially a subclass of 1-exit linearly-recursive RMDPs and RSSGs respectively with a termination objective. In a finite state MDP or SSG with reachability objectives the players wish to maximizing/minimize the probability of reaching a desired set of target states starting from a given start state. We can assume without loss of generality that there is one target state ex which has no outgoing transitions: if there is a set R of target states, we can collapse them into one state ex and remove the outgoing transitions, since once a target state has been reached, the reachability objective has been met. Thus, we can view a finite-state SSG (or MDP) as a 1-RSSG (resp. 1-RMDP) that has one component with exit ex, and has no boxes. The value of the game in any finite MDP or SSG is rational of polynomial size (bit complexity) in the size of the input. In the case of MDPs, the value can be computed in polynomial time. In the case of SSGs, the complexity of computing the value is a well known open problem ([Condon 1992]). It is known to be in the classes PLS and PPAD (and thus, it is unlikely to be NP-hard unless NP=coNP); the decision question of whether the value exceeds a given rational (for example, is the value $\geq 1/2$?) is in NP$\cap$coNP.

In finite-state SSGs, both players can achieve the value of the game. Furthermore, the games are *memorylessly determined* ([Condon 1992]), meaning that both players have optimal deterministic memoryless strategies. A (deterministic) *memoryless strategy* is one which does not depend on the history prior to the current state of the stochastic game. In other words a deterministic memoryless strategy is given by a function from states belonging to a player to neighboring states. As we shall see, 1-RSSGs exhibit an even stronger form of memoryless determinacy. We say that a strategy of a RSSG A is a (deterministic) *Stackless & Memoryless (SM)* strategy if it is not only independent of the history of the game, but also independent of the current call stack, i.e., for every state $\langle \beta, v \rangle$ of the infinite game M_A, the action of the player at the state does not depend on the past history (how the trajectory reached $\langle \beta, v \rangle$), nor on the context β (the stack of boxes), but only depends on the current vertex v (and furthermore the strategy is deterministic). In other words, such a strategy just picks, for every vertex in the RSSG, a particular neighboring vertex to move to whenever it encounters that vertex (regardless of history or calling context). Such a strategy can be given simply by a function that maps every vertex associated with that player to one of its neighbors.

our results that for various special cases of RSSGs, notably 1-exit RSSGs, their game value is always an algebraic number (albeit, in general an irrational one).

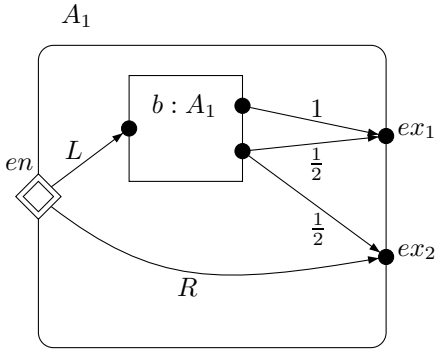

Fig. 3. *Maximizing termination probability at ex_1 in a 2-exit RMDP:* no optimal strategy exists.

Note that there are only finitely many such functions (but exponentially many in the number of vertices belonging to that player). We shall show that 1-RSSG termination games are SM-determined, meaning both players have optimal SM strategies. This fails badly even for multi-exit RMDPs, as we will now observe.

Already for the 2-exit RMDP termination problem, there need not be any optimal strategy at all. This is illustrated in Figure 3. This is a (linearly recursive) 2-exit RMDP that has one component containing a box mapped to the same component. The RMDP starts at node en and the objective is to maximize the probability of terminating at exit ex_1. It can easily be verified that the *supremum* probability of terminating at exit ex_1 starting from en is 1. Specifically, for every $n \geq 0$, consider the strategy L^nR, which chooses the transition L the first n times that vertex en is visited (thus making n nested recursive calls) and chooses the transition R the $(n+1)$th time, thus completing the last call at the second exit. After this, the process will successively return from the n recursive calls and terminate at one of the two exits. The only way that it will terminate at ex_2 is if it always returns from each box at the second port and follows the transition to ex_2; this will happen with probability $\frac{1}{2^n}$. Thus, under the strategy L^nR the process terminates eventually at ex_1 with probability $(1 - \frac{1}{2^n})$. Hence the value of the RMDP is 1. However, there is no optimal strategy for player 1 that actually achieves this value. The deeper the call stack is made by player 1, the higher the probability of termination at ex_1. However, at some depth n, player 1 finally has to decide to follow the transition R, otherwise it will never terminate; the probability of eventually terminating at ex_1 will then be $(1 - \frac{1}{2^n})$. Note also that in this example, any SM strategy for player 1 yields probability 0 of terminating at ex_1, so such strategies are all the worst possible here.

It is worth pointing out however that for the classes of *turn-based* (perfect information) countable-state stochastic games generated by RSSG termination games, we only need to consider deterministic memoryless strategies, i.e., randomization and memory do not add any extra power for either player, though the context (the stack of active boxes) is important. We state a general theorem capturing this for a suitable class of countable-state turn-based stochastic games which easily subsumes RSSGs termination games.

THEOREM 2.1. *Suppose $G = (V = V_0 \cup V_1 \cup V_2, \Delta, \mathtt{pl}, r)$ is a countable-state, turn-based (perfect-information) stochastic game, with a non-negative reward function r on transitions, and with expected total reward objective (which player 1 wants to maximize and player 2 wants to minimize), and such that, under all pairs of strategies used by the two players, the expected total reward is bounded by a fixed constant K. Suppose furthermore that G is* finitely branching*, meaning that for any state $u \in V$ there are*

a finite number of transitions of the form (u, x, v) in Δ (regardless of whether u is a player's state or a probabilistic state). Then:

(*1*) *Starting from any state $u \in V$, there exists a deterministic memoryless strategy τ^* for player 2 (the minimizer), in the game G starting at u.*
(*2*) *For every $\epsilon > 0$, starting from any state $u \in V$, player 1 has an ϵ-optimal deterministic memoryless strategy for the game G starting at u.*

In particular, such countable-state perfect information stochastic games are (deterministically) memorylessly determined.

Note that RSSG termination games can easily be placed in the reward framework described by Theorem 2.1. Namely, we can augment the countable-state SSG, M_A, associated with a RSSG A, with a non-negative reward function which gives 0 reward everywhere, except that after termination at the desired exit(s) there is an extra transition with reward 1 (with probability 1) to a new dead-end absorbing state which thereafter yields 0 reward. After termination at other (undesired) exits we transition to the dead-end via a 0 reward transition. It is clear that the expected total reward starting at a given vertex of the RSSG (in the empty calling context), under all strategies is the probability of termination at the desired exit(s) of the RSSG. Note that the total expected reward is bounded by 1 for all strategies of both players. Moreover, M_A is clearly finitely branching (in fact, there is a fixed upper bound on the branching at all states of M_A).

Theorem 2.1 is closely related to well known results in the MDP and stochastic game literature. Specifically, it is well known that for the 1-player countable-state MDP version of these games, with non-negative rewards and with the finite branching constraint, that if the goal is to minimize the total expected reward then the minimizer has an optimal deterministic memoryless strategy (see, e.g., Theorem 7.3.6 in [Puterman 1994]), and if the goal is to maximize the total expected reward and the total expected reward is bounded by a constant K then the maximizer has ϵ-optimal strategies (see Theorem 7.2.7 and Corollary 7.2.8 of [Puterman 1994], which are derived from [Ornstein 1969]). Theorem 2.1 can be proved by adapting these results to the setting of perfect-information stochastic games.

We give an outline of the proof of Theorem 2.1 below. The full proof is given in Appendix 11, since we do not actually use Theorem 2.1 further in the paper, and since closely related results are well known, as explained above.

In rough outline, one can first show that for such stochastic games the minimizer always has an optimal deterministic memoryless strategy by arguments similar to those for minimizing MDPs. Namely, one can associate optimality equations on countably many variables to the countable-state stochastic game, and use these to show that it is sufficient for the minimizer to always choose from each state a neighbor from which the value of the game is smallest. The finite branching condition guarantees that such a neighbor exists. To argue that the maximizer has ϵ-optimal strategies, one can argue that if we consider the value v^k of the k-step games associated with these stochastic total-reward games, they form underapproximations of the total reward value in the infinite-horizon game, such that as $k \to \infty$, the values v^k converge to the value v of the infinite horizon game. We can then consider the finite set, S'_k, of states that can possibly be encountered during the k-step finite-horizon game, and consider the infinite-horizon game G_k induced by this finite set of states, which proceeds just like the original infinite-horizon game, but as soon as a transition leaves S'_k, it now moves to a new dead-end state from which we will gain total reward 0 thereafter. This is a finite-state total-reward perfect information stochastic game with infinite horizon (and with the property that under all strategies the total expected reward is upper bounded

by the same fixed constant K). These games have value at least the value of the k-step game, and at most the value of the infinite-horizon game. For such finite-state stochastic games there are always memoryless optimal strategies for both maximizer and minimizer starting from a given state. Thus, since the values of the k-step games converge to the value of the infinite horizon game, for any $\epsilon > 0$ the maximizer has an ϵ-optimal strategy for the game, by just picking a sufficiently large k such that $v - v^k < \epsilon$, and mimicking the maximizer's optimal memoryless strategy in the game G_k when at a state inside S'_k, and playing arbitrarily (but memorylessly) outside of S'_k. For the details of the proof, see Appendix 11.

2.3. The central computational problems

We now formally describe the central computational problems we will address in this paper. Given a (1-exit or multi-exit) RMDP or RSSG A, an initial vertex u and an exit ex of the component of u, we wish to ask:

(1) The *qualitative* termination problem (`Qual-TP`): Is $q^*_{(u,ex)} = 1$?
(2) The *quantitative* termination problems: Given $r \in [0, 1]$, is $q^*_{(u,ex)} \geq r$? Is $q^*_{(u,ex)} = r$?
We may also wish to compute or approximate the exact probabilities $q^*_{(u,ex)}$.

More generally, we can ask model checking questions for general properties: given a RMDP or RSSG A and a property φ on the trajectories (executions) of A, what is the supremum probability with which player 1 can force the trajectory taken to satisfy the property φ? We will give the necessary definitions on properties and model checking in Section 10, where we discuss this problem and prove that it is generally undecidable.

In most of the paper we will focus on 1-RMDPs and 1-RSSGs. In this (single-exit) case, it will follow from the SM determinacy result in Section 4 that optimal deterministic SM strategies exist in 1-RSSG termination games for both the maximizing and minimizing player. Therefore, for 1-RSSGs, deciding whether $q^*_{(u,ex)} \geq r$, or $q^*_{(u,ex)} \leq r$, etc., is equivalent to deciding the existence of a strategy that achieves value r.

However, as the example in Figure 3 showed, this is not the case for general multi-exit maximizing RMDPs, and RSSGs. Thus for multi-exit RSSGs we may wish to consider also the following revised questions:

(1') The *witness qualitative* termination problem: Is there a strategy for maximizer (minimizer) that achieves value 1 (strictly less than 1, respectively)?
(2') The *witness quantitative* termination problems: Given $r \in [0, 1]$, is there a strategy for maximizer (minimizer) that achieves value at least (at most) r?

3. THE SYSTEM OF NONLINEAR MIN-MAX EQUATIONS FOR 1-RSSGS

We shall show that there is a monotone nonlinear min-max system of equations associated with a 1-RSSG, which captures its termination values (as the least non-negative solution to the equations). These systems generalize both the linear Bellman's equations for MDPs, as well as the nonlinear system of polynomial equation for RMCs studied in [Etessami and Yannakakis 2009]. Recall that for 1-RSSGs q^*_u denotes the value of the termination game starting at a vertex u. Let us use a variable x_u for each such unknown q^*_u, and let $\mathbf{x}$ be the vector of all $x_u, u \in Q$. The system S_A has one equation of the form $x_u = P_u(\mathbf{x})$ for each vertex u. Suppose that u is in component A_i with (unique) exit ex. There are five cases based on the "*Type*" of u. We partition the vertices into five types: $exit$, $call$, $random$, max, and min. Exit and call vertices have no outgoing edges, while the other three types $rand, max, min$ have outgoing edges that are controlled respectively by randomness (chance), player 1 (the maximizer) and player 2 (the minimizer).

(1) $u \in Type_{exit}$: $u \in Ex$. In this case: $x_u = 1$.
(2) $u \in Type_{rand}$: $\mathtt{pl}(u) = 0$ and $u \in Q \setminus (Ex \cup \mathit{Call})$. In this case $x_u = \sum_{\{v|(u,p_{u,v},v)\in\delta\}} p_{u,v}x_v$.
(3) $u \in Type_{call}$: $u = (b, en)$ is a call port: In this case $x_{(b,en)} = x_{en} \cdot x_v$ where v is the (unique) return port of box b; that is, $v = (b, ex')$, where ex' is the exit of $A_{Y(b)}$.
(4) $u \in Type_{max}$: $\mathtt{pl}(u) = 1$. In this case $x_u = \max_{\{v|(u,\perp,v)\in\delta\}} x_v$. (If u has no outgoing transitions, we define $\max(\emptyset) = 0$.)
(5) $u \in Type_{min}$: $\mathtt{pl}(u) = 2$. In this case $x_u = \min_{\{v|(u,\perp,v)\in\delta\}} x_v$. (If u has no outgoing transitions, we define $\min(\emptyset) = 0$.)

In vector notation, we denote the system S_A by $\mathbf{x} = P(\mathbf{x})$. Given 1-RSSG A, we can easily construct S_A in linear time.

Example 3.1.
Consider the 1-RSSG of Figure 2. The system has one variable and one equation for each vertex. $x_s = \frac{1}{4}x_{u_1} + \frac{1}{4}x_t + \frac{1}{2}x_{(b_1,s)}$, $x_{u_1} = \max(x_{u_2}, x_{u_3}, x_{u_4}, x_{u_5})$, $x_{u_2} = x_{(b_2,s)}$, $x_{u_3} = \frac{1}{2}x_{u_2} + \frac{1}{2}x_t$, $x_{u_4} = \min(x_{(b_2,s)}, x_t)$, $x_{u_5} = x_{u_5}$, $x_t = 1$, $x_{(b_1,s)} = x_s x_{(b_1,t)}$, $x_{(b_1,t)} = x_{(b_2,s)}$, $x_{(b_2,s)} = x_s x_{(b_2,t)}$, $x_{(b_2,t)} = x_t$ □

We now identify a particular solution to $\mathbf{x} = P(\mathbf{x})$, called the *Least Fixed Point* (LFP) solution, and we show that it is precisely the termination game value vector $\mathbf{q}^*$.

For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, define the partial-order $\mathbf{x} \leq \mathbf{y}$ to mean $x_j \leq y_j$ for every coordinate j. For $D \subseteq \mathbb{R}^n$, a mapping $H : \mathbb{R}^n \mapsto \mathbb{R}^n$ is called *monotone* on D, if: for all $\mathbf{x}, \mathbf{y} \in D$, if $\mathbf{x} \leq \mathbf{y}$ then $H(\mathbf{x}) \leq H(\mathbf{y})$. Define $P^1(\mathbf{x}) = P(\mathbf{x})$, and define $P^k(\mathbf{x}) = P(P^{k-1}(\mathbf{x}))$, for $k > 1$. Let $\mathbf{q}^* \in \mathbb{R}^n_{\geq 0}$ denote the n-vector $\langle q^*_u \mid u \in Q\rangle$. For $k \geq 0$, let $\mathbf{q}^k$ denote, similarly, the n-vector $\langle q^k_u \mid u \in Q\rangle$ of the values of the k-step termination game for the different starting vertices $u \in Q$. 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 $\mathbf{x}^k = P(\mathbf{x}^{k-1}) = P^k(\mathbf{0})$.

For the equation system $\mathbf{x} = P(\mathbf{x})$ corresponding 1-RSSG, it is easy to check (case by case, based on the five types of equations) that the operator P is monotone on $\mathbb{R}^n_{\geq 0}$, and that moreover it is monotone on the unit n-cube $[0,1]^n$ and maps $[0,1]^n$ to itself. Since the n-cube $[0,1]^n$ forms a complete lattice (under the partial order on n-vectors given by componentwise inequality), by the Tarski-Knaster fixed point theorem ([Tarski 1955]) the operator P has a a *Least Fixed Point* (LFP) $x^* \in [0,1]^n$. As we shall establish in the following theorem, this LFP is precisely the vector q^* of optimal termination probabilities. (The theorem actually establishes the existence of the LFP in this setting, so Tarski's fixed point theorem is not used.)

THEOREM 3.2. [3] *Let $\mathbf{x} = P(\mathbf{x})$ be the system S_A associated with 1-RSSG A.*

(1) The map $P : \mathbb{R}^n \mapsto \mathbb{R}^n$ is monotone on $\mathbb{R}^n_{\geq 0}$. Hence, for all $k \geq 0$, $\mathbf{0} \leq \mathbf{x}^k \leq \mathbf{x}^{k+1}$.

(2) For all $k \geq 0$, $\mathbf{q}^k \leq \mathbf{x}^{k+1} \leq \mathbf{q}^{2^k}$.

[3] We note here that subsequent to the conference publication of this paper we established that a similar theorem holds for the more general class of *1-exit recursive concurrent stochastic games* (1-RCSG): see Theorem 3.1 of [Etessami and Yannakakis 2008]. Namely, we showed there that systems of nonlinear equations that additionally use a minimax "value" operator for 2-player zero-sum matrix games can be used to give a similar characterization of the game values for 1-RCSGs. Here we provide the theorem and full proof for 1-RSSG and nonlinear-min-max equations. We do so not just for completeness, but because the proof differs from the proof for 1-RCSGs in important ways which we shall use. In particular, the proof here will directly yield the existence of optimal *deterministic* Stackless and Memoryless optimal strategies for the minimizing player in 1-RSSG termination games, whereas such deterministic strategies do not even exist for general 1-RCSGs.

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

PROOF. We prove each of the assertions of the theorem in turn.

(1) That P is monotone on $\mathbb{R}^n_{\geq 0}$ follows immediately from the fact that all coefficients in the polynomials P_j defining P are non-negative, and the fact that, if $x \leq y$, then clearly $\min_{i\in I} x_i \leq \min_{i\in I}(y_i)$, and $\max_{i\in I} x_i \leq \max_{i\in I} y_i$, for any subset $I \subseteq \{1, \ldots, n\}$. Thus, if $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y}$ then $\mathbf{0} \leq P(\mathbf{x}) \leq P(\mathbf{y})$. By induction on $k \geq 0$, $\mathbf{0} \leq \mathbf{x}^k \leq \mathbf{x}^{k+1}$.
(2) By induction on $k \geq 0$. For $k = 0$: $\mathbf{x}^1 = P(\mathbf{0})$ is an n-vector where $P_u(\mathbf{0}) = 1$ if $u \in Ex$, and $P_u(\mathbf{0}) = 0$ otherwise. Hence, for each vertex u, $\mathbf{x}^1_u = \mathbf{q}^0_u$, the probability of terminating in (at most) 0 steps starting from u. Hence, also clearly, $\mathbf{x}^1_{(u,ex)} \leq \mathbf{q}^{2^0}_{(u,ex)}$.
Inductively, suppose $\mathbf{q}^k \leq \mathbf{x}^{k+1} \leq \mathbf{q}^{2^k}$. Consider $\mathbf{x}^{k+2}_u$ for a vertex u. There are five cases, based on what type of vertex u is:
(a) $u \in Type_{exit}$. If $u \in Ex$, then clearly $\mathbf{q}^j_u = 1$ for all $j \geq 0$. Note that since $P_u(\mathbf{x}) = 1$, also $\mathbf{x}^{j+1}_u = P_u(\mathbf{x}^j) = 1$, for all $j \geq 0$. Thus $\mathbf{q}^j_u = \mathbf{x}^{j+1}_u = \mathbf{q}^{2^j}_u = 1$ for all $j \geq 0$.
(b) $u \in Type_{rand}$. In this case, $\mathbf{q}^{k+1}_u = \sum_v p_{u,v}\, \mathbf{q}^k_v$. Thus, by inductive hypothesis

$$\mathbf{x}^{k+2}_u = P_u(\mathbf{x}^{k+1}) = \sum_v p_{u,v}\, \mathbf{x}^{k+1}_v \geq \sum_v p_{u,v}\, \mathbf{q}^k_v = \mathbf{q}^{k+1}_u$$

Likewise, by inductive hypothesis

$$\mathbf{x}^{k+2}_u = \sum_v p_{u,v}\, \mathbf{x}^{k+1}_v \leq \sum_v p_{u,v}\, \mathbf{q}^{2^k}_v = \mathbf{q}^{2^k+1}_u \leq \mathbf{q}^{2^{k+1}}_u$$

(c) $u \in Type_{call}$. Here, $u = (b, en) \in \mathit{Call}_b$. Let ex be the (unique) exit node of the component of u, and ex' the exit node of the component $A_{Y(b)}$ corresponding to the box b. We argue first that $\mathbf{q}^{k+1}_u \leq \mathbf{q}^{k-1}_{en} \cdot \mathbf{q}^{k-1}_{(b,ex')} \leq \mathbf{q}^{2k}_u$.
We can see that the first inequality holds as follows. For any strategies σ, τ of the two players, in the resulting Markov chain $M_A^{u,\sigma,\tau}$, starting from $\langle \epsilon, u\rangle$, in order for a trajectory to reach the exit $\langle \epsilon, ex\rangle$ which is in the same component as u and terminate in at most $k+1$ steps, it first needs to transition to $\langle b, en\rangle$ (in one step); then it needs to get in some number m of steps from $\langle b, en\rangle$ to $\langle b, ex'\rangle$ (i.e., get from the entry en of the component $A_{Y(b)}$ labeling box b to the unique exit ex' of $A_{Y(b)}$); then it will need to transition from $\langle b, ex'\rangle$ to $\langle \epsilon, (b, ex')\rangle$ (in one step); then it will need to get from that box-exit to $\langle \epsilon, ex\rangle$ in some number m' of steps; such that, overall, $m + m' + 2 \leq k + 1$, i.e. $m + m' \leq k - 1$. In the formula for the upper bound, we have relaxed the requirements and only require that each of m and m' is $\leq k - 1$. Note that this holds regardless what strategies σ and τ are employed. Hence the first inequality.
For the second inequality, $\mathbf{q}^{k-1}_{en} \cdot \mathbf{q}^{k-1}_{(b,ex')} \leq \mathbf{q}^{2k}_u$, observe that one way to get from $\langle \epsilon, u\rangle$ to $\langle \epsilon, ex\rangle$ in at most $2k$ steps is to get from $\langle b, en\rangle$ to $\langle b, ex'\rangle$ in at most $k-1$ steps, and then to get from $\langle \epsilon, (b, ex')\rangle$ to $\langle \epsilon, ex\rangle$ in at most $k - 1$ steps. Thus $\mathbf{q}^{k-1}_{en} \cdot \mathbf{q}^{k-1}_{(b,ex')} \leq \mathbf{q}^{2k}_u$.

Now, by the inductive assumption, $\mathbf{q}^{2^k} \geq \mathbf{x}^{k+1} \geq \mathbf{q}^k$. Hence, using the inequality, and substituting, we get

$$\mathbf{q}_u^{k+1} \leq \mathbf{x}_{en}^{k+1}\, \mathbf{x}_{(b,ex')}^{k+1} = P(\mathbf{x}^{k+1})_u = \mathbf{x}_u^{k+2}.$$

We also get

$$\mathbf{x}_u^{k+2} = \mathbf{x}_{en}^{k+1}\, \mathbf{x}_{(b,ex')}^{k+1} \leq \mathbf{q}_{ex}^{2^k} \mathbf{q}_{(b,ex')}^{2^k} \leq \mathbf{q}_{(b,ex')}^{2^{k+1}}.$$

(d) $u \in Type_{max}$: In this case, it is easy to see that $\mathbf{q}_u^{k+1} = \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^k$. Thus, by inductive hypothesis, $\mathbf{q}_u^{k+1} = \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^k \leq \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{x}_v^{k+1} = \mathbf{x}_u^{k+2}$. Likewise, $\mathbf{x}_u^{k+2} = \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{x}_v^{k+1} \leq \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^{2^k} = \mathbf{q}_u^{2^k+1} \leq \mathbf{q}_u^{2^{k+1}}$.

(e) $u \in Type_{min}$: As in the previous case, $\mathbf{q}_u^{k+1} = \min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^k \leq \min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{x}_v^{k+1} = \mathbf{x}_u^{k+2}$. Again, like the max case, $\mathbf{x}_u^{k+2} \leq \mathbf{q}_u^{2^{k+1}}$.

We have established assertion (2).

(3) Assertion (3) follows from the definition of $\mathbf{q}^*$. Suppose $\mathbf{q}^* \neq P(\mathbf{q}^*)$. The vector $\mathbf{q}^*$ clearly satisfies the equations for vertices u of type $exit, rand, call$. Thus, the only possibility is that $\mathbf{q}_u^* \neq P_u(\mathbf{q}^*)$ for some vertex u of type max or min.
Suppose u is of type max. Then, clearly, $\mathbf{q}_u^* \geq \mathbf{q}_v^*$ for any neighbor of u, with $(u, \perp, v) \in \delta$, because if $\mathbf{q}_u^* < \mathbf{q}_v^*$, then player 1 could play the transition $(u, \perp, v)$ at the beginning of the game M_A starting at u and improve its payoff. Likewise, $\mathbf{q}_u^* \leq \mathbf{q}_v^*$, for some neighbor v, because otherwise, no matter what initial move player 1 makes from u, its payoff would be less than the purported $\mathbf{q}_u^*$. Similarly, suppose u is of Type min. Then, again, $\mathbf{q}_u^* \leq \mathbf{q}_v^*$ for any neighbor of u, with $(u, \perp, v) \in \delta$, because if $\mathbf{q}_u^* > \mathbf{q}_v^*$, then player 2 can switch to a strategy which, starting at u, moves initially to v, and then regardless of how player 1 plays, player 2 would have a strategy to limit the payoff to $\mathbf{q}_v^* < \mathbf{q}_u^*$, a contradiction. Likewise, $\mathbf{q}_u^* \geq \mathbf{q}_v^*$, for some neighbor v, because otherwise, no matter what initial move player 2 makes from u, player 1 can play in such a way that, no matter what player 2 does, player 1's ultimate payoff would be strictly greater than the purported $\mathbf{q}_u^*$. Hence $\mathbf{q}^*$ is a fixed-point of P.

(4) Note that P is monotonic, and that $\mathbf{q}^*$ is a fixed-point of P. Since $\mathbf{x}^0 = 0 \leq \mathbf{q}^*$, it follows, by induction on $k \geq 0$, that $\mathbf{x}^k \leq \mathbf{q}^*$, for all $k \geq 0$.

(5) Consider any fixpoint $\mathbf{q}'$ of the equations, i.e., where $\mathbf{q}' = P(\mathbf{q}')$. We shall argue that $\mathbf{q}^* \leq \mathbf{q}'$. Let τ' be the (stationary) strategy for player 2 that always picks, at any state $\langle \beta, u \rangle$, for vertex $u \in \mathtt{pl}^{-1}(2)$, the particular successor v of u such that $v = \arg\min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}'_v$ (breaking ties, say, lexicographically).

LEMMA 3.3. *For all strategies $\sigma \in \Psi_1$ of player 1, and for all $k \geq 0$, $\mathbf{q}^{k,\sigma,\tau'} \leq \mathbf{q}'$.*

PROOF. By induction, similar to the proof of assertion (2). The base case $\mathbf{q}^{0,\sigma,\tau'} \leq \mathbf{q}'$ is trivial. For the induction step, consider a vertex u. We distinguish 5 cases depending on the type of u.

(a) Type $exit$. If $u \in Ex$, then for all $j \geq 0$, clearly $\mathbf{q}_u^{j,\sigma,\tau'} = \mathbf{q}'_u = 1$.

(b) Type $rand$. Let σ' be the strategy defined by $\sigma'(\beta) = \sigma(\langle \epsilon, u \rangle \beta)$ for all $\beta \in V^*$. Then,

$$\mathbf{q}_u^{k+1,\sigma,\tau'} = \sum_v p_{u,v}\, \mathbf{q}_v^{k,\sigma',\tau'} \leq \sum_v p_{u,v}\, \mathbf{q}_v^{'} = \mathbf{q}_u^{'}.$$

(c) Type $call$. In this case, $u = (b, en) \in \mathit{Call}_b$. Let (b, ex') be the return port of box b, i.e., ex' is the unique exit node of the component $A_{Y(b)}$ assigned to b.

