Recursive Stochastic Games with Positive Rewards

K. Etessami[1], D. Wojtczak[1], and M. Yannakakis[2]

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

Abstract. We study the complexity of a class of Markov decision processes and, more generally, stochastic games, called 1-exit Recursive Markov Decision Processes (1-RMDPs) and Simple Stochastic Games (1-RSSGs) with strictly positive rewards. These are a class of finitely presented countable-state zero-sum stochastic games, with total expected reward objective. They subsume standard finite-state MDPs and Condon's simple stochastic games and correspond to optimization and game versions of several classic stochastic models, with rewards. Such stochastic models arise naturally as models of probabilistic procedural programs with recursion, and the problems we address are motivated by the goal of analyzing the optimal/pessimal expected running time in such a setting.
We give polynomial time algorithms for 1-exit Recursive Markov decision processes (1-RMDPs) with positive rewards. Specifically, we show that the exact optimal value of both maximizing and minimizing 1-RMDPs with positive rewards can be computed in polynomial time (this value may be ∞). For two-player 1-RSSGs with positive rewards, we prove a "stackless and memoryless" determinacy result, and show that deciding whether the game value is at least a given value r is in NP $\cap$ coNP. We also prove that a simultaneous strategy improvement algorithm converges to the value and optimal strategies for these stochastic games. Whether this algorithm runs in P-time is open, just like its classic version for finite SSGs. We observe that 1-RSSG positive reward games are "harder" than finite-state SSGs in several senses.

1 Introduction

Markov decision processes and stochastic games are fundamental models in stochastic optimization and game theory (see, e.g., [28, 26, 16]). In this paper, motivated by the goal of analyzing the optimal/pessimal expected running time of probabilistic procedural programs, we study the complexity of a reward-based stochastic game, called *1-exit recursive simple stochastic games* (1-RSSGs), and its 1-player version, *1-exit recursive Markov decision processes* (1-RMDPs). These form a class of (finitely presented) countable-state turn-based zero-sum stochastic games (and MDPs) with strictly positive rewards, and with an undiscounted expected total reward objective.

Intuitively, a 1-RSSG (1-RMDP) consists of a collection of finite-state component SSGs (MDPs), each of which can be viewed as an abstract finite-state procedure (subroutine) of a probabilistic program with potential recursion. Each component procedure has some nodes that are probabilistic and others that are controlled by one or the other of the two players. The component SSGs can call each other in a recursive manner, generating a potentially unbounded call stack, and thereby an infinite state space. The "1-exit" restriction essentially restricts these finite-state subroutines so they do not return

a value, unlike multi-exit RSSGs and RMDPs in which they can return distinct values. (We shall show that the multi-exit version of these reward games are undecidable.) An example 1-RSSG is depicted in Figure 1 of the appendix. 1-RMDPs and 1-RSSGs were studied in [12, 13] in a setting without rewards, where the goal of the players was to maximize/minimize the probability of termination. Such termination probabilities can be irrational, and quantitative decision problems for them subsume long standing open problems in exact numerical computation. Here we extend 1-RSSGs and 1-RMDPs to a setting with positive rewards. Note that much of the literature on MDPs and games is based on a reward structure. This paper is a first step toward extending these models to the recursive setting. Interestingly, we show that the associated problems actually become more benign in some respects in this strictly positive reward setting. In particular, the values of our games are either rational, with polynomial bit complexity, or ∞.

The 1-RMDP and 1-RSSG models can also be described as optimization and game versions of several standard stochastic models, including stochastic context-free grammars (SCFGs) and (multi-type) branching processes. These are classic stochastic models, with applications in many areas, including natural language processing [24], biological sequence analysis ([30, 7, 22]), and population biology [18, 17]. Another model that corresponds in a precise sense to a strict subclass of SCFGs is "random walks with back-buttons" studied in [15] as a model of web surfing. See [11] for details on the relationships between these various models.

A 1-RSSG with positive rewards, can be equivalently reformulated as the following game played on a stochastic context-free grammar (appendix A.2 details why they are equivalent). We are given a context-free grammar where the non-terminals are partitioned into three disjoint sets: `random`, `player-1`, and `player-2`. Starting from a designated start non-terminal, S_{init}, we proceed to generate a derivation by choosing a remaining *left-most* non-terminal, S, and expanding it. As we soon discuss, the precise derivation law (left-most, right-most, etc.) doesn't effect the game value in our strictly positive reward setting, but does if we allow 0 rewards. If S belongs to `random`, it is expanded randomly by choosing a rule $S \to \alpha$, according to a given probability distribution over the rules whose left hand side is S. If S belongs to `player-i`, then player i chooses which grammar rule to use to expand this S. Each grammar rule also has an associated (strictly positive) *reward* for player 1, and each time a rule is used during the derivation, player 1 accumulates this associated reward. Player 1 wants to maximize its total expected reward (which may be ∞). This being a zero-sum game, player 2 wants to minimize this total expected reward. The case where we have only one of the two players is a minimizing or maximizing 1-RMDP.

We assume strictly positive rewards on all transitions (rules) in this paper. This assumption is very natural for modeling optimal/pessimal expected running time in probabilistic procedural programs: each discrete step of the program is assumed to cost some non-zero amount of time. Strictly positive rewards also endow our games with a number of important robustness properties. In particular, in the above grammar presentation, with strictly positive rewards these games have the same value regardless of what derivation law is imposed. This is not the case if we also allow 0 rewards on grammar rules. In that case, even in the single-player setting, the game value can be wildly different (e.g., can be 0 or ∞) depending on the derivation law (e.g., left-most or right-most). Moreover, for 1-RMDPs, if we allow 0 rewards, then there may not

even exist any ϵ-optimal strategies. Furthermore, even in a purely probabilistic setting without players (1-RMCs), with 0 rewards the expected reward can be irrational. Even the decidability of determining whether the supremum expected reward for 1-RMDPs is greater than a given rational value is open, and subsumes other open decidability questions, e.g., for optimal reachability probabilities in non-reward 1-RMDPs ([12, 2]). (See appendix A.3 for simple examples that illustrate these issues.) As we shall show, none of these pathologies arise in our setting with strictly positive rewards.

We show that 1-RMDPs and 1-RSSGs with strictly positive rewards have a value which is either rational (with polynomial bit complexity) or ∞, and which arises as the least fixed point solution (over the extended reals) of an associated system of linear-min-max equations. Both players do have optimal strategies in these games, and in fact we show the much stronger fact that both players have *stackless and memoryless* (SM) optimal strategies: deterministic strategies that depend only on the current state of the running component, and not on the history or even the stack of pending recursive calls.

We provide polynomial-time algorithms for computing the exact value for both the maximizing and minimizing 1-RMDPs. The two cases are not equivalent and require separate treatment. We show that for the 2-player games (1-RSSGs) deciding whether the game has value at least a given $r \in \mathbb{Q} \cup \{\infty\}$ is in NP $\cap$ coNP. We also describe a practical simultaneous strategy improvement algorithm, analogous to similar algorithms for finite-state stochastic games, and show that it converges to the game value (even if it is ∞) in a finite number of steps. A corollary is that computing the game value and optimal strategies for these games is contained in the class PLS of polynomial local search problems ([20]). Whether this strategy improvement algorithm runs in worst-case P-time is open, just like its version for finite-state SSGs.

We observe that these games are essentially "harder" than Condon's finite-state SSG games in the following senses. We reduce Condon's quantitative decision problem for finite-state SSGs to a special case of 1-RSSG games with strictly positive rewards: namely to deciding whether the game value is ∞. By contrast, if finite-state SSGs are themselves equipped with strictly positive rewards, we can decide in P-time whether their value is ∞. Moreover, it has recently been shown that computing the value of Condon's SSG games is in the complexity class PPAD (see [14] and [21]). The same proof however does not work for 1-RSSG positive reward games, and we do not know whether these games are contained in PPAD. Technically, the problem is that in the expected reward setting the domain of the fixed point equations is not compact, and indeed the expected reward is potentially ∞, so the problem can not in any obvious way be formulated as a Brouwer fixed point problem. In these senses, the 1-RSSG reward games studied in this paper appear to be "harder" than Condon's SSGs, and yet as we show their quantitative decision problems remain in NP $\cap$ coNP.

Finally, we show that the more general multi-exit RSSG model is undecidable. Namely, even for single-player multi-exit RMDPs with strictly positive rewards, it is undecidable whether the optimal reward value is ∞.

The tool PReMo [32] implements a number of analyses for RMCs, 1-RMDPs, and 1-RSSGs. Most recently, the strategy improvement algorithm of this paper was implemented and incorporated in the tool. See the PReMo web page ([32]) for very encouraging experimental results based on the algorithms of this paper.

Related work.

Two (equivalent) purely probabilistic recursive models, Recursive Markov chains and probabilistic Pushdown Systems (pPDSs) were introduced in [11] and [8], and have been studied in several papers recently. These models were extended to the optimization and game setting of (1)-RMDPs and (1)-RSSGs in [12, 13], and studied further in [2]. As mentioned earlier, the games considered in these earlier papers had the goal of maximizing/minimizing termination or reachability probability, which can be irrational, and for which quantitative decision problems encounter long standing open problems in numerical computation, even to place their complexity in NP. On the other hand, the qualitative decision problem ("is the termination game value exactly 1?") for 1-RMDPs with a termination criterion was shown to be in P, and for 1-RSSGs in NP $\cap$ coNP in [13] using an eigenvalue characterization and linear programming. These results are related to the results in the present paper as follows. If termination occurs with probability strictly less than 1 in a strictly positive reward game, then the expected total reward is ∞. But the converse does not hold: the expected reward may be ∞ even when the game terminates with probability 1, because there can be *null recurrence* in these infinite-state games. Thus, not only do we have to address this discrepancy, but also our goal in this paper is quantitative computation (compute the optimal reward), whereas in [13] it was purely qualitative (almost sure termination). Our proofs here avoid the eigenvalue techniques used in [13], relying instead on basic properties of non-negative matrices over the extended reals, and LP theory. Our proof of SM determinacy, and the *simultaneous* strategy improvement algorithm, modifies and strengthens an intricate argument from [12] which relied on analytic properties of certain power series. In our setting here these functions become linear over the *extended* reals, and retain similarly useful properties.

Condon [4] originally studied finite-state SSGs with termination objectives (no rewards), and showed that the quantitative termination decision problem is in NP $\cap$ coNP; it is a well-known open problem whether it is in P. In [5] strategy improvement algorithms for SSGs were studied, based on variants of the classic Hoffman-Karp algorithm [19]. It remains open whether the simultaneous version of strategy improvement runs in P-time. This is also the case for our simultaneous strategy improvement algorithm for 1-RSSGs with positive rewards. (Single-vertex updates per step in strategy improvement is known to require exponentially many steps in the worst case.)

There has been some recent work on augmenting purely probabilistic multi-exit RMCs and pPDSs with rewards in [9, 3]. These results however are for RMCs without players. We in fact show in Theorem 8 that the basic questions about multi-exit RMDPs and RSSGs are undecidable.

Models related to 1-RMDPs have been studied in Operations Research and stochastic control, under the name Branching Markov Decision Chains (a controlled version of multi-type Branching processes). These models are close to the single-player SCFG model, with non-negative rewards, but with a simultaneous derivation law. They were studied by Pliska [27], in a related form by Veinott [31], and extensively by Rothblum and co-authors (e.g., [29, 6]). Besides the restriction to simultaneous derivation, these models were restricted to the single-player MDP case, and moreover to simplify their analysis they were typically assumed to be "transient" (i.e., the expected number of visits to a node was assumed to be finite under all strategies). None of these earlier results from the OR literature yield a P-time algorithm for computing the optimal expected reward, given a 1-RMDP with positive rewards.

2 Definitions and Background

Let $\mathbb{R}_{>0} = (0, \infty)$ denote the positive real numbers, $\mathbb{R}_{\geq 0} \doteq [0, \infty)$, $\overline{\mathbb{R}} \doteq [-\infty, \infty]$, $\mathbb{R}^{\infty}_{>0} \doteq (0, \infty]$, and $\mathbb{R}^{\infty}_{\geq 0} \doteq [0, \infty]$. The extended reals $\overline{\mathbb{R}}$ have the natural total order. We assume the following usual arithmetic conventions on the non-negative extended reals $\mathbb{R}^{\infty}_{\geq 0}$: $a \cdot \infty = \infty$, for any $a \in \mathbb{R}^{\infty}_{>0}$; $0 \cdot \infty = 0$; $a + \infty = \infty$, for any $a \in \mathbb{R}^{\infty}_{\geq 0}$. This extends naturally to matrix arithmetic over $\mathbb{R}^{\infty}_{\geq 0}$.

We first define general multi-exit RSSGs (for which basic reward problems turn out to be undecidable). Later, we will confine these to the 1-exit case, 1-RSSGs. In the appendix we explain why 1-RSSGs are essentially equivalent to SCFG games with left-most derivation.

A *Recursive Simple Stochastic Game (RSSG) with positive rewards* is a tuple $A = (A_1, \ldots, A_k)$, where each *component* $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \mathtt{pl}_i, \delta_i, \xi_i)$ 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*, $Call_b = \{(b, en) \mid en \in En_{Y(b)}\}$, and a set of *return ports*, $Return_b = \{(b, ex) \mid ex \in Ex_{Y(b)}\}$. Let $Call^i = \cup_{b \in B_i} Call_b$, $Return^i = \cup_{b \in B_i} Return_b$, 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"). We assume $\mathtt{pl}_i(ex) = 0$ for all $ex \in Ex_i$.
- A transition relation $\delta_i \subseteq (Q_i \times (\mathbb{R}_{>0} \cup \{\bot\}) \times Q_i \times \mathbb{R}_{>0})$, where for each tuple $(u, x, v, c_{u,v}) \in \delta_i$, the source $u \in (N_i \setminus Ex_i) \cup Return^i$, the destination $v \in (N_i \setminus En_i) \cup Call^i$, and x is either (i) $p_{u,v} \in (0, 1]$ (the transition probability) if $\mathtt{pl}_i(u) = 0$, or (ii) $x = \bot$ if $\mathtt{pl}_i(u) = 1$ or 2; and $c_{u,v} \in \mathbb{R}_{>0}$ is the positive reward associated with this transition. We assume that for every eligible pair of vertices u and v there is at most one transition in δ from u to v. For computational purposes we assume the given probabilities $p_{u,v}$ and rewards $c_{u,v}$ are rational. Probabilities must also satisfy consistency: for every $u \in \mathtt{pl}_i^{-1}(0)$, $\sum_{\{v' \mid (u, p_{u,v'}, v', c_{u,v}) \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or exit node, neither of which have outgoing transitions, in which case by default $\sum_{v'} p_{u,v'} = 0$.
- Finally, the mapping $\xi_i : Call_i \mapsto \mathbb{R}_{>0}$ maps each call port u in the component to a positive rational value $c_u = \xi(u)$.[1]

We use the symbols (N, B, Q, δ, etc.) without a subscript, to denote the union over all components. Thus, e.g., $N = \cup_{i=1}^{k} N_i$ is the set of all nodes of A, $\delta = \cup_{i=1}^{k} \delta_i$ the set of all transitions, etc. Let $next(u) = \{v \mid (u, \bot, v, c_{u,v}) \in \delta\}$, if u is a *min* or *max* node and $next(u) = \{v \mid (u, p_{u,v}, v, c_{u,v}) \in \delta\}$ otherwise. An RSSG A defines a global denumerable simple stochastic game, with rewards, $M_A = (V = V_0 \cup V_1 \cup V_2, \Delta, \mathtt{pl})$ 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 states $V \subseteq B^* \times Q$ and transitions Δ are defined inductively as follows:

[1] This mapping is not strictly necessary, and it is restricted to positive values only for convenience in proofs: c_u's can also be any non-negative values and all our results would hold because of the structure of 1-RSSGs.