Then $\mathbf{q}_u^{k+1,\sigma,\tau'} \leq sup_\rho \mathbf{q}_{en}^{k-1,\rho,\tau'} \cdot sup_\rho \mathbf{q}_{(b,ex')}^{k-1,\rho,\tau'}$. Now, by the inductive assumption, $\mathbf{q}^{k-1,\rho,\tau'} \leq \mathbf{q}'$ for all ρ. Moreover, since $\mathbf{q}' = P(\mathbf{q}')$, $\mathbf{q}_u^{'} = \mathbf{q}_{en}^{'} \cdot \mathbf{q}_{(b,ex')}^{'}$. Hence, using these inequalities and substituting, we get

$$\mathbf{q}_u^{k+1,\sigma,\tau'} \leq \mathbf{q}_{en}^{'}\, \mathbf{q}_{(b,ex')}^{'} = \mathbf{q}_u^{'}.$$

(d) Type max: In this case, starting at $\langle \epsilon, u \rangle$, whatever player 1's strategy σ is, initially it has to move to some neighbor $\langle \epsilon, v \rangle$ from which the probability of termination in at most k steps is precisely $\mathbf{q}_v^{k,\sigma',\tau'}$, where σ' is defined as in case (b). Thus $\mathbf{q}_u^{k+1,\sigma,\tau'} \leq \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^{k,\sigma',\tau'}$. By the inductive hypothesis $\mathbf{q}_v^{k,\sigma',\tau'} \leq \mathbf{q}_v^{'}$ for all v. Thus, $\mathbf{q}_u^{k+1,\sigma,\tau'} \leq \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^{'} = \mathbf{q}_u^{'}$.

(e) Type min: Since $\mathbf{q}' = P(\mathbf{q}')$, we know that $\mathbf{q}'_u = \min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}_v^{'}$. We also know that $\tau'(u) = v$, where $v = \arg\min_{\{v|(u,\perp,v)\in\delta\}} q_v^{'}$. But then, by the inductive hypothesis, $\mathbf{q}_u^{k+1,\sigma,\tau'} = \mathbf{q}_v^{k,\sigma',\tau'} \leq \mathbf{q}'_v = \min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}'_v = \mathbf{q}'_u$.

□

Now, by the lemma, $\mathbf{q}^{*,\sigma,\tau'} = \lim_{k\to\infty} \mathbf{q}^{k,\sigma,\tau'} \leq \mathbf{q}'$. This holds for any strategy $\sigma \in \Psi_1$. Therefore, $\sup_{\sigma\in\Psi_1} q_u^{*,\sigma,\tau'} \leq \mathbf{q}'_u$, for every vertex u. Thus, $\mathbf{q}^*_u = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} q_u^{*,\sigma,\tau} \leq \sup_{\sigma\in\Psi_1} q_u^{*,\sigma,\tau'} \leq \mathbf{q}'_u$, for all vertices u. In other words, $\mathbf{q}^* \leq \mathbf{q}^{'}$.

(6) Finally, observe that $\lim_{k\to\infty} \mathbf{x}^k$ exists and is bounded within $[0,1]^n$. The sequence x^k, $k \to \infty$ is monotonically non-decreasing, and by definition $\lim_{k\to\infty} \mathbf{x}^k$ is a fixed point of $\mathbf{x} = P(\mathbf{x})$. By part (4), $\lim_{k\to\infty} \mathbf{x}^k \leq \mathbf{q}^*$. Thus, by part (5), $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{q}^*$. On the other hand, since $\mathbf{q}^k \leq \mathbf{x}^{k+1} \leq \mathbf{q}^{2^k}$, we have $\lim_{k\to\infty} \mathbf{x}^k = \lim_{k\to\infty} \mathbf{q}^k$.

□

Example 3.4.
Repeated iteration of the operator P of the system of equations in Example 3.1 starting from 0 converges to the LFP q^* of this system, which is equal to the termination game values of the 1-RSSG of Figure 2, and is as follows: $x_s = 0.75$, $x_{u_1} = 0.875$, $x_{u_2} = 0.75$, $x_{u_3} = 0.875$, $x_{u_4} = 0.75$, $x_{u_5} = 0$, $x_{(b_1,s)} = 0.5625$, $x_{(b_1,t)} = 0.75$, $x_{(b_2,s)} = 0.75$, $x_{(b_2,t)} = 1$, $x_t = 1$. □

4. STACKLESS AND MEMORYLESS DETERMINACY FOR 1-RSSGS

We now identify a very restricted kind of strategy that suffices as an optimal strategy in 1-RSSGs. Recall that a strategy is called a (deterministic) *Stackless & Memoryless (SM)* strategy 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 (and furthermore the strategy is deterministic, i.e., does not use any randomization). Such a strategy can be given simply by a function that maps every vertex associated with that player to one of its neighbors. For example, in the 1-RSSG of Figure 2, an optimal strategy for the max player is to select at vertex u_1 always the neighboring vertex u_3, and an optimal strategy for the min player is to move from vertex u_4 always to the call port (b_2, s) of box b.

COROLLARY 4.1. *In every 1-RSSG termination game, player 2 (the minimizer) has an optimal (deterministic) SM strategy.*

PROOF. Consider the strategy τ' in the proof of part (5) of Theorem 3.2, 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 SM strategy and thus the game is *SM-determined*, meaning both players have optimal SM strategies. (Note that the game is not symmetric with respect to the two players.)[4]

For this proof, we will introduce a new flexible technique for showing stackless and memoryless determinacy, based on a strategy improvement argument, which at the same time provides a "strategy improvement method" for solving the games, if one were able to solve the 1-player 1-RMDP games efficiently. Our proof technique relies on parametrizing the game with respect to the game value at a designated vertex, and viewing the game value starting at other vertices as a function of the value at the designated vertex. These functions are definable as infima over certain power series which have very special analytic properties. By exploiting these analytic properties, and properties of the fixed points of such functions, we are able to deduce via a strategy improvement argument that the maximizing player must have an optimal SM strategy, and thus that these games are SM determined. This proof technique is quite flexible (e.g., one can do the parametrization with respect to more than one vertex at a time), and can also be used as an alternative method to prove older known results about memoryless determinacy and strategy improvement for classes of finite-state stochastic games.

THEOREM 4.2. *Every 1-RSSG termination game is SM-determined. Moreover, there is a (deterministic) SM strategy $\sigma^* \in \Psi_1$ that maximizes the value of $q_u^{*,\sigma}$ for all u, and likewise a (deterministic) SM strategy $\tau^* \in \Psi_2$ that minimizes the value of $q_u^{*,\tau}$ for all u.*

PROOF. By Corollary 4.1, we only need to show that player 1 has an optimal SM strategy. Let σ be any SM strategy for player 1. Consider $\mathbf{q}^{*,\sigma} = \inf_{\tau \in \Psi_2} \mathbf{q}^{*,\sigma,\tau}$ (where the infimum is applied independently for each component). First, let us note that if $\mathbf{q}^{*,\sigma} = P(\mathbf{q}^{*,\sigma})$ then $\mathbf{q}^{*,\sigma} = \mathbf{q}^*$. This is so because, by Theorem 3.2, $\mathbf{q}^* \leq \mathbf{q}^{*,\sigma}$, and on the other hand, σ is just one strategy for player 1, and for every vertex u, $\mathbf{q}_u^* = \sup_{\sigma' \in \Psi_1} \mathbf{q}_u^{*,\sigma'} \geq \mathbf{q}_u^{*,\sigma}$. Now, we claim that, for all vertices u that do not belong to player 1 (i.e., such that u is not of $\text{Type}_{\max}$) $\mathbf{q}_u^{*,\sigma}$ satisfies its equation in $\mathbf{x} = P(\mathbf{x})$. In other words, $\mathbf{q}_u^{*,\sigma} = P_u(\mathbf{q}^{*,\sigma})$. To see this, note that for vertices u of types $\{exit, rand, call\}$, no choice of either player is involved, thus the equation holds by definition of $\mathbf{q}^{*,\sigma}$. For vertices of $\text{Type}_{\min}$, which belong to player 2 (the minimizer), we have the equation $\mathbf{x}_u = \min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{x}_v$. But note that the best player 2 can do against strategy σ, starting at $\langle \epsilon, u \rangle$, is to move to a neighboring vertex v such that $v = \arg\min_{\{v|(u,\perp,v)\in\delta\}} \mathbf{q}^{*,\sigma}_{(v,ex)}$.

Thus, the only equations that may fail are those of $\text{Type}_{\max}$, of the form $\mathbf{x}_u = \max_{\{v|(u,\perp,v)\in\delta\}} \mathbf{x}_v$. Suppose $\sigma(u) = v$, for some neighbor v. Clearly then, $\mathbf{q}_u^{*,\sigma} = \mathbf{q}_v^{*,\sigma}$. Thus, $\mathbf{q}_u^{*,\sigma} \leq \max_{\{v'|(u,\perp,v')\in\delta\}} \mathbf{q}_{v'}^{*,\sigma}$. Thus equality fails iff there is another vertex $w \neq v$, with $(u, \perp, w) \in \delta$, such that $\mathbf{q}_v^{*,\sigma} < \mathbf{q}_w^{*,\sigma}$. Consider such a vertex w, and consider now a revised SM strategy σ', which is identical to σ, except that $\sigma'(u) = w$.

Next, consider a parametrized 1-RSSG, $A(t)$, which is identical to A, except that all edges out of vertex u are removed, and replaced by a single edge labeled by probability

[4]In the subsequent work [Etessami and Yannakakis 2008], we established analogous results to Corollary 4.1 and Theorem 4.2 for the more general class of 1-exit recursive concurrent stochastic games (1-RCSGs), by explicitly using the results here and extending them. For 1-RCSGs the nature of the results changes substantially. Specifically, there are no *deterministic* optimal strategies in general, and while player 2 (the minimizer) has *randomized* SM optimal strategies, player 1 (the maximizer) may only have ϵ-optimal randomized SM strategies. Note that the proof of the analogous result to Theorem 4.2, namely Theorem 4.2 in [Etessami and Yannakakis 2008], makes explicit use of the proofs of this section (see in particular the proof of Lemma 4.4 in [Etessami and Yannakakis 2008], which employs results established in this section).

variable t to the exit of the same component and an edge with probability $1-t$ going to a dead absorbing state. Fixing the value t determines an RSSG, $A(t)$.

Note that if we restrict the SM strategies σ or σ' to all vertices other than u, then they both define the same SM strategy for the RSSG $A(t)$. For every vertex z, define $q_z^{*,\sigma,\tau,t}$ to be the probability that the Markov chain $M_{A(t)}^{z,\sigma,\tau}$. starting from $\langle \epsilon, z\rangle$ eventually terminates, i.e. reaches the state $\langle \epsilon, ex\rangle$ where ex is the exit of the component of z. Now, for each vertex z, define the function $f_z(t) = \inf_{\tau\in\Psi_2} q_z^{*,\sigma,\tau,t}$. In other words, $f_z(t)$ is the infimum over all strategies of player 2, of the probability of termination in $A(t)$ starting from z where player 1 uses strategy σ. This probability is parametrized by t.

Let $t_1 = \mathbf{q}_u^{*,\sigma}$. We first observe that $f_z(t_1) = \mathbf{q}_z^{*,\sigma}$ for every z. This is so because, any strategy for minimizing the probability of terminating from z (with ϵ context) would, upon hitting a state $\langle \beta, u\rangle$, be best off minimizing the probability of exiting from the last recursive call starting at u (i.e., reaching $\langle \beta, ex\rangle$ where ex is the exit of the component of u), regardless of the context β, for, without exiting from this last call, it could not hope to terminate in the empty context.

Note that, by Corollary 4.1, in the 1-RSSG termination game on $A(t)$, for any value of t, and any start vertex z, player 2 has an optimal SM strategy $\tau_{z,t}$, such that $\tau_{z,t} = \arg\min_{\tau\in\Psi_2} q_z^{*,\sigma,\tau}$. Let $g_{(z,\tau)}(t) = q_z^{*,\sigma,\tau}$. Note that $f_z(t) = \min_\tau g_{z,\tau}(t)$, where the minimum is over SM strategies. Now, note that the function $g_{z,\tau}(t)$ is the probability of reaching an exit in a Recursive Markov Chain starting from a particular vertex. Thus, by [Etessami and Yannakakis 2009], we can define a polynomial system $\mathbf{x} = R(\mathbf{x})$ of equations with non-negative coefficients whose least fixed point is the vector of termination probabilities of the RMC, and $g_{z,\tau}(t) = (\lim_{k\to\infty} R^k(\mathbf{0}))_z$. The parameter t appears as one of the coefficients of the system $\mathbf{x} = R(\mathbf{x})$. The limit $(\lim_{k\to\infty} R^k(\mathbf{0}))_z$ can be described by a power series in the variable t with non-negative coefficients. Therefore, $g_{z,\tau}(t)$ has the following properties: it is a continuous, differentiable, and non-decreasing function of $t \in [0,1]$, with continuous and non-decreasing derivative, $g'_{z,\tau}(t)$ and since the limit defines probabilities we also know that for $t \in [0,1]$, $g_{z,\tau}(t) \in [0,1]$. Thus $g_{z,\tau}(0) \geq 0$ and $g_{z,\tau}(1) \leq 1$.

LEMMA 4.3.

(1) If $g_{z,\tau}(t) > t$ for some $t \in [0,1]$ then $g_{z,\tau}(t') > t'$ for all $t' \in [0,t]$.
(2) If $g_{z,\tau}(t) < t$ for some $t \in [0,1]$ then $g_{z,\tau}(t') < t'$ for all $t' \in [t,1)$.
(3) If $f_z(t) > t$ for some $t \in [0,1]$ then $f_z(t') > t'$ for all $t' \in [0,t]$.
(4) If $f_z(t) < t$ for some $t \in [0,1]$ then $f_z(t') < t'$ for all $t' \in [t,1)$.

PROOF.

1. Assume $g_{z,\tau}(t) > t$. Suppose that $g'_{z,\tau}(t) \geq 1$. Since $g'_{z,\tau}$ is a non-decreasing function, this implies that for all $t'' > t$, we have $g'_{z,\tau}(t'') \geq 1$ and hence $g_{z,\tau}(t'') > t''$. This contradicts the fact that $g_{z,\tau}(1) = 1$. Therefore, $g'_{z,\tau}(t) < 1$. Since $g'_{z,\tau}$ is non-decreasing, this implies that for all $t' \leq t$ we have $g'_{z,\tau}(t') < 1$, i.e. $g_{z,\tau}(t') - t'$ is a decreasing function in $[0,t]$, and since $g_{z,\tau}(t) - t > 0$ it follows that $g_{z,\tau}(t') - t' > 0$ for all $t' \in [0,t]$.

2. Assume $g_{z,\tau}(t) < t$. If for some $t'' > t$, $t'' < 1$, $g_{z,\tau}(t'') \geq t''$, then since $g'_{z,\tau}$ is non-decreasing and $g_{z,\tau}(t) < t$, it must be the case that $g'_{z,\tau}(t'') > 1$. But then (as in part 1), this implies that $g_{z,\tau}(1) > 1$, which is a contradiction.

3. Assume $f_z(t) > t$ at some point $t \in [0,1]$. Then $g_{z,\tau}(t) > t$ for all τ, and hence, by part 1, for all $t' < t$ and for all τ, we have $g_{z,\tau}(t') > t'$. Therefore also $f_z(t') > t'$ for all $t' \in [0,t]$.

4. Assume $f_z(t) < t$ at $t \in [0,1]$. Then there must be some τ' such that $g_{z,\tau'}(t) < t$. Hence $g_{z,\tau'}(t'') < t''$, for all $t'' \in [t,1)$, and hence also $f_z(t'') < t''$ for all $t'' \in [t,1)$. □

Recall that $\sigma(u) = v$, there is another neighbor w of u such that $\mathbf{q}_v^{*,\sigma} < \mathbf{q}_w^{*,\sigma}$, and from the strategy σ we defined a revised SM strategy σ', which is identical to σ, except that $\sigma'(u) = w$. Recall also that $t_1 = q_u^{*,\sigma} = q_v^{*,\sigma}$ and that $f_z(t_1) = q_z^{*,\sigma}$ for every vertex z. Let $t_2 = q_w^{*,\sigma}$. We know $t_2 = q_w^{*,\sigma} = f_w(t_1) > t_1 = q_v^{*,\sigma}$. Therefore $f_w(t) > t$ for all $t < t_1$ by Lemma 4.3. Also, $f_w(t) > t$ for all $t \in [t_1, t_2)$, because $f_w(t_1) = t_2$ and f_w is non-decreasing. Therefore, the least fixed point (i.e., least solution) of $f_w(t) = t$ is $\geq t_2$. Now if we switch strategy σ to σ', where $\sigma'(u) = w$, then $q_w^{*,\sigma'}$ is a fixed point, t_3, of $f_w(t) = t$, so $t_3 \geq t_2 > t_1$ and $q_z^{*,\sigma'} = f_z(t_3) \geq q_z^{*,\sigma}$ for all z, with strict inequality for u: $q_u^{*,\sigma'} = t_3 > q_u^{*,\sigma} = t_1$. Thus, switching to the new SM strategy σ', we get $\mathbf{q}^{*,\sigma'}$ which dominates $\mathbf{q}^{*,\sigma}$, and is strictly greater in some coordinate. But there are only a finite number of SM strategies, thus repeating this process we must eventually get to SM strategy σ^* that can't be improved in this way. Thus $\mathbf{q}^{*,\sigma^*} = P(\mathbf{q}^{*,\sigma^*})$, and hence by our earlier claim $\mathbf{q}^{*,\sigma^*} = \mathbf{q}^*$. Thus, player 1 has an optimal SM strategy. □

5. QUANTITATIVE TERMINATION PROBLEMS FOR 1-RMDPS & 1-RSSGS

We show that quantitative termination problems for 1-RMDPs and 1-RSSGs can be solved in PSPACE by appealing to algorithms for deciding the *existential theory of the reals*. 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: $f_i(\mathbf{x})\theta 0$, where f_i, $i = 1, \ldots, m$, are multi-variate polynomials with rational coefficients over the variables $\mathbf{x} = x_1, \ldots, x_n$, and where θ is any comparison operator among $=, \neq, \geq, \leq, <, >$. The fragment that we will be concerned with is the existential theory of reals, which we refer to as $\exists\mathbf{Th}(\mathbb{R})$. This fragment consists of the true sentences (in prenex form) of the form: $\exists x_1, \ldots, x_n R(x_1, \ldots, x_n)$, where R is a boolean combination of "atomic predicates". The decidability and complexity of the first-order theory of reals and its fragments like $\exists\mathbf{Th}(\mathbb{R})$ has been deeply investigated going back to Tarski. In the current state of the art, it is known that $\exists\mathbf{Th}(\mathbb{R})$ can be decided in PSPACE and furthermore in exponential time where the exponent depends (linearly) only on the number of variables [Canny 1988; Renegar 1992]; hence for a fixed number of variables the time is polynomial.

For a 1-RSSG whose corresponding system of nonlinear min-max equations is $x = P(x)$, first let us consider how to write predicates of the form $x_i = P_i(x)$ as quantifier-free predicates in the theory of reals. For Type $\{exit, rand, call\}$ nodes, $P_i(x)$ is just a polynomial, so it is obvious how to do this. It is also easy to encode, with arithmetic using inequalities, the predicates "$x_i = P_i(x)$" in the cases (Type$_{\max}$) where $P_i(x)$ has the form $\max_{j \in J} x_j$, and (Type$_{\min}$) where it has the form $\min_{j \in J} x_j$, for some subset $J \subseteq \{1, \ldots, n\}$. Namely, 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$_{\min}$ 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 of the form $x_i = P_i(x)$ as a Boolean combination of quantifier-free predicates in the theory of reals.

Now suppose we want to know the relationship between the termination game value q_u^* and some $c \in [0, 1]$, say, we want to know whether $q_u^* \leq c$. Consider the existential sentence:

$$\varphi \equiv \exists x_1, \ldots, x_n (\bigwedge_{i=1}^{n} (x_i = P_i(x))) \wedge (\bigwedge_{i=1}^{n} (0 \leq x_i \leq 1)) \wedge (x_u \leq c)$$

Clearly φ holds precisely when there exists a fixed point $q' \in [0,1]^n$, such that $q' = P(q')$, and $q'_u \leq c$. Since $\mathbf{q}^* \in [0,1]^n$ is the least such fixed point, clearly φ holds iff $q^*_u \leq c$. If we wanted instead to check whether $q^*_u \geq c$, we could instead query a slight modification of φ, replacing in it the predicate $q^*_u \leq c$ with $q^*_u < c$. In this case $q^*_u \geq c$ iff the revised $\exists\mathbf{Th}(\mathbb{R})$ sentence is false. Similarly, we can answer whether $q^*_u \theta c$ holds for any comparison operator $\theta \in \{\leq, \geq, =, \neq, >, <\}$, by using $\exists\mathbf{Th}(\mathbb{R})$ queries.

Furthermore, if we wish to approximate q^*_u to within a given number of bits of precision, we can do so easily by a "binary search". Namely, we start knowing that $0 \leq q^*_u \leq 1$. If we already know, for some a and b, that $a \leq q^*_u \leq b$, we do one additional query, to decide whether $a \leq q^*_u \leq (a+b)/2$, by substituting $(a+b)/2$ for c in φ. If the sentence is true then we know that $a \leq q^*_u \leq (a+b)/2$ and continue the binary search in the interval $[a, (a+b)/2]$, otherwise we continue in the interval $((a+b)/2, b]$. In this way, with i queries we can approximate q^*_u to within i bits of precision. More precisely, we can compute values $a, b \in [0,1]$ such that $b - a = 1/2^i$ and $q^*_u \in [a,b]$. This discussion yields:

THEOREM 5.1. (Quantitative termination problems for 1-RSSGs are in PSPACE) *Given a 1-RSSG A, vertex u and a rational probability p, there is a PSPACE algorithm to decide whether $q^*_u \leq p$ (or $q^*_u \geq p$, or $q^*_u < p$, etc.). The running time of the algorithm is $O(|A|^{O(n)})$ where n is the number of vertices of the 1-RSSG (i.e., the number of variables in $\mathbf{x} = P(\mathbf{x})$). Hence the running time is polynomial if n is bounded by a fixed constant.*

Furthermore, we can approximate the vector $\mathbf{q}^*$ of values to within a specified number of bits i of precision (i given in unary), in PSPACE and in time $O(i|A|^{O(n)})$.[5]

A natural question is whether these PSPACE upper bounds can be improved upon. Unfortunately, we showed in [Etessami and Yannakakis 2009] that already for 1-RMCs, i.e., in the purely probabilistic setting without any players, deciding whether the probability of termination is at least a given rational $p \in (0,1)$ is as hard as several long standing open problems in the complexity of numerical computation. We now recall these problems. The SQRT-SUM problem asks the following: given a list of natural numbers $\langle d_1, \ldots, d_n, k \rangle \in \mathbb{N}^{n+1}$, decide whether $(\sum_{i=1}^n \sqrt{d_i}) \leq k$. This problem arises in many settings, e.g., in geometric computation where one often wishes to compare sums of distances in Euclidean space. It has been open since the 1970's whether the problem is even contained in NP [Garey et al. 1976]. Another difficult problem, which is known to be harder than SQRT-SUM (via P-time Turing reduction) is the PosSLP problem: given an arithmetic circuit (straight line program) over the basis $\{+, *, -\}$, with integer inputs, decide whether the output of the circuit is > 0. PosSLP captures everything that is computable in polynomial time in the unit-cost arithmetic RAM model of computation with discrete inputs (in the sense that PosSLP is hard for this class via P-time Turing reductions). The best current upper bounds known for solving SQRT-SUM and PosSLP are in the 4th level "counting hierarchy" [Allender et al. 2009], i.e. just (slightly) below PSPACE, but it remains a major open problem to place these problems even in NP.

THEOREM 5.2. *([Etessami and Yannakakis 2009]) The SQRT-SUM and PosSLP problems are polynomial time (many-one) reducible to the quantitative termination problem for 1-RMCs, i.e., decide whether the probability of termination starting at a given vertex is at least a given $p \in (0,1)$, (Thus, trivially, these problems are also reducible to the quantitative termination problems for 1-RMDPs and 1-RSSGs.)*

[5]This PSPACE upper bound was subsequently generalized by us in [Etessami and Yannakakis 2008] to the setting of 1-RCSGs.

Thus, without a major breakthrough, we can not hope to substantially improve on the PSPACE complexity upper bounds for quantitative decision problems for 1-RMDPs and 1-RSSGs. See [Etessami and Yannakakis 2009] for much more information on these numerical computation problems and the implications of these relative hardness results.

The special case of $p = 0$ in the quantitative problem, i.e. determining those vertices u whose corresponding value $q^*_u = 0$, is much simpler and can be solved easily in polynomial time.

THEOREM 5.3. (Value 0 for 1-RSSGs in P-time) *Given a 1-RSSG A we can compute in polynomial time the set of vertices u whose corresponding value $q^*_u = 0$. This set does not depend on the actual probabilities of the transitions but only on the transition structure. These results hold in particular also for maximizing and minimizing 1-RMDPs.*[6]

PROOF. Given a 1-RSSG A, we will compute the set T of vertices u for which $q^*_u = 0$. We will compute the complementary set S of vertices u such that $q^*_u > 0$ as follows.
Initialize set $S := Ex$.
Repeat the following until there is no change in the set S:

— If a probabilistic vertex or max vertex u has a successor in S then add u to S.
— If a min vertex u has all successors in S then add u to S.
— If $u = (b, en)$ is a call port of box b and both en (the entry of the corresponding component) and the return port (b, ex') of the box are in S then add u to S.

When the process finishes, we let $T = Q - S$. This is the set of vertices that have value 0.

We can view the algorithm as essentially following the repeated application of the operator P of the min-max linear equations, starting from the $\mathbf{0}$ vector, but keeping only track of the set S of vertices that have nonzero value, and stopping when this set does not change; this will happen after at most n iterations. At the end, we let T be the set $Q - S$ of remaining vertices which have still value 0. Player 2's strategy is to pick for each vertex $u \in T$ of type min a successor that is also in T. If we remove all other edges out of min-vertices in T, then all successors of all vertices in T are also in T, and T does not contain any exit nodes. Therefore, all the vertices of T have no way of reaching their exits and hence their value is 0. All other vertices have a positive value, thus the algorithm outputs the correct set T. □

The case of value $p = 1$ for the qualitative termination problem, is much more involved. In the next section we address this problem for both maximizing and minimizing 1-RMDPs, and in the following section we discuss 1-RSSGs.

6. QUALITATIVE TERMINATION FOR 1-RMDPS IN P-TIME

We show that, for both maximizing 1-RMDPs and minimizing 1-RMDPs, qualitative termination can be decided in polynomial time. Please note that the two cases are not symmetric. We provide distinct algorithms for each of them. An important result that we shall make use of is this:

THEOREM 6.1. *([Etessami and Yannakakis 2009]) The qualitative termination problem for 1-RMCs is decidable in polynomial time.*

[6]A more general result, a P-time upper bounds for the Value 0 problem for 1-RCSGs, was established by us subsequently in [Etessami and Yannakakis 2008]. We provide the proof for 1-RSSGs here for completeness. (The proof for 1-RCSGs in [Etessami and Yannakakis 2008] is somewhat more terse, because it refers to this proof and assumes it as given.)

We summarize the key elements of that algorithm (please see [Etessami and Yannakakis 2009] for more details). The algorithm employs graph-theoretic processing and a spectral radius characterization of *moment matrices* associated with 1-RMCs. Given a 1-RMC A, we can construct a system of polynomial equations, $x = P(x)$, similar to the system for 1-RSSGs, except that there are no min and max equations in this case. The moment matrix B of a system $x = P(x)$ is the square Jacobian matrix of $P(x)$ (i.e., the matrix whose (i, j)'th entry is the partial derivative $\partial P_i(x)/\partial x_j$) evaluated at the all 1 vector (i.e., $x_u \leftarrow 1$ for $u \in Q$). That is, if i is a probabilistic vertex (i.e., $i \in Type_{rand}$) then $B[i, j]$ is the transition probability p_{ij} for every j; if i is a call port $i = (b, en)$ then $B[i, j]$ is 0 for all j except for the entry en of the component $A_{Y(b)}$ assigned to box b and for the return port (b, ex) of b, for which $B[i, j] = 1$. The spectral radius (largest eigenvalue) of matrix B is denoted $\rho(B)$. From the system $x = P(x)$, we construct a *dependency graph* G_A, which has one node u for each variable x_u, i.e. for each vertex u of the 1-RMC, and has a (directed) edge $u \to v$ if x_v appears in $P_u(x)$. We first compute in polynomial time the set Z_0 of vertices u whose value q^*_u is 0 (by an algorithm similar to that of Theorem 5.3), and eliminate them from the graph G_A. We can determine whether another vertex $u \notin Z_0$ has value $q^*_u < 1$ or $= 1$ as follows: Compute the set R of nodes reachable from u in G_A, let B_R be the moment matrix of the equation system restricted to the variables and equations of vertices in R, and compute its spectral radius $\rho(B_R)$. If $\rho(B_R) > 1$ then $q^*_u < 1$ and if $\rho(B_R) \leq 1$ then $q^*_u = 1$.