1. $\langle \epsilon, u \rangle \in V$, for $u \in Q$. (ϵ denotes the empty string.)
2. if $\langle \beta, u \rangle \in V$ & $(u, x, v, c) \in \delta$, then $\langle \beta, v \rangle \in V$ and $(\langle \beta, u \rangle, x, \langle \beta, v \rangle, c) \in \Delta$.
3. if $\langle \beta, (b, en) \rangle \in V$ & $(b, en) \in \mathit{Call}_b$, then $\langle \beta b, en \rangle \in V$ & $(\langle \beta, (b, en) \rangle, 1, \langle \beta b, en \rangle, \xi((b, en))) \in \Delta$.
4. if $\langle \beta b, ex \rangle \in V$ & $(b, ex) \in \mathit{Return}_b$, then $\langle \beta, (b, ex) \rangle \in V$ & $(\langle \beta b, ex \rangle, 1, \langle \beta, (b, ex) \rangle, 0) \in \Delta$.

The mapping $\mathtt{pl} : V \mapsto \{0, 1, 2\}$ is given as follows: $\mathtt{pl}(\langle \beta, u \rangle) = \mathtt{pl}(u)$ if u is in $Q \setminus (\mathit{Call} \cup \mathit{Ex})$, and $\mathtt{pl}(\langle \beta, u \rangle) = 0$ if $u \in \mathit{Call} \cup \mathit{Ex}$. 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. An RSSG where $V_2 = \emptyset$ ($V_1 = \emptyset$) is called a maximizing (minimizing, respectively) *Recursive Markov Decision Process* (RMDP); an RSSG where $V_1 \cup V_2 = \emptyset$ is called a *Recursive Markov Chain* (RMC) ([11, 10]); A *1-RSSGs* is a RSSG where every component has one exit, and we likewise define *1-RMDPs* and *1-RMCs*. This entire paper is focused on 1-RSSGs and 1-RMDPs, except for Theorem 8, where we show that multi-exit RMDP reward games are undecidable. In a (1-)RSSG with positive rewards the goal of player 1 (maximizer) is to maximize the total expected reward gained during a play of the game, and the goal of player 2 (minimizer) is to minimize the total expected reward. A *strategy* σ for player i, $i \in \{1, 2\}$, is a function $\sigma : V^* V_i \mapsto V$, where, given the history $ws \in V^* V_i$ of play so far, with $s \in V_i$ (i.e., it is player i's turn to play a move), $\sigma(ws) = s'$ determines the next move of player i, where $(s, \bot, s', c) \in \Delta$. (We could also allow randomized strategies, but this won't be necessary, as we shall see.) Let Ψ_i denote the set of all strategies for player i. A pair of strategies $\sigma \in \Psi_1$ and $\tau \in \Psi_2$ induce in a straightforward way a Markov chain $M_A^{\sigma,\tau} = (V^*, \Delta')$, whose set of states is the set V^* of histories. Let $r_u^{k,\sigma,\tau}$ denote the expected reward in k steps in $M_A^{\sigma,\tau}$, starting at initial state $\langle \epsilon, u \rangle$. Formally, we can define the total expected reward gained during the i'th transition, starting at $\langle \epsilon, u \rangle$ to be given by a random variable Y_i. The total k-step expected reward is simply $r_u^{k,\sigma,\tau} = E[\sum_{i=1}^{k} Y_i]$. When $k = 0$, we of course have $r_u^{0,\sigma,\tau} = 0$. Given an initial vertex u, let $r_u^{*,\sigma,\tau} = \lim_{k \to \infty} r^{k,\sigma,\tau} = E[\sum_{i=1}^{\infty} Y_i] \in [0, \infty]$ denote the total expected reward obtained in a run of $M_A^{\sigma,\tau}$, starting at initial state $\langle \epsilon, u \rangle$. Clearly, this sum may diverge, thus the need to consider $r^{*,\sigma,\tau} \in [0, \infty]$. Note that, because of the positive constraint on the rewards out of all transitions, the sum will be finite if and only if the expected number of steps until the run terminates is finite.

We now want to associate a "value" to 1-RSSG games. Unlike 1-RSSGs with termination probability objectives, it unfortunately does not follow directly from general determinacy results such as Martin's Blackwell determinacy (see [25, 23]) that these games are determined, because those determinacy results require a Borel payoff function to be bounded, whereas the payoff function for us is unbounded. Nevertheless, we will establish that determinacy does hold for 1-RSSG positive reward games, as part of our proof of a stronger Stackless & Memoryless determinacy result. For all vertices u, let $r_u^* \doteq \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} r_u^{*,\sigma,\tau}$. We will in fact show that $r_u^* = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} r_u^{*,\sigma,\tau}$, and thus that r_u^* denotes the *value* of the game starting at vertex u.

We are interested in the following problem: *Given A, a 1-RSSG (or 1-RMDP), and given a vertex u in A, compute r_u^* if it is finite, or else declare that $r_u^* = \infty$. Also, compute optimal strategies for both players.*

For a strategy $\sigma \in \Psi_1$, let $r_u^{*,\sigma} = \inf_{\tau \in \Psi_2} r_u^{*,\sigma,\tau}$, and for $\tau \in \Psi_2$, let $r_u^{*,\cdot,\tau} = \sup_{\sigma \in \Psi_1} r_u^{*,\sigma,\tau}$. Call a deterministic strategy *Stackless & Memoryless (SM)* if it depends

neither on the history of the game nor on the current call stack, i.e., only depends on the current vertex. Such strategies, for player i, can be given by a map $\sigma : V_i \mapsto V$. We call a game *SM-determined* if both players have optimal SM strategies.

In ([12]) we defined a monotone system S_A of nonlinear min-max equations for the value of the termination probability game on 1-RSSGs, and showed that its *Least Fixed Point* solution yields the desired probabilities. Here we show we can adapt this to obtain analogous linear min-max systems in the setting of positive reward 1-RSSGs. We use a variable x_u for each unknown r^*_u. 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 5 cases based on the "*Type*" of u.