We can in fact classify all the vertices together in one pass, rather than testing each vertex individually, as follows. After computing the set Z_0 of vertices with value 0 and eliminating them from the dependency graph G_A, we decompose G_A into strongly connected components (SCCs), topologically sort the SCCs, and process them bottom-up, assigning to each SCC either the value 1, meaning that all vertices in the SCC have value 1, or we mark it \$, which means that all the vertices of the SCC have value strictly between 0 and 1. This is done as follows. After we removed the 0-valued vertices, the only bottom SCCs of G_A correspond to exit nodes and get value 1. We then process the remaining SCCs bottom-up. Iteratively, suppose that C is a lowest unprocessed SCC. If C has any successor node that is marked 0 or \$, then we mark C also \$. Otherwise we restrict the equation system $x = P(x)$ to the SCC C, we compute the corresponding moment matrix B_C and compare its spectral radius, $\rho(B_C)$, to 1; if $\rho(B_C) \leq 1$ then C is marked 1, otherwise it is marked \$. The tests $\rho(B_C) \leq 1$ can be done using linear programming ([Etessami and Yannakakis 2009]), or even by solving linear systems of equations [Esparza et al. 2013] (using the latter method yields a strongly polynomial algorithm).

6.1. Maximizing 1-RMDPs

We present a polynomial time algorithm for maximizing 1-RMDPs.

THEOREM 6.2. *Given a maximizing 1-RMDP A, there is a polynomial time algorithm which determines for every vertex u of A which of the following three cases holds: (1) $q^*_u = 0$, or (2) $q^*_u = 1$, or (3) $0 < q^*_u < 1$.*

PROOF. Given a maximizing 1-RMDP, A, we shall determine for all vertices u, whether $q^*_u = 1$, $q^*_u = 0$, or $0 < q^*_u < 1$. From the system of equations $x = P(x)$ for A we construct a labeled *dependency graph*, $G_A = (Q, \to)$, as follows: the nodes Q of G_A are the vertices of A, and there is an edge $u \to v$ iff x_v appears on the right hand side of the equation $x_u = P_u(x)$. Each node u is labeled by its *Type*. If $u \in Type_{rand}$, i.e., u is a probabilistic vertex, and x_v appears in the weighted sum $P_u(x)$ as a term

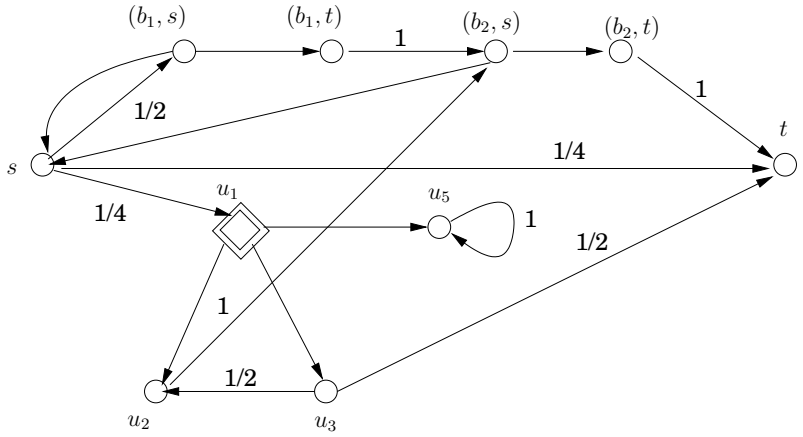

Fig. 4. Dependency graph

$p_{u,v}x_v$, then the edge from u to v is labeled by the probability $p_{u,v}$. Otherwise, the edge is unlabeled.

For example, let M be the maximizing 1-RMDP that is obtained from the 1-RSSG of Figure 2 by removing the type-min vertex u_4. The dependency graph of M is shown in Figure 4.

We wish to partition the nodes of the dependency graph into three classes: $Z_0 = \{u \mid q^*_u = 0\}$, $Z_1 = \{u \mid q^*_u = 1\}$, and $Z_\$ = \{u \mid 0 < q^*_u < 1\}$. In our algorithm we will use a fourth partition, $Z_?$, to denote those nodes for which we have not yet determined to which partition they belong. We first compute Z_0. By Theorem 5.3, this can be done easily in P-time. Once we have computed Z_0, the remaining nodes belong either to Z_1 or $Z_\$$. Clearly, exit nodes belong to Z_1.

Initialize: $Z_1 \leftarrow Ex$; $Z_\$ \leftarrow \emptyset$; and $Z_? \leftarrow Q \setminus (Z_1 \cup Z_0)$;

Next, we do one "preprocessing" step to categorize some remaining "easy" nodes into Z_1 and $Z_\$$, as follows:

repeat
  if $u \in Z_? \cap (Type_{rand} \cup Type_{call})$ has all of its successors in Z_1
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_1 \leftarrow Z_1 \cup \{u\}$;
  if $u \in Z_? \cap Type_{max}$ has some successor in Z_1
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_1 \leftarrow Z_1 \cup \{u\}$;
  if $u \in Z_? \cap (Type_{rand} \cup Type_{call})$ has some successor in $Z_0 \cup Z_\$$
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_\$ \leftarrow Z_\$ \cup \{u\}$;
  if $u \in Z_? \cap Type_{max}$ has all successors in $(Z_0 \cup Z_\$)$
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_\$ \leftarrow Z_\$ \cup \{u\}$;
until (there is no change to $Z_?$)

For example, in the case of the dependency graph of Figure 4, the above preprocessing step will put u_5 in Z_0, vertices t and (b_2, t) in Z_1, and the rest of the vertices in $Z_?$.

The correctness of the assignments in this preprocessing step follows from the fact that $\mathbf{q}^* = P(\mathbf{q}^*)$ by an easy induction, i.e., if a vertex u is added to Z_1 in some step then indeed $q^*_u = 1$, and if a vertex u is added to $Z_\$$ then $0 < q^*_u < 1$.

The preprocessing step will not, in general, empty $Z_?$, and we need to categorize the remaining nodes in $Z_?$. We will construct a set of linear inequalities (an LP without an objective function) which has a solution iff there is any remaining node in $Z_?$ which

belongs in Z_1, and if so, the solution we obtain to the LP will let us find and remove from $Z_?$ some more nodes that belong in Z_1. Note that, if we can do this, then we can solve our problem, because all we need to do is iterate: we repeatedly do a preprocessing step, followed by the LP step to remove nodes from $Z_?$, until no more nodes can be removed, at which point we are done: the remaining nodes in $Z_?$ all belong to $Z_\$$.

For the LP step, restrict attention to the vertices remaining in $Z_?$. These vertices induce a subgraph of G_A, call it G'_A. Call a remaining probabilistic node u in $Z_?$ *leaky* if it does not have full probability on its outgoing transitions inside G'_A. Note that this happens if and only if some of u's outedges in G_A lead to nodes in Z_1 (otherwise, if u had an outedge to a node in Z_0 or $Z_\$$, it would already have been removed from $Z_?$ during preprocessing). Let $\mathcal{L}$ denote the set of remaining leaky nodes in $Z_?$. We add an extra terminal node t to G'_A, and for every $u \in \mathcal{L}$ we add a probabilistic edge $u \stackrel{p_{u,t}}{\rightarrow} t$, where $p_{u,t} = 1 - \sum_{v \in Z_?} p_{u,v}$.

W.l.o.g., we assume for simplicity that all the entries of components and return nodes of boxes are probabilistic nodes. This can easily be assured by minor adjustments to the input 1-RMDP: if such a node u is not probabilistic, then add a new node u' that has the same type and all the edges of u, change u to a probabilistic node and add a probability 1 edge from u to u'.

The LP that we construct has a variable y_i for every node $i \in Z_?$ that is not $Type_{max}$, and has a variable $y_{i,j}$ for every $Type_{max}$ node $i \in Z_?$ and successor $j \in Z_?$ of i. In addition there are "flow" variables $f_{i,j,k}$ for each node $i \in Z_?$, and every edge $j \rightarrow k$ in G'_A. The constraints are as follows.

(1) For every $j \in Type_{rand} \cup Type_{call}$ that is not a component entry or a return:

$$y_j \geq \sum_{i \rightarrow j \ \wedge \ i \in Type_{rand}} p_{i,j} y_i + \sum_{i \rightarrow j \ \wedge \ i \in Type_{max}} y_{i,j}$$

(2) For every $j \in Type_{max}$:

$$\sum_k y_{j,k} \geq \sum_{i \rightarrow j \ \wedge \ i \in Type_{rand}} p_{i,j} y_i + \sum_{i \rightarrow j \ \wedge \ i \in Type_{max}} y_{i,j}$$

(3) For every node i that is an entry of a component, say A_r:

$$y_i \geq \sum_{j=(b,i) \in Type_{call} \ \wedge \ Y(b)=r} y_j$$

(4) For every node i that is a return port, say of box b: $y_i \geq \sum_{j \in Call_b} y_j$:

(5) $\sum_i y_i + \sum_{i,j} y_{i,j} = 1$.

(6) $y \geq 0$.

Regard the dependency graph G'_A (with the extra terminal node t) as a network flow graph with capacity on each edge $i \rightarrow j$ coming out of a max node equal to $y_{i,j}$ and capacity of edges $i \rightarrow j$ coming out of the other vertices equal to y_i. We set up one flow problem for each $i \in Z_?$, with source i, sink t and flow variables f_{ijk}.

7. For every vertex i, we have flow conservation constraints on the variables $f_{i,j,k}$, i.e., $\sum_k f_{i,j,k} = \sum_k f_{i,k,j}$, for all nodes $j \in Z_?$, $j \neq i, t$.
8. Non-negativity constraints: $f_{i,j,k} \geq 0$ for all i, j, k.
9. Capacity constraints: $f_{i,j,k} \leq y_{j,k}$ for every $j \in Type_{max}$ with successor k, and for every node i; and $f_{i,j,k} \leq y_j$ for every $j \in Type_{rand} \cup Type_{call}$ and successor k in G'_A and every node i.
10. Source constraints: $\sum_k f_{i,i,k} = y_i / 2^{2m}$, for every $i \in Type_{rand} \cup Type_{call}$, and $\sum_k f_{i,i,k} = \sum_j y_{i,j} / 2^{2m}$, for $i \in Type_{max}$, where m is defined as follows. Suppose our LP in constraints (1.-6.) has r variables and constraints, and that its rational entries have numerator and denominator with at most l bits. If there is a solution

to (1.-6.), then (see, e.g., [Grötschel et al. 1993]), there is a rational solution whose numerators and denominators require at most $m = poly(r, l)$ bits to encode, where $poly(r, l)$ is a polynomial in r and l. Note $r \in O(|G'_A|)$, l is bounded by the number of bits required for the transition probabilities $p_{u,v}$ in A, hence m is polynomial in the input size.

Before we prove formally that the LP has the desired property, i.e. it has a solution if and only if there are vertices $v \in Z_?$ that have value $q^*_v = 1$, we give informally some intuition for the variables and constraints of the LP. The construction of the LP is based on the qualitative classification of the vertices for 1-RMCs. An optimal strategy σ of the maximizing player induces a 1-RMC. There are some vertices of $Z_?$ that have value 1 iff there is a bottom SCC C in the dependency graph of this 1-RMC that contains a leaky vertex, and whose moment matrix B_C has spectral radius $\rho(B_C) \leq 1$. Since B_C is an irreducible nonnegative matrix, the condition that $\rho(B_C) \leq 1$ is equivalent to the condition that there exists a nonnegative vector $u \neq 0$ that satisfies $u \geq u \cdot B_C$. Indeed, we can take u to be a nonnegative eigenvector associated with the eigenvalue $\rho(B_C)$, and we can assume u has been normalized so that its coordinates sum to 1. The variables $y_i, i \notin Type_{max}$ of the LP stand for the (unknown) values u_i, and the variables $y_{i,j}$ for the vertices $i \in Type_{max}$ and their successors j, stand for the quantities $u_i\sigma(i, j)$, where $\sigma(i, j)$ is the (unknown) probability that the optimal strategy assigns to the edge (i, j). Note that we do not assign separate variables to $u_i, i \in Type_{max}$ and $\sigma(i, j)$, but rather to their product, to keep the constraints linear. (We could have included variables y_i also for $i \in Type_{max}$ to stand for u_i, and added constraints $y_i = \sum_j y_{i,j}$, but they are redundant.) Constraints 1-4 of the LP express the inequality $u \geq u \cdot B_C$. Constraints 5 and 6 correspond to the fact that u is normalized and $u \geq 0$. The purpose of constraints (7-10) is to ensure that every vertex i with a nonzero variable y_i or nonzero edge variable $y_{i,j}$ can reach a leaky vertex in the subgraph of G'_A induced by the support of the y solution vector.

We proceed now to the formal proof.

LEMMA 6.3. *There exists a vertex $v \in Z_?$ such that $q^*_v = 1$ if and only if the LP constraints in (1.–10.) are feasible. Moreover, from a solution to the LP we can find a (partial) strategy for the maximizing player that forces termination from some such v with probability $= 1$.*

PROOF.
($\Rightarrow$) Suppose there exists $v \in Z_?$ with $q^*_v = 1$. Fix an optimal strategy of the maximizing player. By Theorem 4.2, there is an SM strategy σ that maximizes all the values at all the nodes and which picks one successor for each max node. Fix the edges chosen by σ as outedges of $Type_{max}$ vertices, and eliminate all other outedges from these vertices. This turns the 1-RMDP into a 1-RMC, M.

Since σ is optimal, some vertices $v \in Z_?$ in M have probability of termination equal to 1. Consider the dependency graph of M, call this G_M, and its decomposition into SCCs. There must exist a lowest SCC C of G_M which contains a node $v \in Z_?$ which has probability 1 of termination in M. Moreover, C must be a nontrivial SCC which contains more than one vertex, and which contains at least one "leaky" probabilistic node in $\mathcal{L}$, for otherwise the preprocessing step would have already moved v into Z_1 (or earlier into Z_0). Let B_C be the (nonnegative) moment matrix associated with the SCC C. From results in [Etessami and Yannakakis 2009] (see section 8, Lemma 8.2, in [Etessami and Yannakakis 2009]), it follows that B_C has spectral radius $\rho(B_C) \leq 1$. Since B_C is nonnegative, the standard Perron-Frobenius theory for nonnegative matrices (see, e.g., Theorem (8.3.1) of [Horn and Johnson 1985]) tells us that there is a row vector $u \geq 0$, $u \neq 0$, such that $uB_C = \rho(B_C)u \leq u$. By normalizing, we can find such

a vector that also satisfies $\sum_i u_i = 1$ (since the system $u \geq uB_C$ is homogeneous), and moreover such that all entries have at most m bits in numerator and denominator, so they are between $1/2^m$ and 2^m. Note that since C is strongly connected, any solution to the linear constraints $\{u \geq uB_C; \sum_i u_i = 1; u \geq 0\}$ automatically satisfies $u > 0$, i.e., $u_k > 0$ for all indices k.

We can now form a solution to our LP constraints (1.-10.): set to 0 all variables y_i and $y_{i,j}$ in our LP where i is not in C, as well as the variables $y_{i,j}$ where the node $i \in Type_{max}$ is in C, but such that edge $i \to j$ is not the edge that the optimal strategy σ picks for vertex i. For a non-max node $i \notin Type_{max}$ in C, set $y_i = u_i$, and for a max node $i \in Type_{max}$ in C where the selected edge by σ is $i \to j$, set $y_{i,j} = u_i$. It can be verified that constraints (1.–4.) are satisfied by y because $u \geq uB_C$. The vector y satisfies (5.) because $\sum_i u_i = 1$, and y obviously satisfies also (6.).

Since the nodes of C have probability 1 in M, and since C is nontrivial and must contain a leaky node z, for each node i in C, we take a path to z and route a flow of value $= u_i/2^{2m} \leq 1/2^m$ through that path. Note that since $u > 0$, $u_i > 0$. Moreover, all capacities are $\geq 1/2^m$, so no capacities are violated. All other flows $f_{i,j,k}$ are set to 0. Thus, all constraints (1.–10.) are satisfied.

($\Leftarrow$) Conversely, suppose that the LP has a solution y,f. We'll define a (partial) strategy for the maximizing player which yields value 1 for some vertices in $Z_?$.

Remove from the 1-RMDP and the dependency graph G'_A all edges $i \to j$ from nodes $i \in Type_{max}$ where $y_{i,j} = 0$, and remove all nodes $i \notin Type_{max}$ where $y_i = 0$, as well as nodes $i \in Type_{max}$ where $\sum_j y_{ij} = 0$. The remaining dependency graph G'' is "downward closed", i.e., if a node i is in G'' then so are all its successors. This is because of constraints (1.-4.): if a variable on the right hand side is > 0 then the same must be true for the variable on the left.

We define the (randomizing) strategy, which for every node $i \in Type_{max}$ with $\sum_j y_{i,j} > 0$, picks successor j with probability $p_{i,j} = y_{i,j}/\sum_j y_{i,j}$. We can now regard each such node as a probabilistic node, and letting, $y_i = \sum_j y_{i,j}$, we have transition probabilities $p_{i,j} = y_{i,j}/y_i$. With this interpretation, constraints (2.) of the LP become the same as (1.). We now have an 1-RMC M' with dependency graph G'' on the remaining nodes that are in the support of y.

Constraints (1.-4.), restricted to the positive y_i's, can in fact be written as $y \geq yB$, where B is the nonnegative moment matrix of M'. Since $y > 0$ (i.e., positive in all coordinates), this implies that $\rho(B) \leq 1$ (see, e.g., [Horn and Johnson 1985], Corollary (8.1.29)). Hence, it follows that $\rho(B_i) \leq 1$ for the moment matrices B_i for all the bottom SCCs C_i of G'' (because, $\rho(B_i) \leq \rho(B)$).

From the flow constraints (7.–10.), every bottom SCC C_i of G'' has a leaky node (because we can push positive flow to t from all of the nodes in the reduced graph). We claim that every vertex of a bottom SCC can reach the exit of its component (and terminate) in the 1-RMC M'. It will follow that the 1-RMC qualitative termination algorithm of ([Etessami and Yannakakis 2009], section 8, Theorem 8.1) will determine that the value is 1 for all the vertices of the bottom SCCs of G''.

Let S be the vertices of M' that cannot reach their exit (i.e., have 0 probability of termination in M') and let T be the vertices that can reach their exit. The leaky vertices in $\mathcal{L}$ are clearly in T because they have an edge to a node that we already assigned 1, so they can reach their exit. (a.) If i is a node in S that is not in $Type_{call}$, i.e., not a call node, then all its successors must also be in S. (b.) If i is a call node in S then at least one of its successors in the dependency graph (i.e. the return of the same box or the entry of the component corresponding to the box) must be in S; otherwise i could reach the exit of its component. Add up all the LP constraints (1.-4.) for nodes i in S and let $\sum_{i \in S} y_i \geq rhs$ be the resulting inequality (here rhs denotes the sum of the

right hand sides of the constraints 1.-4.). By (a.), for every node $i \in S$ with $i \notin Type_{call}$, the rhs includes $\sum_j p_{i,j} y_i = y_i$ because all these successors j are in S. By (b.), for every $i \in Type_{call} \cap S$, the rhs includes y_i (from constraint 3. or 4.). It follows that the rhs of these constraints cannot include anything else, in particular there is no edge $j \to i$ of G'' with j in T and i in S. Since every bottom SCC contains a leaky node, it follows that S contains no nodes from any bottom SCC. Thus, all vertices of the bottom SCCs can reach their exit. Furthermore, since the bottom SCCs all have spectral radius ≤ 1, by the 1-RMC algorithm from ([Etessami and Yannakakis 2009], section 8, Theorem 8.1) all these vertices will terminate with probability 1. Note, moreover, that the solution we obtained to the LP allows us to fix a (randomized) strategy for the maximizing player which will yield probability 1 for these vertices. □

So to summarize, we first compute the set Z_0 of vertices that have value 0, and perform the preprocessing step. Then we set up and solve the LP. If there is no solution, then for all remaining vertices $u \in Z_?$, $q^*_u < 1$, and thus $u \in Z_\$$. If there is a solution, use the above partial (randomized) strategy (given by the proof of the lemma) for some of the max nodes, leaving the strategy for other nodes unspecified. This allows us to set to 1 some vertices (vertices in the bottom SCCs of the resulting 1-RMC), and thus to move them to Z_1. We can then iterate the preprocessing step and then the LP step until we reach a fixed point, at which point we have categorized all vertices into one of Z_0, Z_1 or $Z_\$$. □

Example 6.4.
Consider again the 1-RMDP M that is obtained from the 1-RSSG of Figure 4 by removing vertex u_4. As mentioned earlier, the preprocessing step will set $Z_0 = \{u_5\}$, $Z_1 = \{t, (b_2, t)\}$, and $Z_? = Q - Z_0 - Z_1$. The LP will contain the following constraints in the first 6 groups:
1. $y_{(b_1,s)} \geq \frac{1}{2} y_s, \quad y_{(b_2,s)} \geq y_{(b_1,t)} + y_{u_2}, \quad y_{u_2} \geq \frac{1}{2} y_{u_3} + y_{u_1 u_2}, \quad y_{u_3} \geq y_{u_1 u_3}$
2. $y_{u_1 u_2} + y_{u_1 u_3} \geq \frac{1}{4} y_s$
3. $y_s \geq y_{(b_1,s)} + y_{(b_2,s)}$
4. $y_{(b_1,t)} \geq y_{(b_1,s)}$
5. $y_s + y_{(b_1,s)} + y_{(b_1,t)} + y_{(b_2,s)} + y_{u_1 u_2} + y_{u_1 u_3} + y_{u_2} + y_{u_3} = 1$
6. $y \geq 0$.

There is no solution to this set of constraints. This can be seen as follows. Combining (3) with (4) and (1), we have $y_s \geq y_{(b_1,s)} + y_{(b_2,s)} \geq \frac{1}{2} y_s + y_{(b_1,t)} + y_{u_2} \geq \frac{1}{2} y_s + y_{(b_1,s)} + y_{u_2} \geq y_s + y_{u_2}$. Hence $y_{u_2} = 0$. This implies then via the constraints that all the other y variables are also 0, contradicting constraint (5). Therefore, the LP has no solution, and hence the value for all the vertices in $Z_?$ is strictly between 0 and 1.

Consider however the 1-RMDP M' which differs from M in the outgoing edge of vertex u_2, and suppose this edge goes to vertex u_1 instead of the call port (b_2, s) of box b_2. The preprocessing step will be the same as for M. The set of constraints will be the same except that the constraint for (b_2, s) will be $y_{(b_2,s)} \geq y_{(b_1,t)}$, and the constraint for u_1 will be $y_{u_1 u_2} + y_{u_1 u_3} \geq \frac{1}{4} y_s + y_{u_2}$. It is easy to see that the set of constraints (1-6) for M' has a solution: $y_s = \frac{4}{15}$, $y_{(b_1,s)} = y_{(b_1,t)} = y_{(b_2,s)} = y_{u_3} = \frac{2}{15}$, $y_{u_2} = \frac{1}{15}$, $y_{u_1 u_2} = 0, y_{u_1 u_3} = \frac{2}{15}$. The flow network for the constraints (7-10) is essentially the same as the dependency graph (see Figure 4), except that nodes $u_5, (b_2, t)$ are not present, and the edge out of u_2 goes to u_1. The edges (u_1, u_2) and (u_1, u_3) out of the type-max vertex u_1 have respectively capacity $y_{u_1 u_2} = 0$ and $y_{u_1 u_3} = 2/15$, and every other edge (i, j) has capacity y_i. Thus, all edges have positive capacity, except for the edge (u_1, u_2), and every node has a path to the terminal t. Therefore there is a feasible flow from every source node $i \in Z_?$ to the terminal t, and thus there is a solution to the constraints (7-10) for these values of the y variables. That is, the LP is feasible. From the solution

of the LP, we set the strategy of the max vertex u_1 to select the successor u_3 for which $y_{u_1u_3} > 0$. All the vertices have positive y's and are assigned to Z_1. Thus, all the vertices of M' have value 1 except for u_5 that has value 0. □

We remark that, since our P-time algorithm uses linear programming, it is not strongly polynomial. Note that even for finite-state MDPs (with a reachability objective), the question of whether there exists a strongly polynomial time algorithm is a well-known open problem.

6.2. Minimizing 1-RMDPs

The qualitative problem for minimizing 1-RMDPs can be solved also in polynomial time. A separate algorithm is needed, because the maximization and minimization case are not symmetric.

THEOREM 6.5. *Given a minimizing 1-RMDP A, there is a polynomial time algorithm which determines for every vertex u of A which of the following three cases holds: (1) $q^*_u = 0$, or (2) $q^*_u = 1$, or (3) $0 < q^*_u < 1$.*

PROOF. As in the previous theorem, we want to classify the vertices into $Z_0, Z_\$, Z_1$, this time under optimal play of the minimizing player. We again consider the dependency graph G_A of A. We will again use $Z_?$ to denote those vertices that have not yet been classified. We compute first the set Z_0 of vertices with value 0, using the algorithm of Theorem 5.3. The remaining vertices are either in $Z_\$$ or Z_1. We use $Z_?$ again to denote the set of vertices that have not been assigned yet.

Initialize: $Z_1 \leftarrow Ex$; $Z_\$ \leftarrow \emptyset$; and $Z_? \leftarrow Q \setminus (Z_1 \cup Z_0)$;

Next, we again do a "preprocessing" step, which is "dual" to that of the preprocessing we did for maximizing 1-RMDPs, and categorizes some remaining "easy" nodes into Z_1 and $Z_\$$:

repeat
  if $u \in Z_? \cap (Type_{rand} \cup Type_{call})$ has all of its successors in Z_1
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_1 \leftarrow Z_1 \cup \{u\}$;
  if $u \in Z_? \cap Type_{min}$ has some successor in $Z_\$$
    then $Z_? \leftarrow Z_? \setminus \{u\}$; $Z_\$ \leftarrow Z_\$ \cup \{u\}$;
  if $u \in Z_? \cap (Type_{rand} \cup Type_{call})$ has some successor in $Z_0 \cup Z_\$$
    then $Z_? \leftarrow Z_? \setminus \{u\}; Z_\$ \leftarrow Z_\$ \cup \{u\}$;
  if $u \in Z_? \cap Type_{min}$ has all successors in (Z_1)
    then $Z_? \leftarrow Z_? \setminus \{u\}; Z_1 \leftarrow Z_1 \cup \{u\}$;
until (there is no change to $Z_?$)

Note that, after the preprocessing step, for every edge $u \to v$ in G_A from $u \in Z_?$ to $v \notin Z_?$, it must be the case that $v \in Z_1$ (otherwise, u would have already been moved to $Z_\$$ or Z_0). After preprocessing, we formulate a (different) LP, which has a solution iff there are more nodes currently in $Z_?$ which belong in $Z_\$$. Restrict attention to nodes in $Z_?$, and consider the subgraph G'_A of G_A induced by the nodes in $Z_?$. The LP has a variable y_i for every remaining vertex $i \in Z_?$ such that $i \notin Type_{min}$, and has a variable y_{ij} for every (remaining) node $i \in Type_{min}$, and successor j of i in G'_A. We shall need the following lemma:

LEMMA 6.6. *Consider a square nonnegative matrix B with at most n rows and having rational entries with at most l bits each. If $\rho(B) > 1$ then $\rho(B) \geq 1 + 1/2^m$ where $m = poly(n, l)$ and $poly(n, l)$ is some polynomial in n and l.*

PROOF. This can be proved in several ways. One way is to observe that since B is nonnegative, $\rho(B)$ is an eigenvalue of B, and that eigenvalues of B are roots of the

characteristic polynomial $h(x) = \det(xI - B)$. Given B, this univariate polynomial can be computed explicitly in polynomial time. It then follows from a result of Mahler [Mahler 1964], that distinct roots α and β of $h(x) * (x-1)$, must have distance $|\alpha - \beta| \geq 1/2^m$ where $m = poly(n, l)$ for some polynomial $poly(n, l)$. From this it follows that $\rho(B) \geq 1 + 1/2^m$.

Alternatively, this can be shown by using LP bounds: If $\rho(B) > 1$ then there is a nonzero vector $u \geq 0$ such that $Bu = ru$, with $r > 1$. Suppose I' is the set of indices i with $u_i > 0$ and let B' be the corresponding square submatrix $B_{I',I'}$ of B induced by the rows and columns in I'. The LP $B'x \geq x + 1$, $x \geq 0$ has a solution (scale $u[I']$ appropriately). Therefore, it has a rational solution with at most $m = poly(n, l)$ bits, hence its entries are at most 2^m. This solution, together with $u_i = 0$ in the rest of the indices satisfies $Bu \geq (1 + 1/2^m)u$, $u \geq 0$, $u \neq 0$. Therefore, $\rho(B) \geq (1 + 1/2^m)$ (by, eg., [Horn and Johnson 1985] Theorem (8.3.2)). □

Let $d = (1 + 1/2^m)$. Assume again w.l.o.g. that all component entries and return ports are in $Type_{rand}$. The constraints of our LP are as follows. For the LP we restrict attention to only those nodes j, i in $Z_?$.

(1) For every $j \in Type_{rand}$ that is not a component entry or a return, as well as for every $j \in Type_{call}$:

$$dy_j \leq \sum_{i \in Type_{rand} \wedge i \to j} p_{i,j} y_i + \sum_{i \in Type_{min} \wedge i \to j} y_{i,j}$$

(2) For every $j \in Type_{min}$:

$$d \sum_k y_{j,k} \leq \sum_{i \in Type_{rand} \wedge i \to j} p_{i,j} y_i + \sum_{i \in Type_{min} \wedge i \to j} y_{i,j}$$

(3) For every node i that is an entry of a component, say A_r:

$$dy_i \leq \sum_{j=(b,i) \in Type_{call} \wedge Y(b) = r} y_j$$

(4) For every node i that is a return node, say of box b: $dy_i \leq \sum_{j \in Call_b} y_j$.

(5) $\sum_i y_i + \sum_{i,j} y_{i,j} = 1$.

(6) $y \geq 0$.

LEMMA 6.7. *There exists a vertex $v \in Z_?$ such that $q^*_v < 1$ if and only if the LP in (1. – 6.) is feasible. Moreover, from a solution to the LP we can find a (partial) strategy for the minimizing player that forces termination from some such v with probability < 1.*

PROOF. ($\Rightarrow$) Assume there exists $v \in Z_?$ with $q^*_v < 1$. Fix an optimal strategy τ of the min player. By Theorem 4.2, there is a strategy that minimizes all the values at all the vertices and which picks one successor for each node $i \in Type_{min}$. This makes the 1-RMDP into a 1-RMC M'. Let G'' be the dependency graph of M'. Since τ is optimal, some of the vertices in $Z_?$ have probability < 1 of termination in M'. Let z be such a vertex in $Z_?$ which is in a lowest SCC of G'', meaning all vertices of $Z_?$ in lower SCCs of G'' (reachable from z) have probability $= 1$ of termination. Let C be the SCC of z and let R be the set of vertices that are reachable from z in G'' (including C). If one of the vertices in R was already placed earlier in $Z_\$$ (or Z_0), then z would also have been placed in $Z_\$$ in the preprocessing step (because z can reach it). We conclude that all the vertices of R are in $Z_?$ or Z_1. Furthermore, since the vertices of C have value < 1, they must all be in $Z_?$.

Consider the application of the qualitative termination algorithm for 1-RMCs from [Etessami and Yannakakis 2009] (section 8 of that paper) on the 1-RMC M'. All the

vertices of R in lower SCCs below C have value 1 in M'. When the algorithm examines C as the lowest unprocessed SCC, it will compute the spectral radius $\rho(B_C)$ of the moment matrix B_C corresponding to the SCC C and test if it is > 1. Since z has probability < 1 of termination in M', we must have $\rho(B_C) > 1$. Since $\rho(B_C) > 1$, we know $\rho(B_C) \geq d$, so (by, e.g., Theorem (8.3.2), of [Horn and Johnson 1985]) there is a nonzero vector $u \geq 0$ such that $du \leq uB_C$. Scale the entries of u so that $\sum u_i = 1$.

Form the following solution to the LP: For each i in C, if $i \notin Type_{min}$ then let $y_i = u_i$, and if $i \in Type_{min}$ then let $y_{i,j} = u_i$ for the single edge $i \to j$ selected by the optimal strategy τ and $y_{i,j} = 0$ otherwise. Note that for each min node i of C the optimal strategy selects a successor j that is also in C, otherwise the value of i would be 1. Set the variables corresponding to the other vertices and edges to 0.

The above vector y clearly satisfies constraints 5 and 6 since $u \geq 0$ and $\sum u_i = 1$. Constraints 1-4 for nodes that are not in C are satisfied because the left hand side is 0 (and the right hand side nonnegative), and the constraints 1-4 for nodes in C follow from $du \leq uB_C$, as in the proof of Theorem 6.2.

($\Leftarrow$) Conversely, suppose that the LP has a solution y. We'll define a (partial) strategy which forces termination with probability < 1 from some of the vertices remaining in $Z_?$.

For each $i \in Z_?$, let $u_i = y_i$ if $i \notin Type_{min}$ and let $u_i = \sum_j y_{i,j}$ if $i \in Type_{min}$. Define a (randomized) strategy where for $i \in Type_{min}$, if $u_i > 0$ then the probability of edge $i \to j$ is $p_{i,j} = y_{i,j}/u_i$; if $u_i = 0$ or i is not in $Z_?$ then pick an arbitrary transition out of i or arbitrary transition probabilities (it does not matter for the following argument). This makes the 1-RMDP into a 1-RMC M' and let G'' be the dependency graph of M'. Let z be a vertex of $Z_?$ with $u_z > 0$ that belongs to a highest SCC of G'', i.e., all vertices z' (of $Z_?$) in higher SCCs (from which z can be reached) have $u_{z'} = 0$. Let R be the set of vertices reachable from z and let C be the SCC of z. From preprocessing, we know that all vertices of R are in $Z_?$ or in Z_1, otherwise z would have been assigned to $Z_\$$ or Z_0. We will argue that z has probability < 1 of termination in M'.

Consider the 1-RMC qualitative termination algorithm from ([Etessami and Yannakakis 2009], Section 8) applied to M' with initial vertex z. All of the vertices in R can reach in M' a terminal exit (because there are no Z_0 vertices among the remaining nodes). Form the moment matrix B_R of R and consider its spectral radius $\rho(B_R)$. Let u be the vector defined above for vertices of R that are in $Z_?$ and 0 for the other vertices (note, all of the others are in Z_1). Clearly, $u \neq 0$ and $u \geq 0$. We claim that $du \leq uB_R$. For a vertex $i \in Z_?$, the inequality $du_i \leq (uB_R)_i$ follows from constraints (1.-4.) of the LP (since predecessor nodes and arcs that are not in R have y value 0). For a vertex i not in $Z_?$, $u_i = 0$ and the inequality clearly holds. Thus, $\rho(B_R) \geq d > 1$, and thus the 1-RMC qualitative termination algorithm will determine that vertex z has probability < 1 of termination. □

To summarize, we find Z_0, then do preprocessing to determine the "easy" Z_1 and $Z_\$$ vertices. Then, we set up and solve the LP, finding some more $Z_\$$ vertices, removing them, and iterating again with a preprocessing and LP step, until we exhaust $Z_?$ or there is no solution to the LP; in the latter case the remaining vertices all belong to Z_1. As for a strategy that achieves these assignments, in each iteration when we solve the LP we fix the strategy for certain of the min nodes in a way that ensures that some new vertices will be added to $Z_\$$ and leave the other min nodes undetermined. Moreover, in preprocessing, if $Type_{min}$ nodes get assigned to $Z_\$$ based on an outedge, we fix the strategy at that node accordingly. □

7. QUALITATIVE TERMINATION FOR 1-RSSGS IN NP∩CONP

The following is a simple corollary of Theorems 4.2, 6.2, and 6.5.

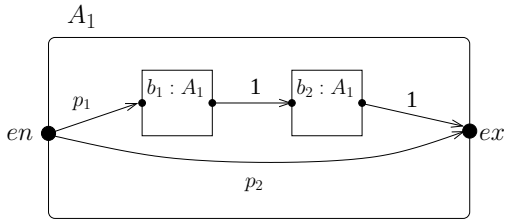

Fig. 5. 1-RMC A'

COROLLARY 7.1. *Given a 1-RSSG, A, and given a vertex u of A, we can decide in both NP and coNP whether $q^*_u = 1$. In other words, the qualitative termination problem for 1-RSSGs is in NP∩coNP.*[7]

PROOF. By Theorem 4.2, both players have optimal SM strategies. Thus, in order to decide in NP whether $q^*_u = 1$, we guess an optimal SM strategy σ for the maximizing player (player 1), yielding a minimizing 1-RMDP, and then verify in P-time using Theorem 6.5 that the optimal value $q^{*,\sigma}$ of u in the minimizing 1-RMDP is 1. Likewise, to decide in coNP whether $q^*_u = 1$, i.e., to decide in NP whether $q^*_u < 1$, we guess an optimal SM strategy τ for the minimizing player (player 2), and verify in P-time using Theorem 6.2 that in the resulting maximizing 1-RMDP $q^{*,\tau}_u < 1$. □

As the following theorem shows, it will not be easy to improve this upper bound. Recall that finite SSGs (with a reachability objective) are a special case of 1-RSSGs (with a termination objective). Define the quantitative termination decision problem for finite SSGs to be the following problem: given a finite SSG G with a target vertex labelled "1", and a starting vertex u of G, where the objective of the max player (resp., the min player) is to maximize (resp. minimize) the probability of reaching vertex 1, decide whether the value of the game $q^*_u \geq 1/2$. Condon [Condon 1992] showed that this problem is in NP ∩ coNP, and it has been a major open problem whether this upper bound can be improved to P-time.

THEOREM 7.2. *There is a P-time reduction from the quantitative termination problem for finite SSGs to the qualitative termination problem for 1-RSSGs.*

PROOF. Consider the 1-RMC depicted in Figure 5, where $p_1 + p_2 = 1$. As shown in ([Etessami and Yannakakis 2009], Theorem 3), in this 1-RMC the probability of termination starting at $\langle \epsilon, en \rangle$ is $= 1$ if and only if $p_2 \geq 1/2$.

Now, given a finite SSG, G with a target vertex labelled "1", and a starting vertex u of G, do the following: first "clean up" G by removing all nodes where the min player (player 2) has a strategy to achieve probability 0. By Theorem 5.3, we can do this in polynomial time. If u is among these nodes, then we are done, so assume it is not. The revised SSG will have two designated absorbing nodes, the old target node, labeled "1", and another node labeled "0". From every node v in the revised SSG which does not have full probability 1 on its outedges, we direct all the "residual" probability to vertex "0", i.e., we add an edge from v to "0" with probability $p_{v,}$"0" $= 1 - \sum_w p_{v,w}$, where the sum is over all remaining nodes w is the SSG. In the resulting finite SSG, we know that if the max player plays optimally, and the min player plays arbitrarily, there is

[7]For the more general class of 1-RCSGs one can not hope to easily establish such upper bounds for qualitative termination problems. In fact, we showed in [Etessami and Yannakakis 2008] that such a result for 1-RCSGs would constitute a breakthrough because it would imply that SQRT-SUM is in NP.

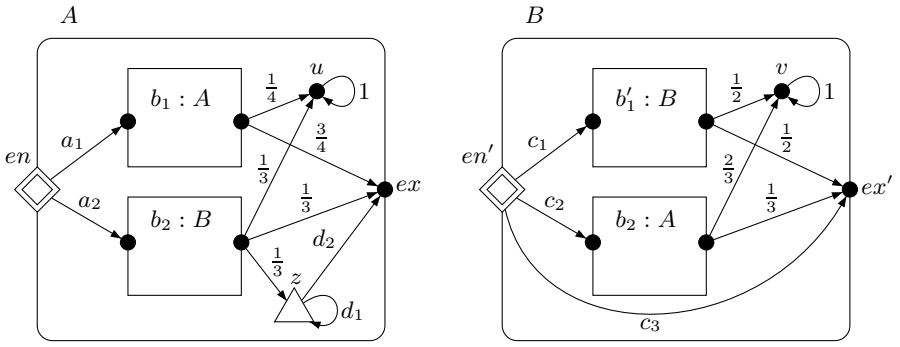

Fig. 6. *A linearly recursive 1-RSSG*

no bottom SCC in the resulting finite Markov chain other than the two designated terminating nodes "0" and "1". In other words, all the probability exits the system, as long as the maximizing player plays optimally.

Let G' be the revised finite SSG. Just add a copy of G' to the component A_1 of the 1-RMC in Figure 5, and replace the two edges incident to the entry en of A_1 by the following edges, all with probability 1: an edge from en to the vertex u of G', an edge from terminal node "1" of G to the exit ex of the component A_1, and an edge from the terminal node "0" of G' to the call port (b_1, en) of the left box b_1. Both boxes map to the unique component A_1. Call this 1-RSSG A.

We now claim that the value $q^*_u \geq 1/2$ in the finite SSG G' for terminating at the terminal "1" iff the value $q^*_{en} = 1$ for terminating in the resulting 1-RSSG, A. The reason is clear: after cleaning up the SSG, we know that under an optimal strategy for the maximizer for reaching "1", all the probability exits G' either at "1" or at "0". We also know that the maximizing player will win (i.e., $q^*_{en} = 1$) iff it can direct $\geq 1/2$ probability to go directly to the exit of the component in A, but this is precisely the probability that the maximizer can direct to terminal node "1" in G'. □

It is not at all clear whether there is a reverse reduction, i.e., a reduction from qualitative termination for 1-RSSGs to quantitative termination for finite SSGs. Note also that the qualitative problem for finite SSGs can be solved in polynomial time.

8. LINEARLY RECURSIVE 1-RMDPS AND 1-RSSGS

Recall that a RMDP or RSSG is *linearly recursive*, or *linear*, if there is no path in any component from a return port of a box to a call port of the same or another box. For instance, the 1-RMC (and thus also 1-RMDP) This corresponds to the notion of linear recursion in procedural programs. An example of a linearly-recursive 1-RSSG is give Figure 6. By contrast, the 1-RMC depicted in Figure 2 is *not* linearly recursive, because it contains a transition from the return port of box b_1 to the call port of box b_2. Clearly, standard finite-state MDPs and SSGs are a special case of linearly recursive 1-RMDPs and 1-RSSGs, respectively.[8] In this section we show that the positive features of finite MDPs and SSGs are inherited by their linearly recursive generalizations: all the vertex

[8]As shown in [Etessami and Yannakakis 2009] and the next section, there is a close connection between 1-RMCs, 1-RMDPs, and 1-RSSGs on the one hand and stochastic context-free grammars and branching processes and their controlled and game extensions on the other. In this correspondence, linear grammars (i.e. grammars where the right hand sides of the rules contain at most one non-terminal) map to linear 1-RMCs, 1-RMDPs, or 1-RSSGs; however, the direct mapping in the opposite direction yields in general a nonlinear grammar. Nevertheless, as we show in this section, the analysis of linear 1-RMDPs and 1-RSSGs is no harder than the case of finite state MDPs and games.

values are rational and have polynomial bit size. In the case of linear 1-RMDPs, the values can be computed exactly in polynomial time. In the case of linear 1-RSSGs, the quantitative problem is in NP $\cap$ coNP, just like the finite SSG case, and furthermore the problems for linear 1-RSSGs and SSG are polynomially equivalent. The qualitative problem can be solved in polynomial time for both linear 1-RMDPs and 1-RSSGs.

We present first the algorithm for the qualitative problem for linear 1-RSSGs. The algorithm is shown in Figure 7. The main call is Prune(A), and calls repeatedly two other procedures PruneMin and PruneMax that process the dependency graph to eliminate vertices until it is left with the vertices that have value 1. Intuitively, these procedures have the following function. PruneMin identifies a subset S of the current set W of vertices that does not include any exits, and such that if the game starts at a vertex of S and is constrained to stay within W then the min player can restrict it further to stay within the subset S; since S does not include any exits, the max player cannot win from any vertex of S and thus the subset S can be removed from the current set W. PruneMax removes from the current set W vertices from which the max player has no control of staying within W in the next step. Note that the call vertices behave like the min vertices: for a call vertex to have value 1, both of its successor vertices in the dependency graph must have value 1. The probabilistic vertices are treated like the max vertices in PruneMin and like the min vertices in PruneMax.

Note also that the algorithm depends only on the structure (vertices and edges) of the 1-RSSG, and not on the probabilities on the edges. This is similar to the finite-state case, and is in contrast to the general case of nonlinear 1-RMDPs and 1-RSSGs, where the values of the probabilities are important for the qualitative problem.

If we apply the algorithm on the example 1-RSSG of Figure 6, the first call to PruneMin will return the set of vertices $\{u, v, z\}$, which will be removed from the current set W. The subsequent call to PruneMax will remove all the other vertices except for the two exits ex, ex' and the entry en' of B. The final set W of vertices with value 1 is $\{ex, ex', en'\}$.

We show the correctness of the algorithm in the following theorem.

THEOREM 8.1. *Given a linear 1-RSSG, we can compute in polynomial time the set of vertices u with value $q^*_u = 1$.*

PROOF. Consider our algorithm, depicted in Figure 7. We claim that a call to Prune(A) returns precisely those vertices in $Z_1 = \{u \mid q^*_u = 1\}$. To establish this, we use several properties of the nested fixed point algorithm.

Property 1: A loop invariant of the algorithm is that at the beginning of each iteration of the repeat-until loop inside Prune(A), if a random, call, or min node is in the current W, then all its successors in G_A are also in W, and every max node of W has a successor in W. Clearly, this holds initially when $W = Q$. To see that it holds at the beginning of each subsequent iteration, note that the assignment $W \leftarrow$ PruneMax(W) at the end of the previous iteration eliminates precisely those vertices from W that violate this condition.

Property 2: During the repeat-until loop of Prune(A) the set W (which keeps track of remaining potential candidates for the set Z_1) monotonically shrinks (or stays unchanged and we terminate). Moreover, every node u that is removed from the set W is correctly removed, i.e., $u \notin Z_1$. This property is not hard to verify by inspection of the PruneMin and PruneMax procedures. In particular, for the nodes S returned by PruneMin(W) the min player has a strategy to terminate with probability 0, by picking, for each $Type_{min}$ node in S, an outedge to another node inside S. (Note that the call nodes in S must also have at least one successor inside S.)

Prune(A)
Input: Linear 1-RSSG A with set of vertices Q.
Output: The set of vertices $\{u | q^*_u = 1\}$.
  Construct the dependency graph G_A of A.
  $W \leftarrow Q$;
  repeat
    $W \leftarrow W \backslash$PruneMin($W$);
    $W \leftarrow$ PruneMax(W);
  until (there is no change in W);
  return W;

PruneMin(W)
  $S \leftarrow W \setminus Ex$;
  repeat
    if there is a node u in $S \cap (Type_{rand} \cup Type_{max})$ that has a successor (in the graph G_A) in $W \setminus S$, **then** $S \leftarrow S \setminus \{u\}$;
    if there is a node u in $S \cap (Type_{min} \cup Type_{call})$ that has no successor in S, **then** $S \leftarrow S \setminus \{u\}$;
  until (there is no change in S);
  return S;

PruneMax(W)
  $S \leftarrow W$;
  repeat
    if there is a node u in $S \cap (Type_{rand} \cup Type_{min} \cup Type_{call})$ that has a successor in $Q \setminus S$, **then** $S \leftarrow S \setminus \{u\}$;
    if there is a node u in $S \cap Type_{max}$ that has no successor in S, **then** $S \leftarrow S \setminus \{u\}$;
  until (there is no change in S);
  return S;

Fig. 7. P-time qualitative termination algorithm for linear 1-RSSGs

Suppose the algorithm reaches a fixed point, i.e. there is an iteration of Prune(A) with no change to W. Consider any SM strategy τ of the min player, and consider the subgraph G'' of G_A induced by the nodes W and by τ, where we take the selected edges out of the min nodes, and all the edges out of the random, call, and max nodes.
Property 3: The only bottom SCCs of G'' are the exit nodes. In proof, suppose there was a bottom SCC, C , of G'' other than the exit nodes. Then all nodes in C would stay in S in the last PruneMin call, and thus they would have been eliminated from W.

We say that a SCC is *trivial* if it contains only one node and no edges, and is *non-trivial* otherwise. All the bottom SCCs of G'' are trivial (they are precisely the exit nodes).
Property 4: Every nontrivial SCC of G'' has at least one edge leaving it from a $Type_{rand}$ or $Type_{max}$ node. To see this, suppose there was an SCC, C', of G'' such that the only edges coming out of C' were only edges coming out of some $Type_{call}$ nodes. Every call node v has two outgoing edges; if one of them leaves C' then the other must go to a node in C', otherwise the SCC C' would be trivial (it would only consist of v). Furthermore, every min node v of C' has exactly one outgoing edge in G''; if that edge leaves C', then again C' cannot contain any other nodes besides v and would be trivial. We conclude that every call and min node of C' has a successor in C', and every random and max node has all the successors in C'. This implies that in the last call of PruneMin(W), all nodes in C' would again have stayed in S, and thus they would have been eliminated from W.

Property 5: If $u = (b, en) \in W$ is a call port then the return port $v = (b, ex)$ of the same box cannot reach u in G_A, nor in G'', and hence it is not in the same SCC as u. This follows from the assumption that A is a linear 1-RSSG: the return port v can only reach ordinary nodes in the 1-RSSG A (no boxes), hence the same is true in the dependency graph G_A, and hence also in its subgraph G''. It follows that v is not in the same SCC as u. Note that $u = (b, en)$ has two edges in the dependency graph, $u \to v$ and $u \to en$; the property says that the first edge leaves the SCC of u.

We now establish that $W \subseteq Z_1$, i.e., for every strategy τ of the min player, from every node in W the max player has a strategy to terminate with probability 1. It suffices to consider an optimal SM strategy τ of the min player. Consider the DAG of SCCs of G'', bottom up. By Property 3, the bottom SCCs are the exit nodes, so they are in Z_1. Inductively, consider an internal SCC, C, for which we have already established that all of its successor SCCs are in Z_1. If C is a trivial SCC, i.e. $C = \{u\}$ for some node u and all edges of u go to lower SCCs, then the claim follows from the induction hypothesis, for all 4 possible types of u: rand, call, min, max. Suppose that C is a nontrivial SCC. By Property 5, the call nodes in C can have at most 1 edge to another node in C, with the other edge going to a lower SCC that was already marked to be in Z_1. Therefore, such call nodes can in effect be identified inside C with their successor in C, because their probability of termination will be the same as that of their successor in C (under any strategy of the max player). Thus, C may be viewed as a finite SSG with no $Type_{call}$ nodes. But by Property 4, C must also have an edge exiting it either from a $Type_{rand}$ node or a $Type_{max}$ node. In either case, the max player has a strategy inside C such that every node of C will terminate with probability 1. Thus $W \subseteq Z_1$. By Property 2, we already know $Z_1 \subseteq W$, so we are done. □

We address now the quantitative problem for linear 1-RMDPs and 1-RSSGs.

THEOREM 8.2.

(*1*) *In a linear 1-RMDP or 1-RSSG (with rational transition probabilities), the values of all the vertices are rational with polynomial bit complexity.*
(*2*) *The values can be computed exactly in polynomial time for both maximizing and minimizing linear 1-RMDP.*
(*3*) *The quantitative problem for linear 1-RSSGs is in NP ∩ coNP. Furthermore, it reduces to the quantitative problem for finite SSGs, i.e. if the latter can be solved in P, then we can compute the values for linear 1-RSSGs in polynomial time.*

PROOF. 1. Let A be a linear 1-RSSG, let σ, τ be optimal SM strategies for the maximizing and the minimizing player respectively, and let A' be the RMC obtained from A by keeping for each max and min vertex the transition selected by the respective optimal strategy, giving it probability 1, and removing the other transitions of these nodes. Clearly, A' is a linear 1-RMC, and every vertex u has the same value q^*_u in the 1-RSSG A as its termination probability in the 1-RMC A'. By Theorem 8.9 of [Etessami and Yannakakis 2009], all these probabilities are rational, of polynomial bit complexity.

2. The equation system $x = P(x)$ associated with a linear 1-RMDP is still a nonlinear min or max system because the equation corresponding to every call port is quadratic. But it can be decomposed into a sequence of linear min/max subsystems. Let A be a (maximizing or minimizing) linear 1-RMDP. Treating each component A_i of A as a directed graph on its set Q_i of vertices, let R_i be the set of vertices that can reach a call port and $S_i = Q_i - R_i$ the remaining vertices. Let $R = \cup R_i$ and $S = \cup S_i$. Since A is linear, all the return ports and the exit are in S_i and all the call ports are in R_i. By the definition, S_i is successor-closed, i.e. if a vertex is in S_i then so are all its successors.

We process the set S_i separately for each component: We view S_i as an ordinary finite (maximizing or minimizing) MDP where the objective is to optimize the probability of

reaching the exit node of A_i. Using the standard Linear Programming method for finite MDPs, we can compute in polynomial time the optimal probabilities for all vertices $u \in S_i$, which are the values q^*_u of these vertices in the 1-RMDP A. After doing this for all components, we have the value q^*_u of all vertices $u \in S$.

Now we process the set R. Restrict the system $x = P(x)$ of A to the equations corresponding to the vertices in R, substitute on the right hand side the values for all the variables in S, and let $y = P'(y)$ be the resulting system. The desired vector of values $q^*_u, u \in R$ is the least fixed point (LFP) of the system $y = P'(y)$. The system has only linear and min or max equations. In particular, for each call port $u = (b, en)$, the corresponding return port $v = (b, ex)$ of the same box belongs to S, thus the equation of u becomes $y_u = q^*_v \cdot y_{en}$, and if en is also in S, then the right hand side is a constant: $y_u = q^*_v \cdot q^*_{en}$. Note that the computed values q^*_v for the vertices $v \in S$ have polynomial size in the input 1-RMDP, thus all coefficients on the right hand sides are of polynomial size.

The LFP of the system $y = P'(y)$ can be computed by solving a Linear program, similar to computing the values for a MDP with a reachability objective. (In fact we could define a min or max MDP whose solution gives the LFP). Specifically, if A is a maximizing RMDP, then the objective of the LP is to minimize $\sum_{u \in R} y_u$ subject to $y \geq 0$ and a set of constraints corresponding to the equations of the system $y = P'(y)$. For each linear equation $y_u = \sum_v p_{uv} y_v + c_u$ of the system, the LP has a constraint $y_u \geq \sum_v p_{uv} y_v + c_u$, and for each max equation $y_u = \max(\{y_v | v \in R, (u, \perp, v) \in \delta\} \cup \{q^*_v | v \in S, (u, \perp, v) \in \delta\})$, the LP has one constraint $y_u \geq y_v$ for each successor $v \in R$ of the max node u, and a constraint $y_u \geq \max\{q^*_v | v \in S, (u, \perp, v) \in \delta\}$. As with MDPs, the LFP of the system is the (unique) optimal solution of the LP.

If A is a minimizing 1-RMDP, then we first compute the set Z_0 of vertices whose value is 0 and remove them and their equation from the system $y = P'(y)$. Then we set up and solve the LP to maximize $\sum_{u \in R} y_u$ subject to $y \geq 0$ and a set of constraints corresponding to the equations of the system $y = P'(y)$. For each linear equation $y_u = \sum_v p_{uv} y_v + c_u$ of the system, the LP has a constraint $y_u \leq \sum_v p_{uv} y_v + c_u$, and for each min equation $y_u = \min(\{y_v | v \in R, (u, \perp, v) \in \delta\} \cup \{q^*_v | v \in S, (u, \perp, v) \in \delta\})$, the LP has one constraint $y_u \leq y_v$ for each successor $v \in R$ of the min node u and a constraint $y_u \leq \min\{q^*_v | v \in S, (u, \perp, v) \in \delta\}$.

3. Membership in NP $\cap$ coNP follows from part 2, as in Corollary 7.1. We can test in NP if $q^*_u \geq p$ for a given vertex u and bound p by guessing a SM strategy σ for the max player, constructing the minimizing 1-RMDP A^σ obtained from A by fixing accordingly the strategy of player 1, and then computing the value of u in the 1-RMDP and checking that it is $\geq p$. Similarly, we can test in NP if $q^*_u < p$ (i.e., test in coNP that $q^*_u \geq p$) by guessing a SM strategy τ for the min player, and verifying that the value of u in the corresponding maximizing 1-RMDP is $< p$.

Suppose that the quantitative problem for finite SSGs can be solved in polynomial time. Note that this implies that we can compute exactly the values in polynomial time, because the values are rational numbers of polynomial bit complexity. Thus, using binary search, a polynomial number of comparison queries of the form $q^*_u \geq p$ suffices to compute the exact values. We will show how to compute the values for linear 1-RSSGs using a method similar to part 2. Given a linear 1-RSSG A, we partition again the set Q_i of vertices of each component into two sets, R_i (those that can reach a call port) and $S_i = Q_i - R_i$. Every set S_i induces a finite SSG with the exit node of component A_i as the target node (terminal 1), thus we can compute the values of all vertices $u \in S = \cup S_i$ in polynomial time.