1. $u \in Type_0$: $u = ex$. In this case: $x_u = 0$.
2. $u \in Type_{rand}$: $\mathtt{pl}(u) = 0$ & $u \in (N_i \setminus \{ex\}) \cup Return^i$: $x_u = \sum_{v \in next(u)} p_{u,v}(x_v + c_{u,v})$.
3. $u \in Type_{call}$: $u = (b, en)$ is a call port: $x_{(b,en)} = x_{en} + x_{(b,ex')} + c_u$, where $ex' \in Ex_{Y(b)}$ is the unique exit of $A_{Y(b)}$.
4. $u \in Type_{max}$: $\mathtt{pl}(u) = 1$ and $u \in (N_i \setminus \{ex\}) \cup Return^i$: $x_u = \max_{v \in next(u)}(x_v + c_{u,v})$
5. $u \in Type_{min}$: $\mathtt{pl}(u) = 2$ and $u \in (N_i \setminus \{ex\}) \cup Return^i$: $x_u = \min_{v \in next(u)}(x_v + c_{u,v})$

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. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we write $\mathbf{x} \leq \mathbf{y}$ to mean $x_j \leq y_j$ for every coordinate j. Let $\mathbf{r}^* \in \mathbb{R}^n$ denote the n-vector of r^*_u's. Let $\mathbf{0}$ denote an all zero n-vector, and define the sequence $\mathbf{x}^0 = \mathbf{0}$, $\mathbf{x}^{k+1} = P^{k+1}(\mathbf{0}) = P(\mathbf{x}^k)$ for $k \geq 0$.

Theorem 1. *(1) The map $P : \overline{\mathbb{R}}^n \to \overline{\mathbb{R}}^n$ is monotone on $\mathbb{R}^\infty_{\geq 0}$ 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 $k \geq 0$, $\mathbf{x}^k \leq \mathbf{r}^*$. (4) For all $\mathbf{r}' \in \mathbb{R}^\infty_{\geq 0}$, if $\mathbf{r}' = P(\mathbf{r}')$, then $\mathbf{r}^* \leq \mathbf{r}'$. (5) For all vertices u, $r^*_u \doteq \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} r^{*,\sigma,\tau}_u = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} r^{*,\sigma,\tau}_u$ (i.e., these games are determined). (6) $\mathbf{r}^* = \lim_{k \to \infty} \mathbf{x}^k$.*
The proof is in the appendix. The following is a simple corollary of the proof.

Corollary 1. *In 1-RSSG positive reward games, the* minimizer *has an optimal deterministic Stackless and Memoryless (SM) strategy.*

Note that for a 1-RMC (i.e., without players) with positive rewards, the vector $\mathbf{r}^*$ of expected total rewards is the LFP of a system $x = Ax + b$, for some non-negative matrix $A \in \mathbb{R}^{n \times n}$, $A \geq 0$, and a positive vector $b > 0$. The following will be useful later.[2]

Lemma 1. *For any $x \in \mathbb{R}^n_{\geq 0}$, $A \in (\mathbb{R}^\infty_{\geq 0})^{n \times n}$ and $b \in (\mathbb{R}^\infty_{>0})^n$, if $x \leq Ax + b$ then $x \leq (\sum_{k=0}^\infty A^k)b$. This holds even if for some indices i we have $b_i = 0$, as long as the entries in any such row i of the matrix A are all zero.*

3 SM-determinacy and strategy improvement

We now prove SM-determinacy, and at the same time show that strategy improvement can be used to compute the values and optimal strategies for 1-RSSG positive reward games. Consider the following *(simultaneous) strategy improvement* algorithm.

[2] Note that if we assume both that $A \in (\mathbb{R}_{\geq 0})^{n \times n}$ and that $(\sum_{k=0}^\infty A^k)$ converges, the lemma is trivial: we have $(I - A)^{-1} = (\sum_{k=0}^\infty A^k)$, and thus $x \leq Ax + b \Rightarrow x - Ax \leq b \Rightarrow (I - A)x \leq b \Rightarrow x \leq (I - A)^{-1}b$. But we need this lemma even when $(\sum_{k=0}^\infty A^k)$ is not convergent.

Initialization: Pick some SM strategy, σ, for player 1 (maximizer).
Iteration step: First compute the optimal value, $r_u^{*,\sigma}$, starting from every vertex, u, in the resulting minimizing 1-RMDP. (We show in Theorem 3 that this can be done in P-time.) Then, update σ to a new SM strategy, σ', as follows. For each vertex $u \in Type_{max}$, if $\sigma(u) = v$ and u has a neighbor $w \neq v$, such that $r_w^{*,\sigma} + c_{u,w} > r_v^{*,\sigma} + c_{u,v}$, let $\sigma'(u) := w$ (e.g., choose a w that maximizes $r_w^{*,\sigma} + c_{u,w}$). Otherwise, let $\sigma'(u) := \sigma(u)$.
Repeat the iteration step, using the new σ' in place of σ, until no further local improvement is possible, i.e., stop when $\sigma' = \sigma$.

The next theorem shows that this algorithm always halts, and produces a final SM strategy, σ, which is optimal for player 1. Thus, combined with Corollary 1, both players have optimal SM strategies, or in other words, these games are SM-determined. Since each "local improvement" step can be carried out in P-time, this also shows that this is a local search problem contained in the complexity class PLS ([20]).

Theorem 2. *(1)* SM-determinacy. *In 1-RSSG positive reward games, both players have optimal SM strategies. (2)* Strategy Improvement. *Moreover, we can compute the value and optimal SM strategies using the above simultaneous*[3] *strategy improvement algorithm. (3) Consequently (combined with Theorem 3) the search problem for computing the value and optimal strategies in these games is contained in the class PLS.*

The proof is intricate, and is given in appendix A.6. Here we briefly sketch the approach. Fix a SM strategy σ for player 1. It can be shown that if $\mathbf{x} = P(\mathbf{x})$ is the linear-min-max equation system for this 1-RSSG, then $\mathbf{r}_u^{*,\sigma} \leq P_u(\mathbf{r}^{*,\sigma})$, for all vertices u, and equality fails only on vertices u_i belonging to player 1 such that $\sigma(u_i) = v_i$ is not "locally optimal", i.e., such that there exists some neighbor w_i such that $r_{w_i}^{*,\sigma} + c_{u_i,w_i} > r_{v_i}^{*,\sigma} + c_{u_i,v_i}$. Let $u_1, \ldots, u_n$ be all such vertices belonging to player 1. Associate a parameter $t_i \in \mathbb{R}^{\infty}_{\geq 0}$ with each such vertex u_i, creating a parametrized game $A(\mathbf{t})$, in which whenever the vertex u_i is encountered player 1 gains additional reward t_i and the game then terminates. Let $g_{u,\tau}(\mathbf{t})$ denote the expected reward of this parametrized game starting at vertex u, when player 1 uses SM strategy σ and player 2 uses SM strategy τ. Let $f_u(\mathbf{t}) = \min_\tau g_{u,\tau}(\mathbf{t})$. The vector $\mathbf{t}^\sigma$, where $t_i^\sigma = r_{u_i}^{*,\sigma}$, is a fixed point of $f_u(\mathbf{t})$, for every vertex u, and so is $\mathbf{t}^{\sigma'}$ where σ' is any SM strategy consistent with σ on all vertices other than the u_i's. The functions $g_{u,\tau}(\mathbf{t})$ can be shown to be continuous and nondecreasing over $[0,\infty]^n$, and expressible as an infinite sum of *linear* terms with non-negative coefficients. Using these properties of $g_{u,\tau}$, and their implications for f_u, we show that if σ' is the SM strategy obtained by locally improving the strategy σ at the u_i's, by letting $\sigma'(u_i) := w_i$, then $t_i^\sigma = \mathbf{r}_{u_i}^{*,\sigma} < \mathbf{r}_{u_i}^{*,\sigma'} = \mathbf{t}_i^{\sigma'}$, and thus also $\mathbf{r}_z^{*,\sigma} = f_z(\mathbf{t}^\sigma) \leq f_z(\mathbf{t}^{\sigma'}) = \mathbf{r}_z^{*,\sigma'}$, for any vertex z. Thus, switching to σ' does not decrease the value at any vertex, and increases it on all the switched vertices u_i. There are only finitely many SM strategies, thus after finitely many iterations we reach a SM strategy, σ, where no improvement is possible. This σ must be optimal. □

[3] *Simultaneous* refers to the fact that in each iteration we switch the strategy at all vertices u which can be improved, not just one. Our proof actually shows the algorithm works if we switch the strategy at any non-empty subset of such improvable vertices. But the simultaneous version has the advantage that it may run in P-time, whereas the single-vertex update version is known to require exponentially many steps in the worst case, even for finite MDPs.