We show now how to compute the values for the vertices in $R = \cup R_i$. Construct a new finite SSG G which includes all the vertices of R, the vertices of S that have incoming

edges in A from some vertices in R, and two new terminal nodes labelled 0 and 1. Every vertex $u \in S$ of G is now a probabilistic vertex (regardless of its original type in A) and has a transition to terminal 1 with probability q^*_u and a transition to terminal 0 with probability $1 - q^*_u$. (If some nodes $u \in S$ have value 0, then we can identify them with terminal 0.) The rand, min and max nodes of R retain the same type and the same transitions as in the given 1-RSSG A. Each call vertex $u = (b, en) \in R$, becomes now a random vertex: if $en \in R$ then u has a transition to en with probability $q^*_{(b,ex)}$ where (b, ex) is the return port of the same box, and a transition with probability $1 - q^*_{(b,ex)}$ to terminal 0; if $en \in S$, then u has a transition with probability $q^*_{en} \cdot q^*_{(b,ex)}$ to terminal 1, and a transition with the remaining probability $1 - q^*_{en} \cdot q^*_{(b,ex)}$ to terminal 0. By construction, for all vertices $u \in S$ of G, their value in the SSG G is q^*_u. If we form the linear min-max system of equations for the SSG G, restrict it to the equations for the vertices in R substituting on the right hand sides the values q^*_u for $u \in S$, value 1 for terminal 1 and 0 for terminal 0, the resulting system is precisely the system $y = P'(x)$ for the vertices in R obtained from the system $x = P(x)$ of the given 1-RSSG A. The LFP of this system is the vector of values for the vertices of R, both in the original 1-RSSG A, and in the constructed finite SSG G. □

The qualitative termination problem can be solved in polynomial time more generally for *piecewise linear* 1-RSSGs. We will call a 1-RSSG A *piecewise linear* if every vertex $v \in Type_{call}$ has at most one successor in the dependency graph G_A that is in the same SCC as v. The piecewise linear class includes obviously the linear 1-RSSGs. It also includes *hierarchical* 1-RSSG. A RSSG A is hierarchical if we can order the components as $A_1, \ldots, A_k$, so that the boxes of each component A_i are mapped to later components $A_j, j > i$ (there could be arbitrary paths between the boxes inside the components). The same algorithm of Theorem 8.1 for the qualitative termination problem applies more generally to piecewise linear 1-RSSGs, and the proof is exactly the same.

In the quantitative problem there is a subtle, but important complication in the piecewise linear case. The values of the vertices are still rational (as in the linearly recursive case), however they may have an exponential number of bits. Specifically the number of bits of the values can be exponential in the height of the DAG of SCCs of the dependency graph; this can happen even for the special case of hierarchical 1-RMCs [Etessami and Yannakakis 2009]. We can compute the values for piecewise linear (maximizing or minimizing) 1-RMDPs using a similar algorithm as in the linear case, by processing the SCCs of the dependency graph one by one bottom-up in topological order. If the height of the DAG of SCCs is bounded by a constant, then the algorithm runs in polynomial time. However, if the height is unbounded, then the numbers can get exponentially long in size, and the algorithm is no longer polynomially bounded.

9. BRANCHING MARKOV DECISION PROCESSES AND GAMES

In [Etessami and Yannakakis 2009] we established a close relationship between the extinction probability of multi-type branching processes and the termination probability of 1-exit RMCs; in particular, the computational problems are polynomial time equivalent. In this section we consider controlled and game generalizations of branching processes with the objective of maximizing or minimizing the extinction probability and show an analogous equivalence with 1-RMDPs and 1-RSSGs with termination probability objectives.

A (multi-type) *Branching Process* (BP) $G = (V, R)$ consists of a (finite) set $V = \{S_1, \ldots, S_n\}$ of *types*, and a (finite) set R of rules r of the form $S_i \xrightarrow{p_r} \alpha_r$, where $S_i \in V$, $p_r \in (0, 1]$, and α_r is a (finite) multi-set whose elements are in V, and such that for

every type S_i, $\sum_{\langle p_r | (S_i \xrightarrow{p_r} \alpha_r) \in R \rangle} p_r = 1$. The rule $S_i \xrightarrow{p_r} \alpha_r$ specifies the probability with which an entity of type S_i generates the multi-set α_r of offsprings in the next generation. As usual, the rule probabilities are assumed to be rational for computational purposes. We also assume that the multi-sets α_r in the right-hand sides of the rules are given in the input by explicit listing of all the elements. An equivalent representation of a multi-set α_r is as a n-vector $v(\alpha_r) = (v_1(\alpha_r), \ldots, v_n(\alpha_r))$ which specifies in unary the number $v_i(\alpha_r)$ of elements in α_r of each type $S_i \in V$. One could also use a more succinct vector representation where the v_i's are given in binary. It is shown in [Etessami and Yannakakis 2009] that the more succinct binary representation can be reduced in polynomial time to the unary (multi-set) representation, by introducing additional types and rules (this reduction carries over to the controlled and game extensions).

Starting from an initial population (set of entities of given types) μ at time (generation) 0, the BP G defines a stochastic process $X_0 = \mu, X_1, X_2, \ldots$ where the population X_k at time $k > 0$ is obtained from X_{k-1} by independently and simultaneously selecting for each entity in X_{k-1} a rule for the entity's type according to the rules' probabilities and replacing the entity with a new set of entities whose types are specified by the right hand side of the rule. The process continues as long as the current set of entities is not empty and terminates if and when it becomes empty. For a given initial population $X_0 = \mu$, the *extinction probability* of μ, denoted $p^*(\mu)$, is the probability that the process terminates, i.e., $X_k = \emptyset$ for some k. To compute the extinction probability for any initial population μ, it suffices to compute for each type $S_i \in V$ the extinction probability when the initial population has just a single entity of type S_i; we denote this probability by p_i^*, i.e. $p_i^* = p^*(\{S_i\})$. The extinction probability $p^*(\mu)$ of a population μ is equal to the product of the extinction probabilities of the types of the entities in μ. We will use the following notation in this section. If μ is a finite population with vector of type multiplicities $v(\mu) = (v_1(\mu), \ldots, v_n(\mu))$ and q is a real n-vector, we let $f(q, \mu)$ denote the expression $\Pi_{i=1}^n (q_i)^{v_i(\mu)}$. Thus, $p^*(\mu) = f(p^*, \mu) = \Pi_i (p_i^*)^{v_i(\mu)}$.

A *Branching Simple Stochastic Game* (BSSG) consists of the following:

— A finite set $V = \{S_1, \ldots, S_n\}$ of types
— A mapping $\mathtt{pl}$ that maps every type to a player in $\{0, 1, 2\}$ that controls the reproduction of the type, where 0 is the random player (chance or nature), player 1 is the *maximizer* and 2 the *minimizer*
— A finite set of actions for players 1 and 2 and a mapping act from every type in $\mathtt{pl}^{-1}(1) \cup \mathtt{pl}^{-1}(2)$ to a set of actions: these are the available actions to the player for entities of this type
— A set $R(S_i)$ of rules for each type S_i. For a type $S_i \in \mathtt{pl}^{-1}(0)$ these are probabilistic rules that have the same format as in a branching process, with left-hand side S_i and with probabilities that sum to 1. For a type $S_i \in \mathtt{pl}^{-1}(1) \cup \mathtt{pl}^{-1}(2)$, they consist of a set $R(S_i, a)$ of probabilistic rules for every action $a \in act(S_i)$, where the rules have left-hand side S_i and the sum of the probabilities of all the rules in the same set $R(S_i, a)$ is 1.

We will call the types mapped to 0, 1 and 2 respectively (and the entities of these types) random, max and min types (and entities). The subclass of BSSGs where no type is mapped to player 2 (respectively, no type is mapped to player 1) is called a maximizing (respectively, minimizing) *Branching Markov Decision Process* (BMDP). The subclass of BSSGs where all the types are mapped to player 0 are of course the standard Branching Processes.

A BSSG operates as follows. Starting from an initial population $X_0 = \mu$, a sequence of populations $X_1, X_2, \ldots$ is generated, where X_k is obtained from X_{k-1} as follows: First the players 1 and 2 select an available action for each entity in X_{k-1} of a max or min

type respectively; then a rule is chosen independently and simultaneously for all the entities of X_{k-1} probabilistically according to the probabilities of the rules for the type of the entity and the selected action in case of a max or min type; and finally, every entity of X_{k-1} is replaced by a new set of entities corresponding to the right-hand side of the selected rule for the entity. When a player 1 (or 2) selects the actions for the entities of max (or min) type, he can use randomization (i.e. pick the actions with some probabilities) and can base the selections (the probabilities for randomized selections) on the entire past history of the process up to time $k-1$. We may include in the history of the process up to time $k-1$ not only the populations $X_0, X_1, \ldots, X_{k-1}$, but also the information on all the past actions and rules applied and the parent-child relationships between all the entities up to this point; the history can be represented by a forest of depth $k-1$, with internal nodes labelled by rules and actions, and whose leaves at level $k-1$ form the population X_{k-1}. Thus, a strategy of a player is a function that maps every finite history (i.e., labelled forest of some finite depth as above) to a probability distribution on the set of tuples of actions for the entities in the current population (i.e. at the bottom level of the forest) that are controlled by the player. Let Ψ_1, Ψ_2 be the set of all strategies of players 1, 2. We say that a strategy is deterministic if for every history it chooses one tuple of actions with probability 1. We say that a strategy is *static* if it is both deterministic and, in addition, for each type S_i controlled by that player the strategy always chooses the same action a_i for all entities of type S_i in all histories. Our notion of an arbitrary strategy is quite general (it can depend on all the details of the entire history, and be randomized, etc.), however, as we will show, both players have optimal static strategies in BSSGs. Static strategies correspond to the (deterministic) SM strategies of RSSGs.

We will be interested in the extinction probability objective. The goal of player 1 is to maximize the extinction probability and of player 2 to minimize it (i.e., maximize the survival probability). For a given initial population μ, integer $k \geq 0$ and strategies $\sigma \in \Psi_1$, $\tau \in \Psi_2$, we denote by $p^{k,\sigma,\tau}(\mu)$ the probability that the process with initial population μ, and strategies σ, τ terminates in at most k steps (i.e., $X_k = \emptyset$), and we denote by $p^{*,\sigma,\tau}(\mu)$ the probability that it terminates in any number of steps. We let $p^k(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} p^{k,\sigma,\tau}(\mu)$, and $p^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} p^{*,\sigma,\tau}(\mu)$; the last quantity is the *value* of the game for the initial population μ. Determinacy holds for these games also as we'll see, i.e. $p^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} p^{*,\sigma,\tau}(\mu) = \inf_{\tau\in\Psi_2} \sup_{*,\sigma\in\Psi_1} p^{*,\sigma,\tau}(\mu)$, and similarly for $p^k(\mu)$. Furthermore, both players have optimal static strategies.

If μ has a single entity of type S_i, we will write p_i^* and p_i^k instead of $p^*(\mu)$ and $p^k(\mu)$. Given a BSSG, the goal is to compute the vector p^* of the p_i^*'s, i.e. the vector of extinction probabilities of the different types under optimal play. As we will see, from the p_i^*'s, we can compute the value $p^*(\mu)$ for any initial population μ, namely $p^*(\mu) = f(p^*, \mu) = \Pi_i (p_i^*)^{v_i(\mu)}$.

Let $x = (x_i | i = 1, \ldots, n)$ be a vector of variables for the different types of a given BSSG G. From the BSSG we can define a set of equations $x = P(x)$, which contains one equation $x_i = P_i(x)$ for each type S_i. There are three cases depending on whether S_i is a random, max, or min type. For each rule r, we let p_r denote the probability of the rule and α_r the right hand side.

(1) S_i is random type: $x_i = \sum_{r\in R(S_i)} p_r f(x, \alpha_r) = \sum_{r\in R(S_i)} p_r \Pi_{j=1}^n (x_j)^{v_j(\alpha_r)}$, where the summation is over all rules r with left hand side (lhs) S_i, p_r is the probability of the rule and α_r the right hand side (rhs). Recall that $v_j(\alpha_r)$ is the number of entities of type S_j in α_r.

(2) S_i is a max type: $x_i = \max_{a\in act(S_i)} \sum_{r\in R(S_i,a)} p_r f(x, \alpha_r) = \max_{a\in act(S_i)} \sum_{r\in R(S_i,a)} p_r \Pi_{j=1}^n (x_j)^{v_j(\alpha_r)}$.

(3) S_i is a min type: $x_i = \min_{a \in act(S_i)} \sum_{r \in R(S_i,a)} p_r f(x, \alpha_r) = \min_{a \in act(S_i)} \sum_{r \in R(S_i,a)} p_r \Pi_{j=1}^{n} (x_j)^{v_j(\alpha_r)}$.

As in the setting of 1-RSSGs, the operator P is again clearly monotone on $\mathbb{R}^n_{\geq 0}$, and on the unit n-cube $[0,1]^n$. Thus the operator P has a *Least Fixed Point* (LFP) $x^* \in [0,1]^n$. As shown in the following theorem, this LFP is precisely the vector p^* of optimal extinction probabilities.

THEOREM 9.1. *The optimal extinction probability vector p^* is the Least Fixed Point of the operator P. Furthermore, for any initial population μ, the optimal extinction probability $p^*(\mu) = f(p^*, \mu) = \Pi_i (p_i^*)^{v_i(\mu)}$ and $p^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} p^{*,\sigma,\tau}(\mu) = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} p^{*,\sigma,\tau}(\mu)$.*

PROOF. Let x^k be the k-fold application of P on the all-0 vector, i.e. $x^0 = 0$, and $x^k = P(x^{k-1})$ for $k > 0$. As in the case of the 1-RSSGs, the sequence x^k is (componentwise) monotonically non-decreasing as a function of k, bounded from above by the all-1 vector, and converges to the LFP, x^*, as $k \to \infty$. We will show the following claim.

CLAIM 1. *For any integer $k \geq 0$ and any finite initial population μ, the probability $p^k(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} p^{k,\sigma,\tau}(\mu)$ of extinction in at most k steps under optimal play is $p^k(\mu) = f(x^k, \mu) = \Pi_{i=1}^{n} (x_i^k)^{v_i(\mu)}$. Furthermore, there are strategies of the two players (and in fact deterministic ones) $\sigma_k \in \Psi_1$, $\tau_k \in \Psi_2$ that achieve this value, i.e, $p^k(\mu) = \inf_{\tau \in \Psi_2} p^{k,\sigma_k,\tau}(\mu) = \sup_{\sigma \in \Psi_2} p^{k,\sigma,\tau_k}(\mu)$.*

PROOF. We show the claim by induction on k. The basis, $k = 0$, is trivial. For the induction part, consider the generation of population X_1 from X_0 in step 1. We show first that $p^k(\mu) \geq \Pi_{i=1}^{n} (x_i^k)^{v_i(\mu)}$. Consider the following strategy σ_k for the max player. For each entity in the initial population $X_0 = \mu$ of a max type S_i, the max player selects in step 1 (deterministically) an action $a \in act(S_i)$ that maximizes the expression $\sum_{r \in R(S_i,a)} p_r f(x^{k-1}, \alpha_r)$ on the right side of the equation $x^k = P(x^{k-1})$. After the min player selects also actions for the entities of min type in X_0, and rules for all the entities are chosen probabilistically to generate the population X_1 for time 1, the max player follows an optimal $(k-1)$-step strategy σ_{k-1} for X_1. If we assume inductively that σ_{k-1} is deterministic, then σ_k is also deterministic. (It is not static however; the action chosen for an entity of a given type in a population X_i in the process may depend on the time i.)

Let τ be any strategy of the min player. Consider a combination of actions chosen with nonzero probability by the min player in step 1 for the entities of min type in $X_0 = \mu$. After this, a combination of rules is chosen independently for all the entities of μ and the population X_1 is generated accordingly with probability that is the product of the rule probabilities that were applied (because the rules are chosen independently). By the induction hypothesis, the probability that X_1 becomes extinct in the next $k-1$ steps (i.e. by time k) is at least $f(x^{k-1}, X_1)$. If we multiply $f(x^{k-1}, X_1)$ with the probability of the combination of rules used in step 1, and sum it over all the rule combinations, we can write the result as a product of $|\mu|$ terms, one for each entity in μ. The term for an entity with a random type S_i is $\sum_{r \in R(S_i)} p_r f(x^{k-1}, \alpha_r) = P_i(x^{k-1}) = x_i^k$; the term for an entity of max or min type S_i is $\sum_{r \in R(S_i,a)} p_r f(x^{k-1}, \alpha_r)$ where a is the action selected for this entity by the min or max player in step 1. For the max player, we selected an action $a \in act(S_i)$ that maximizes this expression, therefore the term for the entity is equal to $P_i(x^{k-1}) = x_i^k$. For an entity that belongs to the min player, no matter which action the player chose, the term is greater than or equal to the minimum value over all available actions, which is $P_i(x^{k-1}) = x_i^k$. Hence, for any combination of actions

chosen by the min player in step 1, the probability that the process terminates by step k under the strategies σ_k, τ is at least $f(x^k, \mu)$. Therefore, this holds also if τ makes a randomized selection in step 1, i.e., assigns nonzero probability to more than one combinations of actions for the min entities in μ. Thus, $\inf_{\tau \in \Psi_2} p^{k,\sigma_k,\tau}(\mu) \geq f(x^k, \mu)$ and hence $p^k(\mu) \geq f(x^k, \mu)$.

We can give a symmetric argument for the min player to prove the reverse inequality. Define strategy τ_k for the min player as follows. In step 1, the min player chooses for each entity of min type S_i in the initial population μ, an action $a \in act(S_i)$ that minimizes the expression $\sum_{r \in R(S_i,a)} p_r f(x^{k-1}, \alpha_r)$ on the right side of the equation $x^k = P(x^{k-1})$, and then, once the max player has chosen actions for the max entities of μ, and rules are selected and applied to generate the population X_1, the min player follows the strategy τ_{k-1} starting from X_1. By a symmetric argument to the max player case, it is easy to see that $\sup_{\sigma \in \Psi_2} p^{k,\sigma,\tau_k}(\mu) \leq f(x^k, \mu)$ and hence $p^k(\mu) \leq f(x^k, \mu)$. It follows that $p^k(\mu) = \inf_{\tau \in \Psi_2} p^{k,\sigma_k,\tau}(\mu) = \sup_{\sigma \in \Psi_2} p^{k,\sigma,\tau_k}(\mu) = f(x^k, \mu)$. □

In particular, for singleton populations μ, the claim implies that $p_i^k = x_i^k$ for all types S_i and all k.

Let $x^* = \lim_{k \to \infty} x^k$ be the LFP of the equation $x = P(x)$. We will show that for any initial population μ, the optimal extinction probability satisfies $p^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} p^{*,\sigma,\tau}(\mu) = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} p^{*,\sigma,\tau}(\mu) = f(x^*, \mu)$. In particular, for singleton populations μ, this property implies that $p_i^* = x_i^*$ for all types S_i.

Since x^k converges to x^* (from below) as $k \to \infty$, the sequence $f(x^k, \mu)$ converges to $f(x^*, \mu)$. Thus for every $\epsilon > 0$ there is a k such that $f(x^k, \mu) > f(x^*, \mu) - \epsilon$. The strategy σ_k of the max player achieves value $p^{*,\sigma_k,\tau}(\mu) \geq f(x^k, \mu) > f(x^*, \mu) - \epsilon$ for any strategy τ of the min player. Since this holds for every $\epsilon > 0$, it follows that $p^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} p^{*,\sigma,\tau}(\mu) \geq f(x^*, \mu)$.

For the converse inequality, let τ^* be the static strategy for the min player which always chooses for each entity of min type S_i an action a_i that minimizes the expression $\sum_{r \in R(S_i,a)} p_r f(x^*, \alpha_r)$. If we fix the actions for all the min types according to τ^*, the BSSG G becomes a maximizing BMDP G' where all the min types of G become now random types. Let $x = P'(x)$ be the set of equations for G'; for the random and max types S_i of G' the equation is the same, i.e., $P'_i = P_i$, while for min types S_i the function on the right-hand side changes from $P_i(x) = \min_{a \in act(S_i)} \sum_{r \in R(S_i,a)} p_r f(x, \alpha_r)$ to $P'_i(x) = \sum_{r \in R(S_i,a_i)} p_r f(x, \alpha_r)$. Thus, $P'(x) \geq P(x)$ for all $x \in \mathbb{R}^n_{\geq 0}$. Let $y^k, k = 0, 1, \ldots$ be the vector resulting from the k-fold application of the operator P' on the all-0 vector. Then $y^k \geq x^k$ for all k, and therefore the LFP y^* of P' is at least as great as the LFP x^* of P. However, x^* is a fixed point of P' since we have chosen actions for all the min types S_i that achieve the minimum in $P_i(x^*)$. Therefore, $x^* = y^*$ is also the least fixed point of P'.

Consider any strategy σ of the max player starting from initial population μ. Applying the Claim to the BMDP G' we know that for every k, the probability of extinction in k steps is at most $p^k(\mu) = f(y^k, \mu)$ Therefore, the probability of extinction in any number of steps is at most $f(y^*, \mu) = f(x^*, \mu)$. That is, $\sup_{\sigma \in \Psi_1} p^{*,\sigma,\tau^*}(\mu) \leq f(x^*, \mu)$. Combining with the previous inequality $p^*(\mu) \geq f(x^*, \mu)$, and since clearly $p^*(\mu) = \sup_{\sigma \in Psi_1} \inf_{\tau \in \Psi_2} p^{*,\sigma,\tau}(\mu) \leq \sup_{\sigma \in \Psi_1} p^{*,\sigma,\tau^*}(\mu)$, we conclude that $p^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} p^{*,\sigma,\tau}(\mu) = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} p^{*,\sigma,\tau}(\mu) = f(x^*, \mu)$. □

The proof of the Theorem shows that the min player has an optimal (deterministic) static strategy τ^* that achieves the optimal value $p^*(\mu)$ for any initial population μ. The same is true for the max player, but this would require a separate proof. We will deduce it instead from the relation we establish next with the 1-RSSGs.

THEOREM 9.2. *1. From any given BSSG G (resp. BMDP G), we can construct in polynomial time a 1-RSSG A (resp. 1-RMDP A) such that there is a bijection from the types of G to the components of A, and the optimal extinction probability of each type is equal to the optimal termination probability of the entry of the corresponding component. Furthermore, from optimal SM strategies of the players in A we can construct efficiently static optimal strategies for the players in G, and vice versa. This implies in particular that there exist optimal static strategies for both players in all BSSGs.*
2. From any given 1-RSSG A (resp. 1-RMDP A), we can construct in polynomial time a BSSG G (resp. BMDP G), such that there is a mapping from the vertices u of A to types S_u of G, such that the optimal termination probability q^*_u in A is equal to the corresponding optimal extinction probability p^*_u in G. Furthermore, from optimal SM strategies of the players in A we can construct efficiently static optimal strategies for the players in G, and vice versa.

PROOF. The constructions are similar to the constructions in [Etessami and Yannakakis 2009] for 1-RMCs and branching processes, extended here to handle the vertices and types controlled by the players.
1. Given a BSSG G with set of types V, we construct a 1-RSSG A that has one component A_i for every type S_i in V. If S_i is a random type, then the entry en_i of A_i is a probabilistic node and A_i has a path π_r from the entry en_i to the exit ex_i for every rule $r \in R(S_i)$. The path consists of a sequence of $|\alpha_r|$ boxes, one for each entity on the right-hand side of the rule r; the order of the boxes is not important. The entry en_i has an edge to the call port of the first box in the path π_r with probability p_r (i.e., equal to the probability of the rule r), the return port of each box has an edge with probability 1 to the call port of the next box, and the return port of the last box has a probability 1 edge to the exit ex_i of component A_i. Each box is mapped to the component for the type of the entity of α_r that corresponds to the box.

If S_i is a max type, then the entry en_i of A_i is a max node. The component A_i has a (probabilistic) node u_a for every action $a \in act(S_i)$ and an edge from en_i to each u_a. For every rule $r \in R(S_i, a)$, the component A_i has a path π_r from u_a to the exit ex_i which consists again of a sequence of boxes corresponding to the entities in the right-hand side α_r, similar to the case of random types. The component A_i for a min type is constructed similarly; the entry en_i is a min node, it has edges to a set of probabilistic nodes u_a corresponding to the actions $a \in act(S_i)$, and each u_a has a set of paths to the exit ex_i corresponding to the rules in $R(S_i, a)$. Note that if G is a maximizing or minimizing BMDP (i.e. has no min or no max types) then A is a maximizing or minimizing 1-RMDP.

Consider the system of equation $x = P(x)$ of A, let $x = (y, z)$ where y is the subvector of x for all the entry nodes and z for all the other vertices. Note that all the other vertices are either random or call or exit vertices. We can eliminate all the z variables from the equations and express them all in terms of the y variables, i.e. we can rewrite the system as $\{y = T(y), z = T'(y)\}$, where the components of T' are polynomials with positive coefficients (no max and min). It is easy to see from the construction that the equation for each variable y_i corresponding to the entry en_i of component A_i is exactly the same as the equation for the corresponding type S_i in the system of equations for the BSSG G. That is, the system for G is precisely $y = T(y)$. Thus, there is a 1-1 correspondence between the fixed points of the system $y = T(y)$ for G and the system $\{y = T(y), z = T'(y)\}$, or equivalently $x = P(x)$ for A, and furthermore, since T' is monotone, the correspondence preserves the least fixed points: if y^* is the LFP of T, which is also the optimal extinction probability vector p^* for the BSSG G, then $x^* = (y^*, T'(y^*))$ is the LFP of P and the optimal termination probability vector q^* for the 1-RSSG A.

There is a 1-1 correspondence between the max and min types of G and the max and min vertices of A, and from the construction, there is a 1-1 correspondence between static strategies in G and SM strategies in A. Suppose that σ^* is an optimal SM strategy for the max player in A. Consider the minimizing BMDP G' derived from G when we fix the strategy of the max player to σ^*, i.e., all the max types become random types, where for each max type we only keep the rules for the selected action and remove all the other rules. Let A' be the 1-RMDP obtained by applying our construction to the BMDP G'. It is easy to see that A' can be derived from A by keeping from each component A_i corresponding to a max type S_i only the path for the selected action a_i (contracting u_{a_i} into the entry en_i), and removing all the other paths corresponding to non-selected actions; that is, A' is essentially the 1-RMDP obtained from A by fixing the strategy of the max player to σ^*. Since σ^* is an optimal strategy in A, the value of all the vertices in A' is the same as in A, hence the value of all the types in G' is the same as in G, i.e, for every type S_i we have $p_i^* = \inf_{\tau \in \Psi_2} p_i^{*,\sigma^*,\tau}$. Therefore, σ^* is an optimal strategy for the max player in the BSSG G.

By a similar argument, if τ^* is an optimal SM strategy for the min player in A then τ^* is an optimal strategy in G (among all strategies, not only static). We have shown that both the max and the min player have optimal SM strategies in 1-RSSGs. It follows that they both also have optimal static strategies in BSSGs. Indeed, by the above arguments, there is a 1-1 correspondence between optimal SM strategies in A and optimal static strategies in G.

2. Given a 1-RSSG A we construct a BSSG G that has one type S_u for each vertex u of A. If u is an exit then S_u is a random type which has only one rule $S_u \xrightarrow{1} \emptyset$. If u is a probabilistic vertex, then u is a random type with one rule $S_u \xrightarrow{p_{uv}} \{S_v\}$ for every edge $u \xrightarrow{p_{uv}} v$ out of v. If u is a call port (b, en) and the return post of the box is $v = (b, ex)$, then S_u is a random type with one rule $S_u \xrightarrow{1} \{S_{en}, S_v\}$. If u is a max vertex, then S_u is a max type, the action set $act(S_u)$ has one action a_v for each edge $u \to v$ out of u, and the rule set $R(S_u, a_v)$ contains only one rule $S_u \xrightarrow{1} \{S_v\}$. Similarly if u is a min vertex. The BSSG G is constructed so that the equation system $x = P(x)$ for G is exactly the same as the equation system for A. It follows that they have the same least fixed point, i.e., the optimal termination probability vector q^* of A is equal to the optimal extinction probability vector p^* of G. Furthermore, there is a 1-1 correspondence between optimal SM strategies in A and optimal static strategies in G. □

Theorem 9.2 implies that all the upper bounds (algorithms) and lower bounds that we show for 1-RSSGs (resp. 1-RMDPs) apply equally to BSSGs (resp. BMDPs).

In [Etessami and Yannakakis 2009] an equivalence was shown also between 1-exit Recursive Markov chains and *Stochastic Context-Free Grammars* (SCFGs), where the termination probability of the 1-RMC is equal to the probability of the language generated by a corresponding SCFG. SCFGs are similar in format to branching processes, except that the symbols on the right-hand sides of the rules are ordered, i.e., the right-hand sides are strings rather than multi-sets (and have in addition 'terminal' symbols that do not reproduce any offsprings), thus the 'population' at any time is also a string. In a SCFG, we do not have to apply the rules simultaneously to all the members of a population, as in a branching process, but we can apply them one at a time following, e.g., a leftmost (or rightmost) order: a leftmost derivation applies in each stage a rule only to the leftmost (non-terminal) symbol of the remaining string. The infinite Markov chain defined by a SCFG with the leftmost derivation discipline is essentially the same as the Markov chain of a corresponding 1-RMC [Etessami and Yannakakis 2009] . A subclass of SCFGs (under a leftmost derivation discipline) corresponds to the

backbutton model of [Fagin et al. 2000]; this is a finite Markov chain model augmented with a 'backbutton' enabling the return to the previous state in the history, just like in web browsing (see [Etessami and Yannakakis 2009]).

We can define an extension of SCFGs and the backbutton model with a controller or two adversarial players, as we did with branching processes. For example, the controlled extension of the backbutton model may reflect different choices in the design of web pages (which links to include, how to place them etc.), which affect the transition probabilities from the pages and hence the evolution of the process. Similar to BMDPs and BSSGs, it can be shown along the same lines that the extensions of SCFGs are equivalent to 1-RMDPs and 1-RSSGs, and that the extensions of the backbutton model define subclasses of them.

10. UNDECIDABILITY: MULTI-EXIT RMDP TERMINATION AND 1-RMDP MODEL CHECKING

In this section we show that the quantitative and qualitative termination problems for multi-exit maximizing or minimizing RMDPs are undecidable. This is so even when the number of exits is bounded by a constant. Furthermore, the values (the optimal probabilities) of maximizing RMDPs cannot be approximated algorithmically within any nontrivial additive constant $< 1/2$. (Approximation within 1/2 is trivial: 1/2 has additive error at most 1/2.) We also show that for 1-exit RMDPs the qualitative model checking problem with respect to ω-regular or LTL properties (specified, say by a Büchi automaton or a LTL formula) is also undecidable. It obviously follows that the same questions for the more general RSSGs are also undecidable.

Before proving the undecidability results, let us first observe that the far easier "value 0" problems for maximizing and minimizing RMDPs, and for RSSGs, are decidable:

PROPOSITION 10.1. (Value 0 problem for RMDPs and RSSGs is decidable)

(*1*) *Given a multi-exit minimizing RMDP, or a RSSG, and given vertex u and exit ex, it is EXPTIME-complete to decide whether $q^*_{(u,ex)} = 0$.*

(*2*) *Given a multi-exit maximizing RMDP, a vertex u and exit ex, we can decide in polynomial time whether $q^*_{(u,ex)} = 0$.*

PROOF.

1. If we are given a multi-exit minimizing RMDP, and a vertex-exit pair (u, ex), the problem of deciding whether the controller has a strategy starting at vertex u to never terminate at exit ex (i.e., terminate there with probability 0), does not actually depend on the probabilities in the RMDP, but only depends on the structure of the underlying Recursive State Machine (RSM). Namely, we can think of the underlying RSM as defining a (non-stochastic) game between two players, A and B. Player A controls the nodes of the min player, and player B controls the probabilistic nodes, but can now choose the edge out of each vertex controlled by it. Starting at u player B wants to terminate at the exit ex and player A wants to avoid this. It is easy to see that $q^*_{(u,ex)} = 0$ if and only if player A has a winning strategy in this non-stochastic termination game. Such two-player termination games on (non-stochastic) Recursive Game Graphs are equivalent to reachability games for Pushdown game graphs, which are known to be EXPTIME-complete ([Walukiewicz 1996]).

If we are given a multi-exit RSSG, then we can similarly turn the question of whether $q^*_{(u,ex)} = 0$ into a two-player Recursive Game Graph (RGG). In this case, both the probabilistic nodes of the RSSG and the nodes controlled by the max player in the RSSG will be controlled by player B in the resulting RGG, whereas player A will con-

trol the nodes of the min player. Again, $q^*_{(u,ex)} = 0$ if and only if player A has a winning strategy in the resulting RGG, so again the problem is EXPTIME-complete.

2. In the case of maximizing RMDPs, the problem of deciding whether $q^*_{(u,ex)} = 0$ corresponds to a 1-player reachability problem, i.e., to whether there exists any path in the underlying Recursive State Machine, starting at vertex u (in the empty calling context) and terminating at exit ex. This is decidable in polynomial time (see [Alur et al. 2005]). □

Our proofs of undecidability for quantitative and qualitative termination problems for RMDPs and RSSGs are based on a connection that we establish between *Probabilistic Finite Automata* (PFAs) [Paz 1971] and (multi-exit) RMDPs. Our reductions show that PFAs can be seen, in effect, as a special case of maximizing RMDPs. We recall the basic definitions on PFAs. A PFA, $M = (V, \Sigma, T, v_1, v_n)$, has a (finite) set V of n states, a (finite) input alphabet Σ, a function T which maps every input letter $a \in \Sigma$ to a stochastic $n \times n$ matrix T_a, an initial state v_1, and an accepting state v_n. The automaton starts at the initial state v_1 and operates as follows for an input string $w \in \Sigma^*$: at each step, if the automaton is in state v_i and the next input letter is a, the automaton transitions to state v_j with probability equal to $T_a[i, j]$. The probability $P_M(w)$ that M accepts the string w is defined as the probability that the automaton is at the accepting state v_n after reading w. The language of a PFA is defined with respect to a given threshold λ: The language is $L(M, \lambda) = \{w \in \Sigma^* | P_M(w) > \lambda\}$. The *PFA emptiness problem* is to decide, given a PFA M and a threshold λ, whether $L(M, \lambda) = \emptyset$. This problem was shown to be undecidable originally in [Paz 1971]; subsequent proofs have established stronger undecidability properties [Condon and Lipton 1989; Madani et al. 2003; Blondel and Canterini 2003]. (The undecidability holds both whether we use strict or weak inequality in the definition of $L(M, \lambda)$.) We note that the qualitative problem for PFAs, namely determining whether there exists a string $w \in \Sigma^*$ such that $P_M(w) = 1$, is decidable (and PSPACE-complete); this is Theorem 12 of [Alur et al. 1995].

We first show undecidability for multi-exit maximizing RMDPs.

THEOREM 10.2. (Multi-exit maximizing RMDP termination problems are undecidable.)

For every fixed rational ϵ with $0 < \epsilon < 1/2$ the following problems are undecidable for maximizing RMDPs, even when the number of exits is bounded by a constant (the constant depends on ϵ).

(1) *Given a maximizing RMDP, A, with only one linearly-recursive component, entry en and exit ex such that either (i) $q^*_{(en,ex)} \geq 1 - \epsilon$ or (ii) $q^*_{(en,ex)} \leq \epsilon$, distinguish which of the two is the case.*

(2) *Given a maximizing RMDP, A, with one component, entry en and exit ex such that either (i) $q^*_{(en,ex)} = 1$ and there is a strategy of the max player that achieves value 1, or (ii) $q^*_{(en,ex)} \leq \epsilon$, distinguish which of the two is the case.*

These imply that both the quantitative and qualitative termination problems, as well as their witness counterparts, are undecidable and non-approximable for maximizing RMDPs and for RSSGs.

PROOF.

We will reduce from the emptiness problem for PFA. Given a PFA M, let $p^*_M = \sup\{P_M(w) | w \in \Sigma^*\}$. For part 1, we will construct a (multi-exit) maximizing RMDP A with an entry en and an exit ex such that $p^*_M = q^*_{(en,ex)}$. Thus, for a threshold λ, the language $L(M, \lambda) = \emptyset$ iff $q^*_{(en,ex)} \leq \lambda$; this establishes the undecidability of the quanti-

tative problem for RMDPs. We use an inapproximability result for PFAs to derive an inapproximability for RMDPs. The number of exits of the RMDPs in this construction that shows inapproximability is not bounded. We present a more involved reduction from 2-counter machines that shows the inapproximability for RMDPs with a bounded number of exits. For the qualitative problem ($q^*_{(en,ex)} = 1?$) we will embed the RMDP A of part 1 into another RMDP A'.

Part 1. We first present a simple reduction from the PFA emptiness problem. Let $M = (V, \Sigma, T, v_1, v_n)$ be a PFA with n states. Define a RMDP A that has one component (call it also A) with a single entry en, and n exits $ex_1, \ldots, ex_n$, one for each state of M. The entry en is a max node and has edges to the call ports of a set of $|\Sigma|$ boxes $b_a, a \in \Sigma$; all the boxes are of course mapped to the single component A. In addition en has an edge to the exit ex_1. The return ports of the boxes b_a are probabilistic vertices. Each return port (b_a, ex_i) has an edge to each exit ex_j with probability $T_a[i, j]$. This concludes the definition of the RMDP A. Note that A is a linear RMDP. Starting from the entry en of A, the max player wants to maximize the probability $q^*_{(en,ex_n)}$ of terminating at exit ex_n. We claim that $q^*_{(en,ex_n)}$ is precisely $p^*_M = \sup\{P_M(w)|w \in \Sigma^*\}$.

For any word $w \in \Sigma^*$, consider the strategy σ_w of the max player which at the i-th step chooses the edge to the box corresponding to the ith letter of the reverse word w^R for $i = 1, \ldots, |w|$; at step $|w| + 1$, the max player chooses the edge to the exit ex_1. From that point on, all the actions are probabilistic. It is easy to see then that by the construction, the sequence of probabilistic actions corresponds to a run of the PFA M on the input string w, and the run ends at the accepting state q_n iff the RMDP terminates at the exit ex_n of the top component. Thus, the probability $q^{*,\sigma_w}_{(en,ex_n)}$ of termination at ex_n under strategy σ_w is $P_M(w)$. It follows that $p^*_M \leq q^*_{(en,ex_n)}$.

Conversely, consider a (in general, randomized) strategy σ of the max player. At each step, when the process is at an entry of a component, the max player has to either choose a letter of Σ and transition to the call port of the corresponding box or decide to move to ex_1 and exit the current box (or it can randomize between these choices). In case it moves to ex_1, the control passes to the probabilistic player and stays with him for the remainder of the game until the process reaches an exit of the top component. If $q^{*,\sigma}_{(en,ex_n)} > 0$, then the max player has to choose the edge to the exit at some point with positive probability (otherwise the process will never terminate). For each finite word w, let $p_\sigma(w)$ be the probability that the max player, using strategy σ, chooses the reverse w^R of w in the first $|w|$ steps and exits in step $|w| + 1$. Then $q^{*,\sigma}_{(en,ex_n)} = \sum_{w\in\Sigma^*} p_\sigma(w)P_M(w)$. Hence $q^{*,\sigma}_{(en,ex_n)} \leq \sum_{w\in\Sigma^*} p_\sigma(w)p^*_M \leq p^*_M$ for every strategy σ of the max player. Therefore, $q^*_{(en,ex_n)} \leq p^*_M$ and the two quantities are equal.

The constructed RMDP above has one component with one entry and many exits. The number of exits of the RMDP is equal to the number of states of the PFA. In [Blondel and Canterini 2003] it is shown that the PFA emptiness problem is undecidable even for PFAs with only 2 letters and 46 states, specifically it is undecidable to determine for a given such PFA and threshold λ whether there is a word w such that $P_M(w) > \lambda$, (and the same holds for the other inequalities $\geq \lambda, < \lambda$ and $\leq \lambda$). It follows that the quantitative problem is undecidable for maximizing linear RMDPs with one component and 46 exits.

Another undecidability proof for PFAs that yields strong nonapproximability results is given in [Condon and Lipton 1989; Madani et al. 2003], where it is shown (see Theorem 3.3 of [Madani et al. 2003]) that for every $0 < \epsilon < 1/2$ it is undecidable, given a PFA M, to distinguish between the case that M accepts some word with probability $> 1 - \epsilon$ and the case that M accepts every word with probability $< \epsilon$, even when

it is guaranteed (promised) that one of the two cases holds. This fact together with our reduction implies part 1 of the theorem, except that the number of exits in this construction is generally not bounded by a constant, because the number of states of the PFA in the construction of [Condon and Lipton 1989; Madani et al. 2003] is not bounded: the reduction there is from the halting problem for a 2-counter machine on empty input, and the number of states of the PFA depends on the number of states of the 2-counter machine (which is in general unbounded) and on ϵ. However, we can combine and modify the reductions, to get a reduction from the 2-counter problem to the RMDP problem where the number of exits in the RMDP depends only on ϵ. We explain below how to do this. First, we review the definition of 2-counter machines and summarize the main ideas in the reduction of [Madani et al. 2003].

A *2-counter machine* (2CM) with empty input has a finite set $S = \{s_1, \ldots, s_m\}$ of states, it has two counters that can hold nonnegative values, and a set of (deterministic) transition rules. One state, s_1, is distinguished as the initial state and another state, s_h, as the halting state. Initially the machine starts at state s_1 with both counters 0. The transitions from each state depend only on whether the counters have zero or nonzero value, i.e. there are four rules for every state $s_i \neq s_h$; at the halting state s_h the machine halts (there are no transitions). Each transition rule specifies the new state and the (possible) change to each counter which can be one of 3 possibilities: increment by 1, decrement by 1 (this is allowed only if the counter is positive) and leave the same value. A *configuration* of the 2CM can be represented by a string $s_i a^j b^l$, where s_i is the state and j, l are the values of the counters. Starting from the initial configuration $c_1 = s_1 a^0 b^0 = s_1$, the computation of the 2CM is a sequence of configurations which either goes on forever or ends with a halting configuration, i.e. one where the state is s_h. We can assume without loss of generality that the machine zeroes its counters before it halts, i.e. that the halting configuration is unique: $s_h a^0 b^0 = s_h$. The computation can be represented by a string $\#c_1\#c_2\#\ldots$, where $\#$ is a separator symbol and $c_1, c_2, \ldots$ are the successive configurations. It is undecidable to determine whether a given 2CM halts on empty input (see [Hopcroft and Ullman 1979] for more details).

From a given 2-counter machine C and rational ϵ, [Madani et al. 2003] constructs a PFA M which has the properties that, if C does not halt on the empty input, then M accepts no string with probability $\geq \epsilon$, and if C halts on empty input and the string w represents the halting computation of C, then there is an integer d such that $P_M(w^d) > 1-\epsilon$. The PFA M checks probabilistically (with some probability of error) that its input string consists of repetitions of the valid halting computation of C on empty input. For each complete computation it checks that (1) it has the right format $\#c_1\#c_2\#\ldots$, it starts at the initial configuration $c_1 = s_1$, ends in the halting configuration s_h, and (2) each configuration c_j follows from the previous c_{j-1} according to the transition rules of the 2CM (or reinitializes the 2CM after reaching the halting configuration), i.e., it checks that (2a) the new state in c_j is updated correctly from the state and the counter values in c_{j-1}, and (2b) the values of the counters in c_j are updated correctly. All of these can be checked by a deterministic finite automaton (DFA), except for property (2b), checking the correctness of the counter updates. If the 2CM C has m states, then properties (1) and (2a) can be checked easily by a DFA M_1 with $O(m)$ states which rejects the input if it finds a violation. As shown in [Madani et al. 2003], property (2b) can be checked probabilistically (with some small probability of error even if the input is valid) by a PFA M_2 with a number of states that depends on ϵ. The overall PFA M constructed in [Madani et al. 2003] from the given 2CM is the product of the DFA M_1 and the PFA M_2, and this is why the number of states depends on both m and ϵ.

To show that part 1 of the theorem holds for RMDPs with a bounded number of exits, we will reduce directly from the 2-CM halting problem. Let us say that a string is *DFA-legal* if it passes the test of the DFA M_1, i.e., it satisfies properties 1 and 2a. We will

construct the RMDP in such a way that the max player can only generate in reverse strings that are DFA-legal. Hence the only thing that remains to be checked probabilistically is property 2b, for which it suffices to use a PFA M_2 with a bounded number of states. In more detail, first it is convenient to tag the symbols in the computation of the 2CM by the transition taking place in each step: map every configuration, consisting of state s_i and values j, l for the counters to the triple (s_i, d_1, d_2) where d_1, d_2 are 0 or 1 depending on whether the counter values j, l respectively are 0 or positive; we call this the *tag* of the configuration. Note that the tag of a configuration determines the transition. Let our new "tagged" alphabet be the set Σ consisting of all tuples (f, s_i, d_1, d_2) where $f \in \{\#, a, b\} \cup S$, $s_i \in S$, $d_1, d_2 \in \{0, 1\}$ such that: if $f \in S$ then $f = s_i$, if $f = a$ then $d_1 = 1$, if $f = b$ then $d_2 = 1$, and if $s_i = s_h$ then $d_1 = d_2 = 0$. Represent configurations and computations by strings over the tagged alphabet Σ, where we add the tag of the configuration to each symbol s_i, a, b, and for each separator $\#$ we attach the tag of the following configuration. Thus, for example the tagged string of the computation starts as $(\#, s_1, 0, 0)(s_1, s_1, 0, 0)\ldots$, and if the computation halts at the end, then the last symbol is $(s_h, s_h, 0, 0)$.

Define Φ to be the set of all pairs of symbols $t, t' \in \Sigma$ that can appear consecutively (t right before t') in a DFA-legal string. Specifically, if $t = (\#, s_i, d_1, d_2)$ then t' must be (s_i, s_i, d_1, d_2); if $t = (s_i, s_i, d_1, d_2)$ and $d_1 = 1$ then $t' = (a, s_i, d_1, d_2)$, if $d_1 = 0$ and $d_2 = 1$ then $t' = (b, s_i, d_1, d_2)$, and if $d_1 = d_2 = 0$ then t' must have the form $t' = (\#, s_j, d_1', d_2')$ where s_j, d_1', d_2' are consistent with the transition of the 2CM from state s_i with zero counters. For $t = (a, s_i, d_1, d_2)$ the next symbol t' can be either the same, (a, s_i, d_1, d_2), or it can be (b, s_i, d_1, d_2) (provided $d_2 = 1$) or $(\#, s_j, d_1', d_2')$ (provided $d_2 = 0$) where s_j, d_1', d_2' is consistent with the move of the 2CM from state s_i with counter 1 nonzero and counter 2 zero. For $t = (b, s_i, d_1, d_2)$ there is a similar set of possibilities for the next symbol t'. For $t = (s_h, s_h, 0, 0)$ corresponding to the symbol for the halting configuration (the last symbol of the halting computation) the only possible next symbol t' is $(\#, s_1, 0, 0)$ starting a new computation.

From the 2-counter machine C and ϵ, we construct the RMDP A which has one component, also denoted A, with many entries and exits. There is one entry t for each symbol t in Σ, and an additional special entry en. The exits correspond to the states of the PFA M_2 that checks property 2b. Let $M_2 = (V, \Sigma, T, v_1, v_k)$ be this PFA with k states (k is a constant that depends on ϵ), where v_1 is the initial state, v_k the accepting state, and T_t the transition matrix corresponding to each symbol $t \in \Sigma$. The RMDP has k exits $ex_1, \ldots, ex_k$. For each $t \in \Sigma$, there is a box b_t (mapped to the only component of the RMDP). All the entries of the RMDP are max nodes. For each pair $(t, t') \in \Phi$ there is a transition from the entry t' of A to the call port (b_t, t) of box b_t. In addition, the entry $(\#, s_1, 0, 0)$ corresponding to the first symbol of a computation has a transition to the exit ex_1. The special entry en has only one transition, namely to the call port (b_{t_h}, t_h) where $t_h = (s_h, s_h, 0, 0)$ is the symbol corresponding to the halting configuration, i.e. the last symbol of a halting computation. Note that all transitions to a box b_t go the entry t of the box (the rest of the call ports are not used). The return ports of the boxes are probabilistic nodes. Each return port (b_t, ex_i) has a transition with probability $T_t[i, j]$ to the exit ex_j (if this probability is positive). This concludes the definition of the RMDP A.

We claim that the RMDP has the property that (i) if C halts on empty input then $q^*_{(en,ex_k)} > 1 - \epsilon$, and (ii) if C does not halt then $q^*_{(en,ex_k)} \leq \epsilon$. Suppose that C halts, let w be the halting computation (over the tagged alphabet Σ), let d be large enough so that w^d is accepted by the PFA M_2 with probability $> 1 - \epsilon$, and let σ_{w^d} be the strategy of the max player which moves to a sequence of boxes indexed by the symbols of the reverse of string w^d. That is, if $w^d = t_1 \ldots t_N$, then the max player moves from the

special entry en (the initial vertex) to entry t_N of box t_N (note that t_N is the symbol $(s_h, s_h, 0, 0)$ of the halting configuration), within that box it moves to entry t_{N-1} of box t_{N-1} and so forth. At the end when it is at the entry $t_1 = (\#, s_1, 0, 0)$ of a box b_{t_1} it follows the direct transition to the exit ex_1 of the box. From then on, all transitions are probabilistic and follow a computation of the PFA M_2 on input w^d. Since the PFA accepts with probability $> 1-\epsilon$, it follows that under this strategy σ_{w^d}, the RMDP will terminate at the exit ex_k with probability $> 1-\epsilon$. Note furthermore that the RMDP terminates in any case to some exit with probability 1; we will need this fact in part 2.

On the other hand, consider any (possibly randomized) strategy σ of the max player. For each finite string w, let $p_\sigma(w)$ be the probability that the player makes a sequence of transitions to a sequence of boxes corresponding to the reverse string of w, and then in step $|w|+1$ the max player exits directly the current box; after this, the remaining transitions will be probabilistic and will follow a run of the PFA on input string w. By the construction of the RMDP, only strings w that are DFA-legal can be generated by the max player, i.e. can have $p_\sigma(w) > 0$. The probability that the process terminates at the exit ex_k under this strategy is $q^{*,\sigma}_{(en,ex_k)} = \sum_w p_\sigma(w) P_{M_2}(w)$. If the 2CM does not halt, then $P_{M_2}(w) < \epsilon$ for all DFA-legal strings w, hence $q^{*,\sigma}_{(en,ex_k)} < \epsilon$, and thus $q^*_{(en,ex_k)} \leq \epsilon$.

Part 2. For the qualitative RMDP problem, we reduce from the quantitative problem of part 1. To prove part 2 for a given value of ϵ where $0 < \epsilon < 1/2$, we let $\delta = \epsilon/(1+\epsilon)$ (note: $0 < \delta < 1/2$), and reduce from the quantitative problem for the RMDP of part 1 with δ in place of ϵ. That is, let A be a (maximizing) RMDP constructed as in part 1 that has one component with an entry u, exit v, for which we want to distinguish between the case (i) that there is a strategy σ of the max player under which the RMDP starting from entry u terminates at exit v with probability $\geq 1-\delta$ and terminates at the other exits with the remaining probability, and the case (ii) that for any strategy, the RMDP starting from u terminates at v with probability $\leq \delta$. We construct an RMDP A' with two components A_1, A_2. Component A_2 is the same as the RMDP A of part 1. Component A_1 has a single entry en and exit ex and has a structure similar to the one shown in Figure 5, except that the entry does not have a direct transition to the box b_1, and we have an additional box b_0, mapped to A_2. The entry en of A_1 is a probabilistic node and has an edge with probability 1 to the call port (b_0, u) of the box b_0. The return port (b_0, v) of b_0 has a probability 1 edge to the exit ex of A_1, while all the other return ports of b_0 have probability 1 edges to the entry of a box b_1 mapped to A_1. The return port (b_1, ex) of b_1 has a probability 1 edge to another box b_2 mapped also to A_1 and the return port (b_2, ex) of b_2 has a probability 1 edge to the exit ex of A_1. This concludes the definition of A'.

Note that all the max vertices of A' are in A_2. Suppose that there is a strategy σ of the max player in A (i.e. in A_2) such that A starting from the entry u terminates at exit v with probability $p_2 \geq 1-\delta$ and terminates at the other exits with the remaining probability. Consider the strategy σ' for A' in which the max player always uses σ whenever the box b_0 is entered. Under this strategy, the RMDP A' will reach with probability $p_2 \geq 1-\delta > 1/2$ the return port (b_0, v) of box b_0 and proceed from there directly to the exit ex, and with the remaining probability $p_1 = 1-p_2 \leq \delta < 1/2$, the RMDP will reach a different return port of box b_0 and then proceed to box b_1. Thus, under the strategy σ' of the max player, the RMDP A' functions like the 1-RMC of Figure 5, where $p_2 > 1/2$. Since $p_2 > 1/2$, it follows that under the strategy σ' the RMDP A' will reach the exit ex and terminate with probability 1.

On the other hand, let p_2 be the supremum, over all strategies of the max player in A, of the probability that A starting from entry u terminates at exit v, and suppose

that $p_2 \leq \delta < 1/2$. Then, for any strategy σ' of the max player for A', every time that the box b_0 is entered, the process reaches the return port (b_0, v) and goes directly to the exit ex at most with probability $p_2 < 1/2$ and with the remaining probability it either reaches one of the other return ports and proceeds to the box b_1 or stays in b_0 for ever and never terminates. Consequently, for any strategy σ' of the max player in A', the probability that it terminates (at the only exit ex) is no greater than that of the 1-RMC of Figure 5 with $p_2 \leq \delta < 1/2$. The probability $q^*_{(en,ex)}$ of the 1-RMC satisfies the equation $x = p_1 x^2 + p_2$ and is the least nonnegative solution. It is thus p_2/p_1 (since $p_2 < p_1$), hence $q^*_{(en,ex)} \leq \delta/(1-\delta) = \epsilon$.

The RMDP A' constructed above has 2 components. If we wish, we can combine them into one component with the entries and exits of both components (we can always do this with any RMDP or RSSG). The number of exits is bounded by a constant that depends on ϵ.

□

We next show undecidability for multi-exit minimizing RMDPs. In this case our undecidability results are slightly weaker in that, unlike the maximization case, they do not imply that any non-trivial approximation of the minimal termination probability is undecidable.

THEOREM 10.3. (Multi-exit minimizing RMDP termination problems are undecidable.)

The following problems are undecidable for minimizing RMDPs, even when the number of exits is bounded by a fixed constant (≤ 47).

(*1*) *For any fixed probability $r \in (0,1)$, given a linearly-recursive minimizing RMDP, A, with entry en and exit ex, it is undecidable whether $q^*_{(en,ex)} < r$.*

(*2*) *Given a minimizing RMDP, A, with one component, entry en and exit ex, it is undecidable whether $q^*_{(en,ex)} = 1$.*

In other words, both the quantitative and qualitative termination problems (as well as their "witness" counterparts, which in this minimization case are trivially equivalent) are undecidable for minimizing RMDPs (and for RSSGs).

PROOF.

Part 1. We will employ a reduction from the PFA emptiness problem similar to the one given at the beginning of the proof of Theorem 10.2, part (1.). However, that construction requires modifications because in the resulting RMDPs of that construction, the controller always has a strategy which never terminates (at any exit), by choosing to go forever deeper into boxes with probability 1. We provide a revised reduction which avoids this problem.

Again, let $M = (V, \Sigma, T, v_1, v_n)$ be a PFA with n states. Define a RMDP A that has two components (call them A' and A) each with a single entry, and such that A' has entry en' and has 2 exits t'_1 and t'_2, and A has entry en and $n+2$ exits $ex_1, \ldots, ex_n$,t_1, and t_2.

We start at entry en' of component A', and A' has a very simple structure: it contains a single box b' mapped to A, and there is a transition with probability 1 from en' to the call port (b', en). There is a probability 1 transition from the return port (b', t_1) to the exit t'_1 of A', and a probability 1 transition from the return port (b', t_2) to exit t'_2. There is a probability 1 transition from the return port (b', ex_n) to t'_1, and there are probability 1 transitions from each of the return ports (b', en_i) to t'_2, where $1 \leq i \leq n-1$. That completes the description of component A'.

Next we describe the structure of component A. The entry en is a probabilistic node, which has three probabilistic transitions: a transition with probability 1/2 to a node z, a second with probability $p/2$ to the exit t_1 and a third with probability $(1-p)/2$ to t_2, where we define $p = 1 - r$.

The rest of component A looks very similar to the component A described at the beginning of the proof of part (1.) of Theorem 10.2, except that node z now plays a role analogous to the entry of the component there. Namely, node z is controlled by the minimizer, and it has edges to the call ports of a set of $|\Sigma|$ boxes $b_a, a \in \Sigma$; all the boxes are mapped to A. In addition, node z has an edge to the exit ex_1. The return ports of the boxes b_a are probabilistic vertices. Each return port (b_a, ex_i) has an edge to each exit ex_j with probability $T_a[i,j]$. This concludes the definition of A. Note that both components of this RMDP are linearly-recursive.