4 The complexity of reward 1-RMDPs and 1-RSSGs

Theorem 3. *There is a polynomial-time algorithm for computing the exact optimal value (including the possible value ∞) of a 1-RMDP with positive rewards, in both the case where the single player aims to maximize, or to minimize, the total reward.*

We consider maximizing and minimizing 1-RMDPs separately.

Maximizing reward 1-RMDPs.

We are given a maximizing reward 1-RMDP (i.e., no $Type_{\min}$ nodes in the 1-RSSG). Let us call the following LP "*max-LP* ":

Minimize $\sum_{u \in Q} x_u$

Subject to:

$x_u = 0$	for all $u \in Type_0$
$x_u \geq \sum_{v \in next(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand}$
$x_u \geq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call}$; ex' is the exit of $Y(b)$.
$x_u \geq (x_v + c_{u,v})$	for all $u \in Type_{\max}$ and all $v \in next(u)$
$x_u \geq 0$	for all vertices $u \in Q$

We will show that, when the value vector $\mathbf{r}^*$ is finite, it is precisely the optimal solution to the above max-LP, and furthermore that we can use this LP to find and eliminate vertices u for which $r^*_u = \infty$. Note that if $\mathbf{r}^*$ is finite then it fulfills all the constraints of the max-LP, and thus it is a feasible solution. We will show that it must then also be an optimal feasible solution. We first have to detect the vertices u such that $\mathbf{r}^*_u = \infty$. For the max-linear equation system P, we define the underlying directed dependency graph G, where the nodes are the set of vertices, Q, and there is an edge in G from u to v if and only if the variable x_v occurs on the right hand side in the equation defining variable x_u in P. We can decompose this graph in linear time into strongly connected components(SCCs) and get an SCC DAG $SCC(G)$, where the set of nodes are SCCs of G, and an edge goes from one SCC A to another B, if and only if there is an edge in G from some node in A to some node in B. Let us sort topologically the SCCs of G as $S_1, S_2, \ldots, S_l$, where the bottom SCCs are listed first, and there is no edge in $SCC(G)$ from S_i to S_j for any $1 \leq i < j \leq l$. We will call a subset $U \subseteq Q$ of vertices *proper* if all vertices reachable in G from the vertices in U are already in U.[4] Clearly, such a proper set U must be a union of SCCs, and the equations restricted to variables in U do not use any variables outside of U, so they constitute a proper equation system on their own. For any proper subset U of G, we will denote by max-LP$|_U$ a subset of equations of max-LP, restricted to the constraints corresponding to variables in U and with new objective $\sum_{u \in U} x_u$. Analogously we define $P|_U$, and let $\mathbf{x}|_U$ be the vector $\mathbf{x}$ with entries indexed by any $v \notin U$ removed. The following is proved in the appendix.

Proposition 1. *Let U be any proper subset of vertices. (I) The vector $\mathbf{r}^*|_U$ is the LFP of $P|_U$. (II) If $r^*_u = \infty$ for some vertex u in an SCC S of G, then $r^*_v = \infty$ for all $v \in S$. (III) If $\mathbf{r}'$ is an optimal bounded solution to* max-LP$|_U$*, then $\mathbf{r}'$ is a fixed point of $P|_U$. (IV) If max-LP$|_U$ has a bounded optimal feasible solution $\mathbf{r}'$, then $\mathbf{r}' = \mathbf{r}^*|_U$.*

Theorem 4. *We can compute $\mathbf{r}^*$ for the max-linear equation system P, including the values that are infinite, in time polynomial in the size of the 1-RMDP.*

[4] For convenience we interchangably use U to refer to both the set of vertices and the corresponding set of variables.

Proof. Build a dependency graph G of P and decompose it into SCC graph $SCC(G)$. We will find the LFP solution to P, bottom-up starting at the lowest SCCs. We solve max-LP$|_{S_1}$ using a P-time algorithm for LP. If the LP is feasible then the optimal (minimum) value is bounded, and we plug in the values of the (unique) optimal solution as constants in all the other constraints of max-LP. We know this optimal solution is equal to $\mathbf{r}^*|_{S_1}$, since S_1 is *proper*. We do the same, in order, for S_2, S_3, ..., S_l. If at any point after adding the new constraints corresponding to the variables in an SCC S_i, the LP is *infeasible*, we know from Proposition 1 (IV), that at least one of the values of $\mathbf{r}^*|_{S_i}$ is ∞. So by Proposition 1 (II), all of them are. We can then mark all variables in S_i as ∞, and also mark all variables in the SCCs that can reach S_i in $SCC(G)$ as ∞. Also, at each step we add to a set U the SCCs that have finite optimal values. At the end of this process we have a maximal *proper* such set U, i.e., every variable outside of U has value ∞. We label the variables not in U with ∞, obtaining the vector $\mathbf{r}^*$. □

Minimizing reward 1-RMDPs.

Given a minimizing reward 1-RMDP (i.e., no $Type_{\max}$ nodes) we want to compute $\mathbf{r}^*$. Call the following LP "*min-LP:* "

Maximize $\sum_{u \in Q} x_u$
Subject to:

$x_u = 0$	for all $u \in Type_0$
$x_u \leq \sum_{v \in next(u)} p_{u,v}(x_v + c_{u,v})$	for all $u \in Type_{rand}$
$x_u \leq x_{en} + x_{(b,ex')} + c_u$	for all $u = (b, en) \in Type_{call}$; ex' is the exit of $Y(b)$.
$x_u \leq (x_v + c_{u,v})$	for all $u \in Type_{\min}$ and all $v \in next(u)$
$x_u \geq 0$	for all vertices $u \in Q$

Recall that a set of variables is *proper* if it is downward closed in the dependency graph of variables, which is defined in the same way as for maximizing 1-RMDPs.

Lemma 2. *For any* proper *set U, if an optimal solution $\mathbf{x}$ to the* min-LP$|_U$ *is bounded, then it is a fixed point to the min-linear operator $P|_U$. Thus, if min-LP$|_U$ has a bounded optimal feasible solution then $\mathbf{r}^*|_U$ is bounded (i.e., is a real vector).*

From min-LP we can remove variables $x_u \in Type_0$, by substituting their occurrences with 0. *Assume, for now, that we can also find and remove all variables x_u such that $r^*_u = \infty$.*. By removing these 0 and ∞ variables from P we obtain a new system P', and a new LP, min-LP$'$.

Lemma 3. *If ∞ and 0 nodes have been removed, i.e., if $\mathbf{r}^* \in (0, \infty)^n$, then $\mathbf{r}^*$ is the unique optimal feasible solution of min-LP$'$.*

Proof. By Corollary 1, player 2 has an optimal SM strategy, call it τ, which yields the finite optimal reward vector r^*. Once strategy τ is fixed, we can define a new equation system $P'_\tau(\mathbf{x}) = A_\tau \mathbf{x} + b_\tau$, where A_τ is a nonnegative matrix and b_τ is a vector of average rewards per single step from each node, obtained under strategy τ. We then have $\mathbf{r}^* = \lim_{k \to \infty} (P'_\tau)^k(0)$, i.e., $\mathbf{r}^*$ is the LFP of $x = P'(x)$.