We claim that there is a word w such that $P_M(w) > p$ if and only if $q^*_{(en,t'_2)} < (1-p)$. To see this, first note that we can associate to any (deterministic) strategy of the min player in this RMDP a unique string over the alphabet Σ, such that its choices along any trajectory are consistent with a prefix of this string.[9] Now, if the min player uses the strategy corresponding to the reverse string w^R (in other words, the sequence of choices it makes along any trajectory are consistent with a prefix of w^R), then the probability of termination at t'_2 is

$$(1-p)(1-1/2^{|w|+1}) + (1-P_M(w))/2^{|w|+1}$$

The first term is contributed by the probability that the game stops prematurely and exits at some t_2 (and thus terminates at t'_2), before we reach the end of the string w^R. (This can happen because from the entry of component A we terminate immediately with probability $1/2$, and we split that probability of termination between t_1 and t_2, exiting at t_2 with probability $(1-p)/2$. Thus the total probability of terminating prematurely at t'_2 is $(1-p)(\sum_{i=1}^{|w|} 1/2^i) = (1-p)(1-1/2^{|w|+1})$). The second term is the contribution from the case when the strategy gets to choose the entire string w^R to the end and then follow the transition to the exit ex_1. This only happens with probability $1/2^{|w|+1}$, and thereafter we terminate at t'_1 with probability $P_M(w)$ and at t'_2 with the remaining probability $(1-P_M(w))$. Thus, if $p^*_M = \sup\{P_M(w) \mid w \in \Sigma^*\} \leq p$, then for all strings w the probability of terminating at t'_2 is $\geq 1-p$. Note also that if the min player tries to play forever, then the probability of terminating at t'_2 is exactly $1-p$, and the probability of terminating at $t'_1 = p$.

On the other hand, if $P_M(w) > p$, then the min player in the RMDP can just play the strategy consistent with w^R. In this case the probability of terminating at t'_2 is $< 1-p$, because $(1-P_M(w)) < (1-p)$ and therefore $(1-p)(1-1/2^{|w|+1}) + (1-P_M(w))/2^{|w|+1} < (1-p)$.

Now, the known undecidability results about PFAs ([Blondel and Canterini 2003; Madani et al. 2003]) yield easily that deciding whether $p^*_M > p$, for any fixed $p \in (0,1)$, is undecidable for PFAs with a fixed bounded number of states (independent of p). This completes the proof of part (1.), since we can choose any $p \in (0,1)$ and we have defined $r = (1-p)$.

Part 2. We wish to show that the qualitative termination problem is also undecidable. We do this again in a way similar to the proof of part (2.) of Theorem 10.2, by using the 1-RMC in Figure 5 as a gadget on top of the RMDP from part (1.).

[9]We could also consider randomized strategies, but by Theorem 2.1, part (2.), we do not need to consider them. In fact, we could easily show directly that randomization doesn't help in the present context by the same kind of averaging argument as done the proof of Theorem 10.2, part (1.).

That is, we construct an RMDP with three components A'', A', A. The components A' and A are exactly the same as those of part 1 of this theorem. The new component A'' has a single entry en'' and single exit ex'' and has a structure similar to the component shown in Figure 5, except that the entry does not have a direct transition to the box b_1, and we have an additional box b_0, mapped to A'. The entry en'' of A'' is has an edge with probability 1 to the call port (b_0, en') of the box b_0. The return port (b_0, t'_2) of b_0 has a probability 1 edge to the exit ex'' of A'', while the other return port (b_0, t'_1) has a probability 1 edges to the entry of a box b_1 mapped to A''. The return port (b_1, ex'') of b_1 has a probability 1 edge to another box b_2 mapped also to A'' and the return port (b_2, ex) of b_2 has a probability 1 edge to the exit ex of A_1. This concludes the definition of the new minimizing RMDP.

We claim that $p^*_M > 1/2$ if and only if $q^*_{(en'',ex'')} < 1$, i.e., if and only if there exists a strategy for the minimizer with which we do not terminate with probability 1 starting at the entry en'' of component A''. To see this note that, by the construction in part (1.), starting at en' in A', using any strategy, τ, for the minimizer we always terminate with total probability 1 at either t'_1 or t'_2, i.e., $q^{*,\tau}_{(en',t'_1)} + q^{*,\tau}_{(en',t'_2)} = 1$. Furthermore, $p^*_M > 1/2$ if and only if there is a strategy τ for the minimizer such that $q^{*,\tau}_{(en',t'_2)} < 1/2$. But note that the structure of A'' ensures that for any strategy τ, $q^{*,\tau}_{(en',t'_2)} < 1/2$ if and only if $q^{*,\tau}_{(en'',ex'')} < 1$. Thus it is undecidable to determine whether there exists such a strategy τ, i.e., to decide whether $q^*_{(en'',ex'')} = 1$. □

We next show undecidability of the model checking problem for 1-RMDPs with respect to ω-regular properties specified by a property automaton (or a Linear Temporal Logic formula). We summarize briefly the main notions for model checking problems. Let A be a given RMDP or RSSG with vertex set Q, let Σ be a finite alphabet of 'labels', and let $\mathcal{L} : Q \mapsto \Sigma$, be a Σ-labelling of the vertices of A. $\mathcal{L}$ naturally extends to the state set V of the infinite RMDP or RSSG M_A corresponding to A, by letting $\mathcal{L}(\langle \beta, v \rangle) = \mathcal{L}(v)$ for each state $\langle \beta, v \rangle \in V$ of M_A. Considering every terminating state of M_A (i.e., state $\langle \epsilon, ex \rangle$ where ex is an exit) as an absorbing state (i.e., with a self-loop with probability 1), the labeling further generalizes to a mapping $\mathcal{L} : V^\omega \mapsto \Sigma^\omega$ from trajectories of M_A, i.e., executions (paths) of A, to infinite Σ-strings: for $t = s_0 s_1 s_2 \ldots \in V^\omega$, $\mathcal{L}(t) = \mathcal{L}(s_0)\mathcal{L}(s_1)\mathcal{L}(s_2)\ldots$. A property specifies a subset of 'good' Σ-strings; a trajectory satisfies the property if it maps to a string in the subset. For example, we can express the property that A terminates at exit ex by having $\mathcal{L}$ map ex to letter a, map all other vertices to letter b, and specify as 'good' strings over the alphabet $\Sigma = \{a, b\}$ all the strings that end in an infinite sequence of a's, i.e., the set represented by the regular expression $(a + b)^* a^\omega$.

Regular properties (usually called ω-regular for infinite strings) are the most important class of properties, and are usually expressed by finite automata or in Linear Temporal Logic (LTL can only express a proper but important subset of ω-regular properties). A *Büchi automaton* $B = (\Sigma, S, q_0, R, F)$, has a finite alphabet Σ, a finite 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 $\pi = q_0 a_0 q_1 a_1 q_2 \ldots$ of alternating states and letters, starting at the initial state q_0 of B, such that for all $i \geq 0$ $(q_i, a_i, q_{i+1}) \in R$. The ω-word associated with run π is $w_\pi = a_0 a_1 a_2 \ldots \in \Sigma^\omega$. The run π is *accepting* if for infinitely many i, $q_i \in F$. The language $L(B)$ of the automaton B is $L(B) = \{w_\pi \mid \pi \text{ is an accepting run of } B\}$. Note that $L(B) \subseteq \Sigma^\omega$. For example, an automaton B for the expression $(a + b)^* a^\omega$ (i.e. strings of a's and b's that end with an infinite number of a's), is the automaton with two states q_0, q_1, initial state q_0, accepting set $F = \{q_1\}$, and a transition relation R where state q_0 has transitions to itself on

inputs a, b, transition to q_1 on a, and state q_1 has a translation to itself on a, but has no transition on b.

Given a RSSG A with a labeling $\mathcal{L}$ and automaton B, an execution t of A *satisfies* the property specified by the automaton B iff $\mathcal{L}(t) \in L(B)$. Let u be a given initial vertex for A, i.e., $s_0 = \langle \epsilon, u \rangle$ is the initial state of M_A. A pair of strategies, σ, τ of the players of A induce the infinite state Markov chain $M_A^{u,\sigma,\tau}$; the set of trajectories t of this chain that satisfy property B (i.e. such that $\mathcal{L}(t) \in L(B)$) is measurable, and we let $P_A^{\sigma,\tau}(L(B))$ be its probability. There is again determinacy in the game, i.e., $\sup_\sigma \inf_\tau P_A^{\sigma,\tau}(L(B)) = \inf_\tau \sup_\sigma P_A^{\sigma,\tau}(L(B))$ and we let $P_A(L(B))$ be the common value, which is the value of the game. As in the case of termination problems, we can define more generally the quantitative and qualitative model checking problems for a given RMDP or RSSG with respect to a given property (automaton) B. For example, given a maximizing RMDP A with a given initial vertex u, and given an automaton B, is $P_A(L(B)) = 1$? That is, does the max player have a strategy that results with probability 1 in an execution that satisfies the property B? We show that there is no algorithm that answers these questions for general properties, even for 1-RMDPs.

THEOREM 10.4. *The qualitative and quantitative model checking problems for both maximizing and minimizing 1-RMDPs are undecidable, and in fact there is a fixed property (expressed by a fixed Büchi automaton or LTL formula) for which this holds. Furthermore, the value cannot be approximated within any nontrivial additive constant factor.*

PROOF. We reduce from the termination problems for multi-exit maximizing RMDPs. Let A be a given RMDP as in Theorem 10.2 with a special entry en and exit ex_k, for which we want to distinguish between the case (i) that there is a strategy such that A starting at en terminates at ex_k with probability 1, and the case (ii) that the supremum probability of termination at ex_k is $\leq \epsilon$; the RMDP has k exits, where k is a constant that depends on ϵ. We can assume that A has only one component.

Construct a new RMDP A' which is the same as A, except that the old exits are now ordinary probabilistic nodes with a probability 1 transition to a new exit node ex' which is the only exit node, the old return ports of the boxes are now ordinary nodes, and the unique new return port of each box (corresponding to the new exit node) is a player node with edges to the old return ports. Let $ex_1, \ldots, ex_k$ be the (nodes corresponding to the) old exits nodes. The labeling function $\mathcal{L}$ maps the nodes corresponding to the old exits $ex_1, \ldots, ex_k$ to letters $\alpha_1, \ldots \alpha_k$, it maps the nodes corresponding to the old return ports of the form (b, ex_i) to letter β_i for $i = 1, \ldots, k$, it maps the new exit ex' to letter γ, and maps every other vertex to 0. Thus, the alphabet is $\Sigma = \{\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k, \gamma, 0\}$.

The property is described by the regular expression $E = [0^*(\alpha_1(\gamma+0)^*\beta_1 + \ldots + \alpha_k(\gamma+0)^*\beta_k)]^*0^*\alpha_k\gamma^\omega$ (the subexpressions $(\gamma+0)^*$ could be replaced also by $\gamma 0$). An automaton B that recognizes this language is the following: B has states $q_0, \ldots, q_k, f$, the initial state is q_0, the accepting set is $F = \{f\}$, and the transitions are as follows: state q_0 has a transition to itself on input 0, and a transition to q_i on input α_i for $i = 1, \ldots, k$; state q_i for $i = 1, \ldots, k$ has transitions to itself on input 0 and γ, and a transition to q_0 on input β_i; in addition q_k has a transition to f on input γ; state f has a transition to itself on input γ. Thus, the automaton B has $O(k)$ states and transitions. The property can be expressed easily also by an LTL formula φ of length $O(k)$. Note that once we fix a value for ϵ (for example $\epsilon = 1/4$), the value of k is fixed, and the property is fixed.

The max player in the RMDP A' has choices at the same nodes as A, and in addition has a choice every time the trajectory visits the return port (b, ex') of a box b. In the latter case, if the trajectory exited the component in the last recursive call through the edge (ex_i, ex') and the max player does not choose the edge to the matching old return port (b, ex_i) then the trajectory will be rejected by the automaton B no matter

what happens from that point on; thus, the optimal choice for the max player is to move always to the node for the matching old return port (b, ex_i). Such a strategy σ' for the max player in A' corresponds 1-to-1 to a strategy σ in A, and there is a 1-1 correspondence between the generated trajectories, which has the property that a trajectory in A terminates at exit ex_k iff the corresponding trajectory in A' satisfies the property B (i.e., is mapped by $\mathcal{L}$ to a string accepted by B). Thus, (i) if the max player has a strategy in A that terminates at exit ex_k with probability 1, then it has a strategy in A' that generates with probability 1 a trajectory that satisfies B, and $P_{A'}(L(B)) = 1$, and (ii) if the supremum probability of termination at ex_k in A is $\leq \epsilon$ then $P_{A'}(L(B)) \leq \epsilon$.

The above reduction proves the theorem for maximizing 1-RMDPs for the qualitative and the witness qualitative problem. By the same reduction, it follows that it holds also for the quantitative problems even for linear 1-RMDPs. To show the theorem for minimizing 1-RMDPs, simply consider the complementary property. Recall that ω-regular properties are closed under complementation, thus we can construct a Büchi automaton B' that accepts all strings not accepted by B (and in the case of LTL specifications, we can negate the LTL formula φ for the property). Let A'' be the minimization RMDP which is the same as A' but with all the player vertices being now min vertices. Then $P_{A''}(L(B')) = 1 - P_{A'}(L(B))$. Thus, (i) if $P_{A'}(L(B)) = 1$ then $P_{A''}(L(B')) = 0$ and there is a strategy for the min player that achieves value 0, and (ii) if $P_{A'}(L(B)) \leq \epsilon$ then $P_{A''}(L(B')) \geq 1 - \epsilon$. □

11. CONCLUSIONS

We have defined and studied RMDPs and RSSGs. These are classes of finitely presented infinite-state MDPs, and turn-based stochastic games, which augment ordinary MDPs and stochastic games with a recursive feature. These models subsume controlled and 2-player game versions of several classic stochastic processes, and are natural models of probabilistic procedural programs with recursion.

We have studied the decidability and computational complexity of some key problem for RMDPs, RSSGs, and for several of their important subclasses, centered around the goal of maximizing/minimizing the probability of termination. We have provided algorithms for a number of problems, including polynomial time algorithms for qualitative termination problems for both maximizing and minimizing 1-RMDPs. We have shown that for 1-RMDPs and 1-RSSGs, the players possess optimal deterministic *stackless and memoryless* optimal strategies. We have shown that the quantitative problems for 1-RSSGs can be decided in PSPACE by exploiting corresponding monotone systems of nonlinear min/max equations which capture the value vector of 1-RSSGs, and we have observed based on our earlier work on 1-RMCs that this complexity can not be improved upon substantially without resolving major open problems in numerical computation, namely the square-root sum problem and key arithmetic circuit decision problems. For general RMDPs and RSSGs, we have shown that things get much harder: both quantitative and qualitative termination problems are undecidable.

Since the publication of the two conference papers on which this paper is based [Etessami and Yannakakis 2005; 2006], there have been a series of follow-up papers by ourselves and others, building on this work and extending it in several directions. We now mention some of this follow-up work:

— As mentioned earlier, in [Etessami and Yannakakis 2008] we extended RSSGs to Recursive Concurrent Stochastic Games (RCSGs), which are no longer turn-based (perfect information) games. At each state, two adversarial players choose actions simultaneously and independently, and their joint actions determine the probability distribution on the next state. This matches the more standard imperfect infor-

mation model of stochastic games, e.g., as originally formulated by Shapley (see, e.g., [Shapley 1953; Filar and Vrieze 1997; Neyman and Sorin 2003]). We studied both quantitative and qualitative termination problems for 1-RCSGs in [Etessami and Yannakakis 2008], and we showed that, like 1-RSSGs, these are decidable in PSPACE, via an extension of the monotone nonlinear min/max equations for the value vector of 1-RSSGs to monotone nonlinear *minimax* equations for RCSGs (where by minimax equation we mean equations which use an operator yielding the minimax value of a one-shot zero-sum matrix game). We showed that for a 1-RCSG termination game the minimizer has randomized SM optimal strategy, and the maximizer has, for every $\epsilon > 0$ randomized SM ϵ-optimal strategies. (Randomization is in general needed by both players.) We showed that the qualitative termination problem for 1-RCSGs is square-root-sum-hard and thus can not be placed in NP $\cap$ coNP (as we have done in this paper for 1-RSSGs) without resolving a major open problem in numerical computation. (In fact, in [Etessami and Yannakakis 2008; 2010] we showed that even for finite-state concurrent stochastic games with reachability objectives, as well as for Shapley's finite-state stochastic games (with discounted reward objectives), the quantitative decision problems are SQRT-SUM-hard.)

— In [Etessami et al. 2008] we have extended RMDPs and RSSGs to a reward setting with strictly positive rewards on all transitions, where the goal of the players is to maximize/minimize the total expected reward (which may be ∞). Such a reward model is very natural, e.g., for modeling the optimal/pessimal expected running time of a procedural program (where every transition, i.e., step of the program, takes some non-zero amount of time). Interestingly, we have shown that in the setting with strictly positive rewards the *quantitative* problems of computing the value for 1-RMDPs and 1-RSSGs actually become easier than in the setting of optimizing termination probability. Namely, they are in P-time and in NP $\cap$ coNP, respectively. This is in essence because we are able to show that certain monotone linear min/max optimality equations can be used to capture the value vector in the positive reward setting.

  On the other hand, when 1-RMDPs and 1-RSSGs are augmented with non-negative rewards (i.e., when reward 0 is allowed on transitions), we do not even know whether the quantitative problems are decidable. In particular, is the following problem decidable: given a 1-RMDP with non-negative rewards, and total expected reward criterion, and given a rational value r, decide whether the optimal (supremum or infimum) expected reward (over all strategies) is $> r$.

— Building on the polynomial time algorithms given in this paper (Section 6) for the qualitative termination problems for (maximizing and minimizing) 1-RMDPs, Brázdil et al. [2006] show that the following qualitative problem is decidable in polynomial time: Given a 1-RMDP, decide whether there exists a *witness* strategy under which we will reach a desired target vertex (which may not be an exit) in *any* calling context (i.e., under any call stack, not just the empty stack), with probability 1 (or with probability < 1). (Their results are phrased in terms of pBPAs, which are essentially controlled SCFGs with leftmost derivation law, but as discussed at the end of Section 9, these are equivalent to 1-RMDPs. In another subsequent paper, Brázdil et al. [2011] have also extended the witness reachability result from [Brázdil et al. 2006], to certain witness reachability results in the setting of two-player 1-RSSGs.) Their algorithm uses our algorithm for the (maximizing and minimizing) 1-RMDP qualitative termination problem as a subroutine, and uses an iterative fixed point algorithm on top of it which is independent of the probabilities labeling transitions in the 1-RMDP, i.e., only depends on its structure.

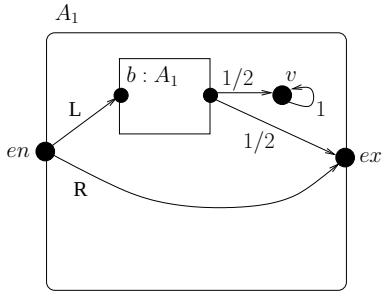

Fig. 8. *Maximizing reachability probability at v in a 1-RMDP:* no optimal strategy exists.

Note that in the maximization case there may not exist any witness strategy that achieves probability 1, and yet the supremum value may be 1. This is illustrated by Figure 8, which is a very slight variation on our earlier 2-exit example in Figure 3. As can easily be seen, in this example there is no optimal strategy for reaching vertex v (in any calling context), but strategy L^nR guarantees that we reach v with overall probability $(1 - 1/2^n)$ (i.e., this is the sum total probability of reaching v in any calling context). Thus the supremum value we can achieve is 1. (In fact, the example in Figure 3 is just this example, but with an extra exit added to "raise an exception" and terminate completely at the new exit whenever we reach v in any context.) Surprisingly, we do not know whether it is decidable to determine whether, for a given 1-RMDP and distinguished vertex v, this supremum value is 1 (while the witness problem is solvable in P-time). In fact, it should be noted that this open problem is easily reducible to the earlier mentioned quantitative problem for 1-RMDPs with non-negative rewards: place 0 reward on all transitions except those entering vertex v, which have reward 1, and remove all outgoing transitions from v. The expected total reward under any strategy is then the probability of reaching v (in any calling context).

— A software tool called PReMo is described in [Wojtczak and Etessami 2007] which implements a number of analysis algorithms for both RMCs and for 1-RMDPs and 1-RSSGs. Included among these are optimized versions of a basic value iteration scheme for computing optimal termination values for 1-RMDPs and 1-RSSGs, which uses iteration on the corresponding monotone nonlinear-min-max equations (the "Bellman equations" in this setting) to converge monotonically from below to the least fixed point solution which, as shown in this paper, captures their vector of termination values. In [Esparza et al. 2008] the authors subsequently considered the termination problem for 1-RMDPs and 1-RSSGs, and devised a more sophisticated iterative "strategy improvement" procedure for numerically converging to these values. Their method combines in an interesting way aspects of both the (decomposed) Newton's method we devised in [Etessami and Yannakakis 2009] for computing termination probabilities for RMCs, together with standard strategy improvement and policy iteration methods used frequently for analyzing finite-state MDPs and SSGs. However, [Esparza et al. 2008] provided no upper bounds for the performance of their algorithms.

— A different subclass of RMDPs and RSSGs, the *1-box* subclass, which corresponds to controlled/game versions of another well-studied class of stochastic processes called (discrete-time) *Quasi-Birth-Death processes* (QBDs), as well as to probabilistic/game extensions of 1-counter automata, was studied recently in a series of papers [Etessami et al. 2010; Brázdil et al. 2010a; Brázdil et al. 2010b; Brázdil et al. 2011] In [Etessami et al. 2010] it was observed that 1-box RMCs restricted to having at

most one box in every component (equivalently, probabilistic Pushdown systems restricted to one symbol in the stack alphabet, or probabilistic 1-counter automata), are equivalent to discrete-time Quasi-Birth-Death processes (QBDs), a stochastic model that is heavily studied in queueing theory and performance evaluation. Controlled and game versions of QBDs can also naturally be viewed as restrictions of RMDPs and RSSGs to the 1-box case, The complexity and decidability of the analogous qualitative and quantitative termination problems for these models equivalent to 1-box RMDPs and 1-box RSSGs were studied in [Brázdil et al. 2010a; Brázdil et al. 2010b; Brázdil et al. 2011], where it has been shown that both qualitative problems as well as certain quantitative *approximation* problems are decidable. However, a number of problems remain open about both the precise computational complexity and decidability of some problems for 1-box RMDPs and 1-box RSSGs.

— Most recently, in a series of papers with Alistair Stewart [Etessami et al. 2012a; 2012b], we have substantially extended the results established in this paper about 1-RMDPs and 1-RSSGs. In particular, in [Etessami et al. 2012b], we have shown that, given a 1-RMDP (or Branching MDP) where the objective is either to maximize or minimize the optimal termination (respectively, extinction) probability, and given a rational error threshold $\epsilon > 0$, there is an algorithm that approximates the optimal value of the 1-RMDP to within additive error $\epsilon > 0$, and runs in time polynomial in both the encoding size of the 1-RMDP, and in $\log(1/\epsilon)$.
The P-time approximation algorithm in [Etessami et al. 2012b] for 1-RMDPs is based on a *Generalized Newton's Method* (GNM), applied to the Bellman optimality equations, $x = P(x)$, of a 1-RMDP, that we have defined and studied in this paper. GNM is an iterative algorithm that in each iteration requires solving a certain linear programming problem, and then performing suitable rounding on the resulting numbers. We show in [Etessami et al. 2012b] that, after first preprocessing the Bellman equations $x = P(x)$ of the 1-RMDP in order to remove all those variables x_i such that the LFP value q_i^* is either 0 or 1, by then applying iterations of GNM starting at the all 0 vector, we can approximate the LFP vector q^* in P-time (in the standard Turing model of computation). Thus, the results of [Etessami et al. 2012b] build directly on the results established in this paper for P-time qualitative termination analysis of 1-RMDPs, and more importantly they also build on and extend our results regarding variants of Newton's method applied to purely probabilistic RMCs ([Etessami and Yannakakis 2009]) and to 1-exit RMCs ([Etessami et al. 2012a]). In [Etessami et al. 2012b] it is also shown that one can compute ϵ-optimal stackless memoryless strategies in P-time, given a (maximizing or minimizing) 1-RMDP. As noted in [Etessami et al. 2012b], it follows readily from these results that the problem of approximating the termination value for 1-RSSGs is in FNP.

There are many other directions for further research and many open problems related to the models discussed in this paper and their extensions. In particular, RMDPs and RSSGs, and their various subclasses, can be studied under some of the more standard objectives studied in the MDP and stochastic game literature, such as discounted and limiting-average rewards.

APPENDIX

Proof of Theorem 2.1

We now phrase and prove Theorem 2.1 in a slightly stronger form than the way it was phrased in the body of the paper.

THEOREM A.1 (REPHRASING OF THEOREM 2.1). *Suppose* $G = (V = V_0 \cup V_1 \cup V_2, \Delta, \mathtt{pl}, r)$ *is a countable-state, turn-based (perfect-information) stochastic game, with*

a non-negative reward function r on transitions, and with expected total reward objective (which player 1 wants to maximize and player 2 wants to minimize). Suppose furthermore that G is finitely branching*, meaning that for any state $u \in V$ there are a finite number of transitions of the form (u, x, v) in Δ (regardless of whether u is a player's state or a probabilistic state). Then:*

(*1*) *There exists a deterministic memoryless strategy τ^* for player 2 (the minimizer), which is optimal starting from any state $u \in V$ of the game G.*
(*2*) *Suppose that, in addition, it holds that under all pairs of strategies used by the two players, the expected total reward is bounded by a fixed constant K.*
Then for every $\epsilon > 0$, and for any state $u \in V$, player 1 has an ϵ-optimal deterministic memoryless strategy in the game G starting at u.
In particular, such countable-state perfect information stochastic games are (deterministically) memorylessly determined.

PROOF. As mentioned earlier, this Theorem is a generalization of well-known results in the MDP literature to the setting of turn-based countable-state stochastic games. The proof we give is also quite similar to a proof of a related result (about 1-RSSGs with strictly positive rewards) given in [Etessami et al. 2008], which was a follow-up to (the conference versions of) this paper.

Before we show (1.), namely, that player 2 (the minimizer) has an optimal deterministic memoryless strategy in G, we proceed to establish more basic facts about such countable-state stochastic games in a way which is very similar to the proof of Theorem 3.2, and Corollary 4.1. We associate optimality (or *value*) equations, on countably many variables, with the countable-state stochastic game G. Specifically, the equations have the following form. For each state $u \in V$, we have a variable x_u. Let $N(u) = \{v \in V \mid \exists (u, x, v) \in \Delta\}$ denote the set of "neighbor" states of u. For $u \in V$ and $v \in N(u)$, let $r(u, v) \geq 0$ denote the non-negative reward labeling the (unique) transition from u to v. Furthermore, for probabilistic vertices $u \in V_0$ and $v \in N(u)$, let $p_{u,v} > 0$ denote the positive probability such that $(u, p_{u,v}, v) \in \Delta$.[10]

$$\begin{array}{ll} x_u = \sum_{v \in N(u)} p_{u,v} \cdot (r(u,v) + x_v) & \text{for all } u \in V_0, \\ x_u = \max_{v \in N(u)} (r(u,v) + x_v) & \text{for all } u \in V_1, \\ x_u = \min_{v \in N(u)} (r(u,v) + x_v) & \text{for all } u \in V_2; \end{array} \tag{1}$$

This yields a system of equations $x = Q(x)$ in countably many variables $(x_0, x_1, x_2, \ldots)$. Let $\overline{\mathbb{R}}_{\geq 0} = [0, \infty]$ denote the non-negative extended reals. Let $\overline{\mathbb{R}}^{\omega}_{\geq 0}$ denote the set of infinite-dimensional extended-real vectors $(r_0, r_1, r_2, \ldots)$ with coordinates indexed by the non-negative integers (i.e., by the set ω). The extended reals are totally ordered in the obvious way. We assume the following usual arithmetic conventions on the non-negative extended reals $\overline{\mathbb{R}}_{\geq 0}$: $a \cdot \infty = \infty$, for any $a \in \overline{\mathbb{R}}_{>0}$; $0 \cdot \infty = 0$; $a + \infty = \infty$, for any $a \in \overline{\mathbb{R}}_{\geq 0}$.

It is clear that $Q(x)$ defines a monotone map from $\overline{\mathbb{R}}^{\omega}_{\geq 0}$ to itself, because the right hand side $Q_i(x)$, of each equation, $x_i = Q_i(x)$, has only non-negative coefficients and

[10] We are assuming, w.l.o.g., that if there is a transition from u to v in Δ, it is unique. To see that this is w.l.o.g., note that for any transition $e = (u, x, v)$ in Δ we can, if necessary, add an auxiliary intermediate state s_e, remove transition e and add two new transitions: $e_1 = (u, x, s_e)$ and $e_2 = (s_e, 1, v)$, with rewards on these transitions given by: $r(e_1) := r(e)$, and $r(e_2) := 0$. This makes sure transitions between states are unique, and furthermore it is clear that it yields a suitably "equivalent" stochastic game with non-negative rewards (with a 1-1 correspondence between strategies of the new and original game, which yield identical expected total rewards).

constant terms, and because $\min$ and $\max$ are monotone operators. Thus $0 \leq a \leq b$, for any vectors $a, b \in \overline{\mathbb{R}}^{\omega}_{\geq 0}$, implies that $0 \leq Q(a) \leq Q(b)$.