Proposition 2. *(I) $\mathbf{r}^* = (\sum_{k=0}^{\infty} A_\tau^k) b_\tau$. (II) If $\mathbf{r}^*$ is finite, then $\lim_{k \to \infty} A_\tau^k = 0$, and thus $(I - A_\tau)^{-1} = \sum_{i=0}^{\infty} (A_\tau)^i$ exists (i.e., is a finite real matrix).*

See the appendix for the proof. Now pick an optimal *SM* strategy τ for the *min* player that yields the finite $\mathbf{r}^*$. We know that $\mathbf{r}^* = (I - A_\tau)^{-1} b_\tau$. Note that $\mathbf{r}^*$ is a feasible solution of the min-LP$'$. We show that for any feasible solution $\mathbf{r}$ to min-LP$'$, $\mathbf{r} \leq \mathbf{r}^*$. From the LP we can see that $\mathbf{r} \leq A_\tau \mathbf{r} + b_\tau$ (because this is just a subset of the constraints) and in other words $(I - A_\tau)\mathbf{r} \leq b_\tau$. We know that $(I - A_\tau)^{-1}$ exists and it is non-negative (and finite), so we can multiply both sides by $(I - A_\tau)^{-1}$ to get $\mathbf{r} \leq (I - A_\tau)^{-1} b_\tau = \mathbf{r}^*$. Thus $\mathbf{r}^*$ is the optimal feasible solution of min-LP$'$. □

For a node $u \in Q$, consider the LP: **Maximize** x_u, **subject to:** the same constraints as min-LP, except, again, remove all variables $x_v \in Type_0$. Call this LP u-min-LP$'$. In the Appendix we prove the following:

Theorem 5. *In a minimizing reward 1-RMDP, for all vertices u, the value $\mathbf{r}^*_u$ is finite iff u-min-LP$'$ is feasible and bounded. Thus, combining this with Lemma 3, we can compute the exact value (even if it is ∞) of minimizing reward 1-RMDPs in P-time.*

Consequence for (1-)RSSGs.

Theorem 6. *Deciding whether the value r^*_u of a 1-RSSG positive reward game is $\geq a$ for a given $a \in [0, \infty]$, is in NP $\cap$ coNP.*

This follows immediately from the P-time upper bounds for 1-RMDPs, and SM-determinacy: we guess one player's SM strategy, and compute the value for the remaining 1-RMDP.

Theorem 7. *Condon's quantitative termination problem for finite SSGs reduces in P-time to the problem of deciding whether $r^*_u = \infty$.*

The proof in the appendix. By contrast, for finite-state SSGs with strictly positive rewards, we can decide in P-time whether the value is ∞, because this is the case iff the value of the corresponding termination game is not 1.[5] Deciding whether an SSG termination game has value 1 is in P-time (see, e.g., [13]).
Finally, we show undecidability of multi-exit RMDPs and RSSGs with positive rewards.

Theorem 8. *For multi-exit positive reward RMDPs it is undecidable to distinguish whether the optimal expected reward for a node is finite or ∞.*

References

1. V. Blondel and V. Canterini. Undecidable problems for probabilistic automata of fixed dimension. *Theory of Computing Systems*, 36:231–245, 2003.
2. T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive markov decision processes. In *Proc. 17th Int. CONCUR*, pages 358–374, 2006.
3. T. Brázdil, J. Esparza, and A. Kucera. Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In *FOCS*, pages 521–530, 2005.
4. A. Condon. The complexity of stochastic games. *Inf. & Comp.*, 96(2):203–224, 1992.
5. A. Condon and M. Melekopoglou. On the complexity of the policy iteration algorithm for stochastic games. *ORSA Journal on Computing*, 6(2), 1994.

[5] This is basically because null-recurrence is not possible in finite state spaces.

6. E. Denardo and U. Rothblum. Totally expanding multiplicative systems. *Linear Algebra Appl.*, 406:142–158, 2005.
7. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of Proteins and Nucleic Acids*. Cambridge U. Press, 1999.
8. J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *LICS*, pages 12–21, 2004.
9. J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: expectations and variances. In *Proc. of 20th IEEE LICS'05*, 2005.
10. K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic state machines. In *Proc. 11th TACAS*, vol. 3440 of LNCS, 2005.
11. K. Etessami and M. Yannakakis. Recursive markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proc. of 22nd STACS'05*. Springer, 2005. (See full version: `http://homepages.inf.ed.ac.uk/kousha/stacs05_journal_version.ps`).
12. K. Etessami and M. Yannakakis. Recursive markov decision processes and recursive stochastic games. In *Proc. of 32nd Int. Coll. on Automata, Languages, and Programming (ICALP'05)*, 2005.
13. K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive markov decision processes and simple stochastic games. In *Proc. of 23rd STACS'06*. Springer, 2006.
14. K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. In *Proc. of 48th IEEE FOCS*, 2007.
15. R. Fagin, A. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons" (extended abstract). In *ACM Symp. on Theory of Computing*, pages 484–493, 2000.
16. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
17. P. Haccou, P. Jagers, and V. A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press, 2005.
18. T. E. Harris. *The Theory of Branching Processes*. Springer-Verlag, 1963.
19. A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Sci.*, 12:359–370, 1966.
20. D. S. Johnson, C. Papadimitriou, and M. Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.
21. B. Juba. On the hardness of simple stochastic games. Master's thesis, CMU, 2006.
22. M. Kimmel and D. E. Axelrod. *Branching processes in biology*. Springer, 2002.
23. A. Maitra and W. Sudderth. Finitely additive stochastic games with Borel measurable payoffs. *Internat. J. Game Theory*, 27(2):257–267, 1998.
24. C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
25. D. A. Martin. Determinacy of Blackwell games. *J. Symb. Logic*, 63(4):1565–1581, 1998.
26. A. Neyman and S. Sorin, editors. *Stochastic Games and Applications*. NATO ASI Series, Kluwer, 2003.
27. S. Pliska. Optimization of multitype branching processes. *Management Sci.*, 23(2):117–124, 1976/77.
28. M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.
29. U. Rothblum and P. Whittle. Growth optimality for branching Markov decision chains. *Math. Oper. Res.*, 7(4):582–601, 1982.
30. Y. Sakakibara, M. Brown, R Hughey, I.S. Mian, K. Sjolander, R. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, 22(23):5112–5120, 1994.
31. A. F. Veinott. Discrete dynamic programming with sensitive discount optimality criteria. *Ann. Math. Statist.*, 40:1635–1660, 1969.
32. D. Wojtczak and K. Etessami. Premo: an analyzer for probabilistic recursive models. In *Proc. of TACAS*, 2007. Tool web page: `http://groups.inf.ed.ac.uk/premo/`.