Let Ψ_i denote the set of all strategies for player i. A pair of strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$ induces in a straightforward way a countable-state Markov chain $M_G^{\sigma,\tau} = (V^*, \Delta')$, whose set of states is the set V^* of histories. Let $r_u^{k,\sigma,\tau}$ denote the expected total reward obtained in k steps in $M_G^{\sigma,\tau}$, starting at initial state u. When $k = 0$, we have $r_u^{0,\sigma,\tau} = 0$, by definition. Let $r_u^{*,\sigma,\tau} \doteq \lim_{k\to\infty} r^{k,\sigma,\tau}$. Thus $r_u^{*,\sigma,\tau}$ is the expected total reward, starting at u, of the infinite-horizon game, assuming the two players use strategies σ and τ, respectively. Later, in order to show that player 1 (maximizer) has ϵ-optimal strategies for all $\epsilon > 0$, we will need to assume that $r^{*,\sigma,\tau} < \infty$ for all strategies σ and τ. However, for now, we can allow for the possibility that the limit may diverge and thus that $r^{*,\sigma,\tau} \in [0, \infty]$.

Let $r_u^* \doteq \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau}$. We will establish that these games are determined, and that thus r_u^* defines the *value* of the game starting at u. For every state $u \in V$, let $r_u^k \doteq \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} r_u^{k,\sigma,\tau}$ denote the value of the k-step game starting at vertex u, where player 1 aims to maximize total reward and player 2 tries to minimize it. It is clear that this k-step game is a finite game, by virtue of the fact that G is finitely branching. Thus, by classic facts about finite extensive form games, the k-step game is determined, i.e., has a value, meaning that $r_u^k = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} r_u^{k,\sigma,\tau}$. Let $x^0 := \mathbf{0}$ and let $x^{k+1} := Q(x^k)$.

The following lemma is an analog of Theorem 3.2, which we established for finitely presented 1-RSSG, but now in the context of arbitrary countable state turn-based stochastic games with non-negative rewards.

LEMMA A.2.

(*1*) *The map* $Q : \overline{\mathbb{R}}^{\omega}_{\geq 0} \to \overline{\mathbb{R}}^{\omega}_{\geq 0}$ *is monotone on* $\overline{\mathbb{R}}_{\geq 0}$ *and* $\mathbf{0} \leq \mathbf{x}^k \leq \mathbf{x}^{k+1}$ *for* $k \geq 0$.
(*2*) $\mathbf{r}^* = P(\mathbf{r}^*)$.
(*3*) *For all* $k \geq 0$, $\mathbf{x}^k = \mathbf{r}^k \leq \mathbf{r}^*$.
(*4*) *For all* $\mathbf{r}' \in \overline{\mathbb{R}}^{\omega}_{\geq 0}$, *if* $\mathbf{r}' = P(\mathbf{r}')$, *then* $\mathbf{r}^* \leq \mathbf{r}'$.
(*5*) *For all vertices* u,

$$r_u^* \doteq \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} r_u^{*,\sigma,\tau} = \inf_{\tau\in\Psi_2} \sup_{\sigma\in\Psi_1} r_u^{*,\sigma,\tau}.$$

(In other words, these games are determined.)
(*6*) $\mathbf{r}^* = \lim_{k\to\infty} \mathbf{x}^k$.

Although the proof of this Lemma is very similar to that of Theorem 3.2, there are some differences, so we provide a detailed proof for completeness.

PROOF. (1.) is self-evident. To see why (2.) holds, we consider each kind of vertex in turn. Suppose $u \in V_0$. Then it is easy to see that $r_u^* = \sum_{v\in N(u)} p_{u,v} \cdot (r(u,v) + r_v^*)$, by definition of r_u^*, because the supremum expected payoff for player 1 starting at u is, with probability $p_{u,v}$, equal to $r(u,v)$ plus the supremum expected payoff starting at successor v. Likewise, for $u \in V_1$, then $r_u^* = \max_{v\in N(u)}(r(u,v) + r_v^*)$. This is because the supremum expected payoff starting at u is obtained by choosing the successor v such that $r(u,v) + r_v^*$ is maximized (note that there are only finitely many neighbors, because we assume the game is finitely branching). Finally, for $u \in V_2$, again $r_u^* = \min_{v\in N(u)}(r(u,v)+r_v^*)$, because this is the supremum total reward that can be obtained starting at u, given that player 2 aims to minimize the total reward.

Next, we prove (3.) by induction on k. Note that Q is monotonic, and $\mathbf{r}^*$ is a fixed point of Q. Since $x^0 = \mathbf{0} \leq \mathbf{r}^*$, it follows by induction on k that $\mathbf{x}^k \leq \mathbf{r}^*$, for all $k \geq 0$.

We also show by induction on k that $x^k = r^k$. For $k = 0$, the claim is obvious, because the expected total reward of the 0-step game starting at any vertex is 0.

Suppose, by inductive assumption, that $x^k = r^k$ holds for some $k \geq 0$. We establish that it holds for $k+1$. For any vertex u, consider x_u^{k+1}. There are 3 cases to consider:

(1) $u \in V_0$. In this case $x_u^{k+1} = Q_u(x^k) = \sum_{v \in N(u)} p_{u,v} \cdot (r(u,v) + x_v^k) = \sum_{v \in N(u)} p_{u,v} \cdot (r(u,v) + r_v^k) = r_u^{k+1}$. The last equality holds by definition of optimal expected total reward in $k+1$ steps, starting at $u \in V_0$.
(2) $u \in V_1$. In this case $x_u^{k+1} = Q_u(x^k) = \max_{v \in N(u)}(r(u,v) + x_v^k) = \max_{v \in N(u)}(r(u,v) + r_v^k) = r_v^{k+1}$. Again, the last equality holds by definition of optimal expected total reward in $k+1$ steps, starting at $u \in V_1$. (Note that max suffices instead of sup, because the stochastic game is assumed to be finitely branching.)
(3) $u \in V_2$. Again, $x_u^{k+1} = Q_u(x^k) = \min_{v \in N(u)}(r(u,v) + x_v^k) = \min_{v \in N(u)}(r(u,v) + r_v^k) = r_v^{k+1}$. Again, the last equality holds by definition of optimal expected total reward in $k+1$ steps, starting at $u \in V_1$.

This completes the proof of part (3.).

Next we prove part (4.), namely that for $r' \in \overline{\mathbb{R}}^{\omega}_{\geq 0}$, if $r' = Q(r')$ then $r^* \leq r'$.

Consider any fixed point $\mathbf{r}' \in \overline{\mathbb{R}}^{\omega}_{\geq 0}$ of the equation system $x = Q(\mathbf{x})$. Let us denote by τ^* a strategy for the *minimizer* that picks for each vertex the successor with the minimum value in $\mathbf{r}'$, i.e., for each state u, where u belongs to player 2 (*minimizer*), we choose $\tau^*(u) = \arg\min_{v \in N(u)}(r(u,v) + r'_v)$ (breaking ties, say, lexicographically). *Note that the strategy τ^* is well defined, because the game G is finitely branching.*

LEMMA A.3. *For all strategies $\sigma \in \Psi_1$ of player 1, and for all $k \geq 0$, $\mathbf{r}^{k,\sigma,\tau^*} \leq \mathbf{r}'$.*

PROOF. We prove the lemma by induction on k. The base case, $\mathbf{r}^{0,\sigma,\tau^*} = \mathbf{0} \leq \mathbf{r}'$, is trivial. For the induction step we have three cases, according to the type of the vertex u that corresponds to a component of $\mathbf{r}^{k,\sigma,\tau^*}$.

(1) If $u \in V_0$ is a random node, we define a strategy $\sigma'(\theta) = \sigma(u\theta)$ and we get:

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} = \sum_{v \in N(u)} p_{u,v}(r(u,v) + \mathbf{r}_v^{k,\sigma',\tau^*}) \leq \sum_{v \in N(u)} p_{u,v}(r(u,v) + \mathbf{r}'_v) = \mathbf{r}'_u$$

based on the inductive assumption and the fact that $\mathbf{r}'$ is a fixed point of $Q(\mathbf{x})$.

(2) For $u \in V_1$ we claim

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} = \max_{v \in N(u)}(r(u,v) + \mathbf{r}_v^{k,\sigma',\tau^*})$$

because player 1 has to move to some neighbor v of u in one step, and thus it cannot obtain total reward more that $\max_{v \in N(u)} r(u,v) + \mathbf{r}_v^{k,\sigma',\tau^*}$, where σ' is again defined by $\sigma'(\theta) = \sigma(u\theta)$. Thus

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} = \max_{v \in N(u)} r(u,v) + \mathbf{r}_v^{k,\sigma',\tau^*} \leq \max_{v \in N(u)}(r(u,v) + \mathbf{r}'_v) = \mathbf{r}'_u$$

(3) For $u \in V_2$ we know that $\tau^*(u) = \arg\min_{v \in N(u)}(r(u,v) + \mathbf{r}'_v) = v^*$, so:

$$\mathbf{r}_u^{k+1,\sigma,\tau^*} = r(u,v^*) + \mathbf{r}_{v^*}^{k,\sigma',\tau^*} \leq r(u,v^*) + \mathbf{r}'_{v^*} = \min_{v \in N(u)}(r(u,v) + \mathbf{r}'_v) = \mathbf{r}'_u$$

□

Now by the lemma we have $\mathbf{r}_u^{*,\sigma,\tau^*} = \lim_{k\to\infty} \mathbf{r}_u^{k,\sigma,\tau^*} \leq \mathbf{r}'_u$ for every vertex u and for any player 1 strategy σ, so $\sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u$. Thus for all vertices u:

$$\mathbf{r}_u^* = \sup_\sigma \inf_\tau \mathbf{r}_u^{*,\sigma,\tau} \leq \inf_\tau \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau} \leq \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u \tag{2}$$

Next, to prove (5.), in equation (2) above, choose $\mathbf{r}' = \mathbf{r}^*$. Then we have, for all vertices u,

$$\sup_\sigma \inf_\tau \mathbf{r}_u^{*,\sigma,\tau} = \inf_\tau \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau}.$$

Finally, to prove (6.), we know that $\mathbf{z} = \lim_{k\to\infty} \mathbf{x}^k$ exists in $[0,\infty]$, because it is a coordinate-wise monotonically non-decreasing sequence (note some entries may be infinite). In fact we have $\mathbf{z} = \lim_{k\to\infty} Q^{k+1}(\mathbf{0}) = Q(\lim_{k\to\infty} Q^k(\mathbf{0}))$, and thus $\mathbf{z}$ is a fixed point of the equation $x = Q(x)$. So from (4) we have $\mathbf{r}^* \leq \lim_{k\to\infty} \mathbf{x}^k$. Since $\mathbf{x}^k \leq \mathbf{r}^*$ for all $k \geq 0$, $\lim_{k\to\infty} \mathbf{x}^k \leq \mathbf{r}^*$ and thus $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{r}^*$. □

The proof of Lemma A.2 implies the following fact:

COROLLARY A.4. *In such a stochastic game, G, player 2 (the* minimizer*) always has a deterministic memoryless strategy that is optimal regardless which vertex the game starts at.*

PROOF. It is enough to consider the strategy τ^* described in from Part 4 of Lemma A.2, when we let $\mathbf{r}' = \mathbf{r}^*$. In other words, for each vertex $u \in V_2$, $\tau^*(u) = \arg\min_{v\in N(u)}(r(u,v) + \mathbf{r}_v^*)$. Then, by equation (2) above, we have $r_u^* = \sup_\sigma r^{*,\sigma,\tau^*} = \inf_\tau \sup_\sigma r^{*,\sigma,\tau}$. □

This completes the proof of part (1.) of Theorem A.1.

Next we wish to prove part (2.), regarding ϵ-optimal strategies for player 1 (the maximizer). We now crucially need the assumption that under any pairs of strategies used by the two players, the expected total reward is bounded by a fixed constant K. This obviously ensures that the value vector r^* for the game is finite and bounded by K in all coordinates, i.e., $r^* \in [0,K]^\omega$.

We now argue that maximizer has ϵ-optimal deterministic memoryless strategies, for all $\epsilon > 0$.

Note that we have already shown in Lemma A.2 part (3.) and (6.) that the values r^k of the k-step game associated with G, provide underapproximations of the total reward values r^* in the infinite-horizon game, such that for all u, $r_u^* = \lim_{k\to\infty} r_u^k$, and for all k, $r_u^k \leq r_u^{k+1} \leq r_u^*$.

Consider the finite set, $S_k^u \subseteq V$, of states of G that can possibly be encountered during the k-step finite-horizon game, starting at state u. In other words, S_k^u is defined to be the set of all states in V that appear anywhere in the *entire* finite game tree of the k-step game rooted at u (recall that the k-step game is a finite game in extensive form, and thus has an associated finite game tree). Now consider the infinite-horizon game G_k^u induced by the finite set of states S_k^u and an additional state v_{dead}, which is defined as follows. G_k^u proceeds just like the original infinite-horizon game, G, starting at state u, but as soon as a transition leaves S_k^u, the game G_k^u now moves to the extra dead-end state, v_{dead}, from which we have an additional transition $e = (v_{dead}, 1, v_{dead})$ with reward $r(e) = 0$. Thus, we will gain total reward 0 with probability 1 after hitting v_{dead}. The goal of the player 1 (respectively, player 2) in G_k^u is again to maximize (resp. minimize) the expected total reward, over an infinite time horizon. Note that G_k^u defines a finite-state total-reward perfect information stochastic game with *infinite horizon*, and with the property that under all strategies the total expected reward is upper bounded by the same fixed constant K.

Starting at a state u in S_k^u, the game G_k^u clearly has a value z_u^k that is at least the value r_u^k of the k-step game over G starting at u. Furthermore, z_u^k is clearly at most the value r_u^* of the infinite-horizon game. Thus $r_u^k \leq z_u^k \leq r_u^*$. Thus, $\lim_{k\to\infty} r_u^k = r_u^* = \lim_{k\to\infty} z_u^k$.

Furthermore, note that we have already established in Corollary A.4 that player 2, the minimizer, has a deterministic memoryless strategy in G_k^u which is optimal starting from *every* vertex. Let this strategy be τ_u^k. Thus, if we fix the optimal strategy τ_u^k for player 2, we are left with a finite-state infinite-horizon MDP, G_k^{u,τ_u^k}, with non-negative rewards on transitions, and total reward on all trajectories bounded by a fixed constant K, where the goal of the controller is to *maximize* the expected total reward. This is a standard model, known as the *positive bounded model*, which has been studied extensively in the MDP literature. For finite-state positive bounded MDP models, it is well known (see Theorem 7.1.9 of [Puterman 1994]) that the controller (maximizer) has a deterministic memoryless (a.k.a. deterministic stationary) strategy which is optimal starting from every state of the game. It follows that, for every $u \in V$, in the finite-state infinite-horizon two-player stochastic game G_k^u, the maximizer has a deterministic memoryless optimal strategy. Let us call this strategy σ_u^k.

Thus, since the value r_u^k of the k-step game converges to the value r_u^* of the infinite-state game, and since $0 \leq r_u^* \leq K$, and since the values z_u^k of the game G_k^u are sandwiched between r_u^k and r_u^*, i.e., $r_u^k \leq z_u^k \leq r_u^*$, it follows that, starting at any vertex $u \in V$, and for any $\epsilon > 0$, the maximizer has an ϵ-optimal strategy σ_u^ϵ for the game. The strategy σ_u^ϵ consists of choosing a sufficiently large k such that $r_u^* - r_u^k < \epsilon$, and mimicking the maximizer's optimal memoryless strategy σ_u^k in the game G_k^u, whenever it is at a state inside S_k^u, and otherwise playing arbitrarily, but *memorylessly*, outside of S_k^u. Such a strategy clearly forces a value that is at least $z_u^k \geq r_u^k$, and thus a value that is within ϵ of r_u^*. This completes the proof of part (2.) of Theorem A.1. □

REFERENCES

ALLENDER, E., BÜRGISSER, P., KJELDGAARD-PEDERSEN, J., AND MILTERSEN, P. B. 2009. On the complexity of numerical analysis. *SIAM J. Comput. 38,* 5, 1987–2006. Earlier conference version appeared in *Proc. 21st IEEE Conference on Computational Complexity*, 2006.

ALUR, R., BENEDIKT, M., ETESSAMI, K., GODEFROID, P., REPS, T., AND YANNAKAKIS, M. 2005. Analysis of recursive state machines. *ACM Trans. Program. Lang. Syst. 27,* 4, 786–818.

ALUR, R., COURCOUBETIS, C., AND YANNAKAKIS, M. 1995. Distinguishing tests for nondeterministic and probabilistic machines. In *Proc. 20th STOC*. 363–372.

ATHREYA, K. B. AND NEY, P. E. 1972. *Branching processes*. Springer-Verlag.

BAIER, C. AND KWIATKOWSKA, M. 1998. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing 11,* 3, 125–155.

BLONDEL, V. AND CANTERINI, V. 2003. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems 36*, 231–245.

BRÁZDIL, T., BROZEK, V., AND ETESSAMI, K. 2010a. One-counter stochastic games. In *Proc. of FSTTCS'10*. 108–119.

BRÁZDIL, T., BROZEK, V., ETESSAMI, K., AND KUCERA, T. 2011. Approximating the termination value of one-counter mdps and stochastic games. In *Proc. of ICALP'11*.

BRÁZDIL, T., BROZEK, V., ETESSAMI, K., KUCERA, T., AND WOJTCZAK, D. 2010b. One-counter Markov decision processes. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*.

BRÁZDIL, T., BROZEK, V., FOREJT, V., AND KUCERA, A. 2006. Reachability in recursive Markov decision processes. In *Proc. 17th Int. CONCUR*. 358–374.

BRÁZDIL, T., BROZEK, V., KUCERA, A., AND OBDRZÁLEK, J. 2011. Qualitative reachability in stochastic bpa games. *Information & Computation 209,* 8, 1160–1183.

BRÁZDIL, T., KUČERA, A., AND STRAŽOVSKÝ, O. 2005. Decidability of temporal properties of probabilistic pushdown automata. In *Proc. 22nd Symp. on Theoretical Aspects of Comp. Sci. (STACS)*.

CANNY, J. 1988. Some algebraic and geometric computations in PSPACE. In *Proc. 20th ACM Symp. on Theory of Computing (STOC)*. 460–467.

CHATTERJEE, K., DE ALFARO, L., AND HENZINGER, T. A. 2006. The complexity of quantitative concurrent parity games. In *SODA*. 678–687.

CONDON, A. 1992. The complexity of stochastic games. *Inf. & Comp. 96,* 2, 203–224.

CONDON, A. AND LIPTON, R. 1989. The complexity of space bounded interactive proofs. In *Proc. of 30th IEEE FOCS*.

COURCOUBETIS, C. AND YANNAKAKIS, M. 1995. The complexity of probabilistic verification. *Journal of the ACM 42,* 4, 857–907.

COURCOUBETIS, C. AND YANNAKAKIS, M. 1998. Markov decision processes and regular events. *IEEE Trans. on Automatic Control 43,* 10, 1399–1418.

DE ALFARO, L., HENZINGER, T. A., AND KUPFERMAN, O. 2007. Concurrent reachability games. *Theor. Comput. Sci. 386,* 3, 188–217.

DE ALFARO, L., KWIATKOWSKA, M., NORMAN, G., PARKER, D., AND SEGALA, R. 2000. Symbolic model checking of probabilistic processes using MTBDDs and the kronecker representation. In *Proc. of 6th TACAS*. 395–410.

DE ALFARO, L. AND MAJUMDAR, R. 2004. Quantitative solution of omega-regular games. *J. Comput. Syst. Sci. 68,* 2, 374–397.

DENARDO, E. AND ROTHBLUM, U. 2005. Totally expanding multiplicative systems. *Linear Algebra Appl. 406*, 142–158.

DENARDO, E. V. AND ROTHBLUM, U. G. 2006. A turnpike theorem for a risk-sensitive markov decision process with stopping. *SIAM J. on Control and Optimization 45,* 2, 414–431.

EHRENFEUCHT, A. AND MYCIELSKI, J. 1979. Positional strategies for mean payoff games. *Int. J. of Game Theory 8*, 109–113.

EMERSON, E. A. AND JUTLA, C. S. 1991. Tree automata, mu-calculus, and determinacy. In *Proc. of 32nd IEEE Symp. on Foundations of Computer Science (FOCS)*. 368–377.

ESPARZA, J., GAISER, A., AND KIEFER, S. 2013. A strongly polynomial algorithm for criticality of branching processes and consistency of stochastic context-free grammars. *Inf. Process. Lett. 113,* 10-11, 381–385.

ESPARZA, J., GAWLITZA, T., KIEFER, S., AND SEIDL, H. 2008. Approximative methods for monotone systems of min-max-polynomial equations. In *ICALP (1)*. 698–710.

ESPARZA, J., KIEFER, S., AND LUTTENBERGER, M. 2010. Computing the least fixed point of positive polynomial systems. *SIAM J. Comput. 39,* 6, 2282–2335.

ESPARZA, J., KUČERA, A., AND MAYR, R. 2004. Model checking probabilistic pushdown automata. In *Proc. 19th IEEE Symp. on Logic in Computer Science (LICS)*. 12–21.

ETESSAMI, K., STEWART, A., AND YANNAKAKIS, M. 2012a. Polynomial-time algorithms for multi-type branching processes and stochastic context-free grammars. In *Proc. 44th ACM STOC*. Full version is available at ArXiv:1201.2374.

ETESSAMI, K., STEWART, A., AND YANNAKAKIS, M. 2012b. Polynomial time algorithms for branching Markov decision processes and probabilistic min(max) polynomial Bellman equations. In *Proc. of 39th Int. Coll. on Automata, Languages and Programming (ICALP)*. 314–326. Full version available at ArXiv:1202.4798.

ETESSAMI, K., WOJTCZAK, D., AND YANNAKAKIS, M. 2008. Recursive stochastic games with positive rewards. In *Proc. 35th ICALP*. 71–723.

ETESSAMI, K., WOJTCZAK, D., AND YANNAKAKIS, M. 2010. Quasi-birth-death processes, tree-like QBDs, probabilistic 1-counter automata, and pushdown systems. *Perform. Eval. 67,* 9, 837–857. (conference version appeared in QEST'08).

ETESSAMI, K. AND YANNAKAKIS, M. 2005. Recursive Markov decision processes and recursive stochastic games. In *Proc. 32nd Int. Coll. on Automata, Languages, and Programming (ICALP)*. 891–903.

ETESSAMI, K. AND YANNAKAKIS, M. 2006. Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In *Proc. of 23rd STACS'06*. Springer.

ETESSAMI, K. AND YANNAKAKIS, M. 2008. Recursive concurrent stochastic games. *Logical Methods in Computer Science 4,* 4. (conference version appeared in ICALP'06).

ETESSAMI, K. AND YANNAKAKIS, M. 2009. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *J. ACM 56,* 1.

ETESSAMI, K. AND YANNAKAKIS, M. 2010. On the complexity of Nash equilibria and other fixed points. *SIAM J. Comput. 39,* 6, 2531–2597. (Conference version appeared in Proc. 48th IEEE Symp. on Foundations of Computer Science (FOCS'07)).

ETESSAMI, K. AND YANNAKAKIS, M. 2012. Model checking of recursive probabilistic systems. *ACM Transactions on Computational Logic 13*.

EVERETT, C. J. AND ULAM, S. 1948. Multiplicative systems, part i., ii, and iii. Tech. Rep. 683,690,707, Los Alamos Scientific Laboratory.

EVERETT, H. 1957. Recursive games. In *Contributions to the theory of games, vol. 3*. Annals of Mathematics Studies, no. 39. Princeton University Press, 47–78.

FAGIN, R., KARLIN, A., KLEINBERG, J., RAGHAVAN, P., RAJAGOPALAN, S., RUBINFELD, R., SUDAN, M., AND TOMKINS, A. 2000. Random walks with "back buttons" (extended abstract). In *Proc. ACM Symp. on Theory of Computing (STOC)*. 484–493.

FEINBERG, E. AND SHWARTZ, A., Eds. 2002. *Handbook of Markov Decision Processes*. Kluwer.

FILAR, J. AND VRIEZE, K. 1997. *Competitive Markov Decision Processes*. Springer.

GAREY, M. R., GRAHAM, R. L., AND JOHNSON, D. S. 1976. Some NP-complete geometric problems. In *Proc. 8th ACM Symp. on Theory of Computing (STOC)*. 10–22.

GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. 1993. *Geometric Algorithms and Combinatorial Optimization* 2nd Ed. Springer-Verlag.

HACCOU, P., JAGERS, P., AND VATUTIN, V. A. 2005. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press.

HARRIS, T. E. 1963. *The Theory of Branching Processes*. Springer-Verlag.

HART, S., SHARIR, M., AND PNUELI, A. 1983. Termination of probabilistic concurrent programs. *ACM Trans. on Programming Languages and Systems 5,* 3, 356–380.

HOPCROFT, J. E. AND ULLMAN, J. D. 1979. *Introduction to Automata Theory, Formal Languages and Computation*. Addison-Wesley.

HORN, R. J. AND JOHNSON, C. R. 1985. *Matrix Analysis*. Cambridge U. Press.

JAGERS, P. 1975. *Branching Processes with Biological Applications*. Wiley.

KIMMEL, M. AND AXELROD, D. E. 2002. *Branching processes in biology*. Springer.

KOLMOGOROV, A. N. AND SEVASTYANOV, B. A. 1947. The calculation of final probabilities for branching random processes. *Dokaldy 56*, 783–786. (Russian).

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

LATOUCHE, G. AND RAMASWAMI, V. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM series on statistics and applied probability.

MADANI, O., HANKS, S., AND CONDON, A. 2003. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence 147,* 1, 5–34.

MAHLER, K. 1964. An inequality for the discriminant of a polynomial. *Michigan Math J. 11*, 257–262.

MAITRA, A. AND SUDDERTH, W. 1998. Finitely additive stochastic games with Borel measurable payoffs. *Internat. J. Game Theory 27,* 2, 257–267.

MANNING, C. AND SCHÜTZE, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

MARTIN, D. A. 1998. Determinacy of Blackwell games. *J. Symb. Logic 63,* 4, 1565–1581.

NEUTS, M. F. 1981. *Matrix-Geometric Solutions in Stochastic Models:an algorithmic approach*. Johns Hopkins U. Press.

NEYMAN, A. AND SORIN, S., Eds. 2003. *Stochastic Games and Applications*. NATO ASI Series, Kluwer.

ORNSTEIN, D. 1969. On the existence of stationary optimal strategies. *Proc. Amer. Math. Soc. 20*, 563–569.

PAZ, A. 1971. *Introduction to Probabilistic Automata*. Academic Press.

PLISKA, S. 1976. Optimization of multitype branching processes. *Management Sci. 23,* 2, 117–124.

PUTERMAN, M. L. 1994. *Markov Decision Processes*. Wiley.

RENEGAR, J. 1992. On the computational complexity and geometry of the first-order theory of the reals, parts I-III. *J. Symbolic Computation 13,* 3, 255–352.

ROTHBLUM, U. AND WHITTLE, P. 1982. Growth optimality for branching Markov decision chains. *Math. Oper. Res. 7,* 4, 582–601.

SAKAKIBARA, Y., BROWN, M., HUGHEY, R., MIAN, I., SJOLANDER, K., UNDERWOOD, R., AND HAUSSLER, D. 1994. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research 22,* 23, 5112–5120.

SHAPLEY, L. 1953. Stochastic games. *Proc. Nat. Acad. Sci. 39*, 1095–1100.

TARSKI, A. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics 5,* 2, 285–309.

VARDI, M. 1985. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of 26th IEEE FOCS*. 327–338.

WALUKIEWICZ, I. 1996. Pushdown processes: games and model checking. In *Computer-Aided Verification*. 62–75.

WOJTCZAK, D. AND ETESSAMI, K. 2007. PReMo: an analyzer for probabilistic recursive models. In *Proc. 13th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. 66–71. Tool web page: `http://groups.inf.ed.ac.uk/premo/`.

ZWICK, U. AND PATERSON, M. 1996. The complexity of mean payoff games on graphs. *Theoretical Computer Science 158,* 1-2, 343–359.

Received ; revised ; accepted