A Appendix

A.1 Example 1-RSSG

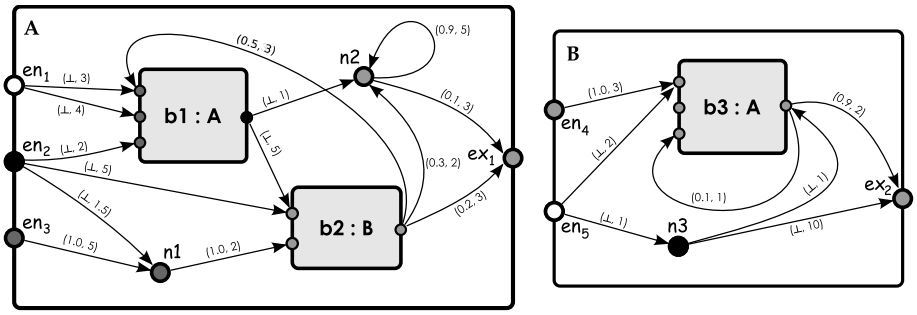

Fig. 1. A 1-RSSG example consisting of two components, A and B. Black vertices belong to player 1, white to player 2, grey vertices to "nature" (random). Each box (labelled, e.g., b1:A) has a name (b1) and is mapped to a component (A). Each edge has a label whose first component is $\perp$ or a probability, and the second component is the reward.

A.2 Definition of SCFG games with positive rewards and left-most derivation, and equivalence to 1-RSSGs with positive rewards

A *stochastic context-free grammar (SCFG) game*, is given by $G = (V, R, X_{start})$, where $V = V_0 \cup V_1 \cup V_2$ is a set of non-terminals, which is partitioned into three disjoint sets: V_0 are the probabilistic non-terminals (controlled by nature), V_1 and V_2, the non-terminals controlled by players 1 and 2, respectively. $X_{start} \in V$ is the start non-terminal. R is a set of rules, where each rule $r \in R$ has the form $r = (X, p_r, c_r, Z_r)$, where $X \in V$, and if $X \in V_0$ then $p_r \in [0, 1]$ is a (rational) probability, otherwise, if $X \in V_i$, $i > 0$, then $p_r = \perp$, $c_r \in \mathbb{Q}_{>0}$ is a rational reward, and $Z_r \in V^*$ is a (possibly empty) string of non-terminals. For each non-terminal, X, let $R_X \subseteq R$ denote the set of rules that have X on the left hand side. For each $X \in V_0$ we have $\sum_{r=(X,p_r,c_r,Z_r)\in R_X} p_r = 1$. The game proceeds as follows: the (countable) set of states of the game is a subset of V^*, i.e., strings of non-terminals. We begin the game in the state X_{start}. In each round, if the state is $\mathcal{S} = X_1 \dots X_k$, then we proceed by a left-most derivation law as follows[6]: choose a rule $r = (X_1, p_r, c_r, Z_r) \in R_{X_1}$. If $X_1 \in V_0$ the rule r is chosen probabilistically among the rules in R_{X_1}, according to the probability p_r. If $X_1 \in V_i$, $i \in \{1, 2\}$, then the rule r is chosen by player i. After the choice is made, the play moves to the new state $Z_r X_2 \dots X_k$. The reward gained in that round by player 1 is c_r. The game continues until

[6] As discussed in the introduction, we can also consider simultaneous derivation, which has different properties when 0 rewards are allowed. We focus our definitions on left-most derivation.

(and unless) we reach the empty-string state $\mathcal{S} = \epsilon$. The total reward gained by player 1 is the sum total of the rewards over every round. A strategy for player $d \in \{1, 2\}$ is a mapping that, given the history of play ending in state $XW \in V^*$, where $X \in V_d$, maps it to a rule $r \in R_X$. [7] Fixing strategies for the two players, we obtain a (denumerable) reward Markov chain whose states are (a subset of) V^*, the total reward is a random variable defined over the trajectories (runs) of this Markov chain. Player 1's goal is to maximize the expected total reward, and player 2's goal is to minimize it.

Let us now explain why 1-RSSGs with positive rewards are basically equivalent to SCFG games with positive rewards and with *left-most* derivation law, as discussed in the introduction. This follows by considering the equation systems $x = P(x)$ for 1-RSSGs (see Theorem 1), and the following Chomsky Normal Form (CNF) for SCFG games: there are only three kinds of rules in the grammar, either of the form (1) $X \mapsto \epsilon$, or (2) $X \mapsto Y$, or (3) $X \mapsto YZ$. Furthermore, all rules of the form $X \mapsto YZ$ are the unique rules associated with the non-terminal X, i.e., X is a probabilistic non-terminal, and the unique rule has the form $X \stackrel{(1,c)}{\mapsto} YZ$ for some positive reward c. It is not difficult to transform any reward SCFG game to one in the above CNF form, and with the same reward value starting from the start non-terminal, by adding some new non-terminals, as follows: a rule $X \stackrel{(p,c)}{\mapsto} X_1 X_2 \dots X_k$ can be replaced by the following rules $X \stackrel{(p,c/k)}{\mapsto} Z_k$; $\{Z_i \stackrel{(1,c/k)}{\mapsto} Z_{i-1} X_i \mid i = 2, \dots, k\}$, where we define $Z_1 \equiv X_1$, and for $i \in \{2, \dots, k\}$, Z_i is a new non-terminal. Now, the system of equations which yield the reward value for such a CNF form SCFG game can easily be seen to have exactly the same form as the equation systems $\mathbf{x} = P(\mathbf{x})$ for 1-RSSGs. It follows that the value vector r^* gives the expected total reward values both in the corresponding 1-RSSG game, starting at each vertex, and in the corresponding CNF form SCFG game, starting at each non-terminal.

A.3 Some examples formulated as SCFG games

In this section we describe some examples of 1-RSSG games using the simple (and expressively equivalent) formulation as a game over stochastic context-free grammars (SCFGs). Specifically, consider the SCFG with rewards given by the following grammar rules: $\{X \stackrel{(1/3,3)}{\mapsto} XX \ ; \ X \stackrel{(2/3,2)}{\mapsto} \epsilon\}$. Here X is the only non-terminal (and there are no terminal symbols). The pair (p, c) of quantities labelling a rule denotes the probability, p, of that rule firing, and the reward, c, accumulated for each use of that rule during a derivation. Consider now a random derivation of this grammar, starting from the non-terminal X, where the derivation proceeds in a left-most manner. In other words, in each round of the derivation we must expand the left-most non-terminal remaining in the derived sequence of non-terminals, by picking a rule according to the probability distribution on the rules whose left hand side is that non-terminal (in this case, the only non-terminal, X). The derivation terminates when it reaches the empty string ϵ. What is the expected total reward accumulated during the entire derivation? It is not hard to see that if we let x denote the total expected reward, then x must satisfy the following equation: $x = (1/3 * (3 + (x + x))) + (2/3 * 2) = (2/3)x + (7/3)$. Therefore,

[7] We could more generally define strategies that can yield probability distributions on the next rule, but this won't be necessary, since we shall see that indeed deterministic "stackless and memoryless" strategies are already optimal.

the total expected reward is the unique solution to this equation, namely $x = 7$. Note that, in general, such a derivation may not terminate with probability 1, and that the expected reward need not be finite (consider the same grammar with modified probabilities: $\{X \overset{(2/3,3)}{\mapsto} XX \ ; \ X \overset{(1/3,2)}{\mapsto} \epsilon\}$).

Now suppose, more generally, that we have a context-free grammar, with no terminal symbols, in which there are three different kinds of non-terminals: `random` non-terminals, as well as `player-1` non-terminals and `player-2` non-terminals, controlled by players 1 and 2, respectively. For each `random` non-terminal, X, we are given a probability distribution on the rules ($X \mapsto \alpha$) where X appears on the left hand side. Each grammar rule has a reward associated with it. Starting from the start non-terminal, a play of the game proceeds to build a derivation of the grammar. In each round, derivation proceeds in left-most manner, i.e., the player (or "nature", who probabilistically expands its non-terminals) expands the left-most non-terminal which remain in the derivation. The play continues, either forever, or until the empty string is derived. Player 1's goal is to maximize the total (possibly infinite) expected reward gained during the entire derivation, while player 2's goal is to minimize the total expected reward.

As explained in the introduction, serious complications arise if we allow 0 rewards on transitions. Indeed, consider the purely deterministic context-free grammar given by the rules: $\{X \overset{(\bot,0)}{\mapsto} XY \ ; \ X \overset{(\bot,0)}{\mapsto} \epsilon \ ; \ Y \overset{(\bot,7)}{\mapsto} \epsilon \}$, where X and Y are non-terminals belonging to the maximizing player, player 1 (so instead of probabilities, we have the label $\bot$). Suppose the start non-terminal is X. If the deterministic game proceeds by left-most derivation, it is easy to see that there is no optimal strategy for maximizing player 1's total payoff. Indeed, there aren't even any ϵ-optimal strategies, because the supremum is ∞. In fact, if player 1 uses the rule $X \overset{(\bot,0)}{\mapsto} XY$, n times, to expand the left-most X in the derivation, and then uses $X \overset{(\bot,0)}{\mapsto} \epsilon$, and finally uses $Y \overset{(\bot,7)}{\mapsto} \epsilon$, n times to expand all n remaining Y non-terminals, the total reward is $7 * n$. But no single strategy will gain ∞ reward. Note in particular that any "stackless and memoryless" strategy, which always picks one fixed rule for each non-terminal, regardless of the history of play and the remaining non-terminals (the "stack"), is the worst strategy possible: its total reward is 0. By contrast, if we require simultaneous expansion of all remaining non-terminals in each round, then there is a single "stackless and memoryless" strategy that gains infinite reward, namely: in each round expand every copy of X using $X \overset{(\bot,0)}{\mapsto} XY$, and (simultaneously) expand every copy of Y using its unique rule. Clearly, after $n \geq 1$ rounds we accumulate $7 * (n - 1)$ reward by doing this. Thus the total reward will be ∞.

Similarly, consider the simple grammar $\{X \overset{(\bot,0)}{\mapsto} XY \ ; \ Y \overset{(\bot,1)}{\mapsto} Y \ \}$, where, again, both non-terminals X, Y are controlled by the maximizing player, and X is the start non-terminal. Under the left-most derivation law, clearly the maximum reward is 0, whereas under the right-most or simultaneous derivation law, the total reward is ∞. So, the supremum total (expected) reward is not robust and can wildly differ, depending on the derivation law, when 0 rewards are allowed on rules.

Consider the following quantitative decision problem. Given a maximizing 1-RMDP, and $p \in [0, 1]$, is there is a strategy for player 1 (maximizer) such that with probability at least p, a desired target non-terminal, S_{target}, eventually appears as the left-most

remaining nonterminal in the derivation. This quantitative "reachability" problem not even known to be decidable. Moreover, it also easily encodes the quantitative termination problem studied in [12] (where the goal is to terminate, i.e., derive a finite string). The value for such termination probabilities can be irrational, even in the setting without players (1-RMCs), and their quantitative termination problem is at least as hard as long standing open problems in numerical computation, like the *square-root sum* problem, whose complexity is not even known to be in NP nor in the polynomial-time hierarchy. (A problem equivalent to the qualitative case of reachability for 1-RMDPs, i.e. where $p = 1$, was shown to be decidable in in P-time in [2], building on the qualitative termination results in [13].) The quantitative reachability problem for 1-RMDPs can trivially be encoded in the setting of 1-player SCFG games with non-negative rewards and leftmost derivation, as follows. We assign reward 0 to all rules, except we remove all grammar rules $S_{target} \overset{p}{\mapsto} \alpha$ whose left hand side is S_{target} and replace them with the two rules $S_{target} \overset{(1,1)}{\mapsto} S_{dead}$ and $S_{dead} \overset{(1,0)}{\mapsto} S_{dead}$. In other words, 0 reward is gained until the first time S_{target} is encountered as the left-most non-terminal in the derivation, after which reward 1 is gained, and then reward 0 is gained forever. It is easy to see that, under any strategy σ, the expected total reward in this 1-RMDP with non-negative rewards is precisely the probability that, under strategy σ, we eventually reach S_{target} as the leftmost non-terminal in the derivation. Determining whether this optimal probability is, say, greater than $1/2$, is not even known to be decidable.

Results analogous to those we give for the 1-RSSG model with strictly postive reward can be shown to hold (with modified proofs) for games over stochastic context-free grammars, even with 0 rewards allowed on rules, but with the *simultaneous* expansion derivation law (i.e., all remaining non-terminals are expanded, by their respective player, in each iteration). But our results are for the 1-RSSG model, with strictly positive rewards, where the presence of 0 rewards would change the game dramatically as the above examples illustrate.

A.4 Proof of Theorem 1

Theorem 1

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

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

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

Proof.

1. All equations in the system $P(x)$ are min-max linear with non-negative coefficients and constants, and hence are monotone.

2. The proof that $\mathbf{r}^* = P(\mathbf{r}^*)$ is similar to the one for 1-RSSG termination games from [12], but it uses in a crucial way the fact that rewards on all transitions are strictly positive.
   (a) For $u = ex \in Type_0$, $\mathbf{r}^*_u = 0$, so it fulfills the corresponding equation $x_u = 0$.
   (b) For $u \in Type_{rand}$, from the definition $r^*_u = \sup_\sigma \inf_\tau r^{*,\sigma,\tau}_u$ it follows that $\mathbf{r}^*_u = \sum_{v \in next(u)} p_{u,v}(\mathbf{r}^*_v + c_{u,v})$. Note that this holds even when some of the expected rewards are infinite, because if $p_{u,v} > 0$ and the game starting at v has infinite reward value, then this is also the case starting at u.
   (c) For $u \in Type_{call}$, $u = (b, en)$ is a call port. We claim that

$$\mathbf{r}^*_u = \mathbf{r}^*_{en} + \mathbf{r}^*_{(b,ex')} + c_u \tag{1}$$

   where ex' is the unique exit of $Y(b)$. For this we make crucial use of the assumption that rewards on all transitions are strictly positive[8]. Consider the game starting at $u = (b, en)$, as a combination of two games: the two players play inside b, starting at en, with player 1's goal to maximize the total (expected) reward. The two players also (in a "separate" game) play starting at (b, en). The payoff to player 1 is as follows: If the game inside b terminates, then the payoff is the total of the payoffs gained in both games, and if the game inside b does not terminate, then the payoff is just the payoff gained inside b.
   It should be clear that this "modified" version of the game in fact describes the same game. In particular, in the original game both players can, upon first encountering (b, ex') (in the empty context) safely ignore the history and try to maximize/minimize the payoff in the game starting at (b, ex'), without changing the reward value.
   Fix strategies for both players. What is the expected total reward starting at u? It is c_u plus the expected reward gained inside box b, plus the expected reward after exiting box b *times the probability of exiting box* b. The key point is that, since all transitions have positive reward, the only circumstance under which the expected reward value within box b is finite, i.e., $\mathbf{r}^*_{en} < \infty$, is when for every strategy of the maximizer there is a strategy for minimizer that assures finite expected reward inside b. This also necessarily assures that box b is exited with probability 1 (because otherwise, since all transitions have positive reward bounded below by some minimum value $c > 0$, infinite expected reward would be gained inside b). Consequently, equality (1) holds when $\mathbf{r}^*_{en} < \infty$. But if $\mathbf{r}^*_{en} = \infty$, then the equality holds regardless of the value of $\mathbf{r}^*_{(b,ex')}$, so it holds in all circumstances.
   (d) For $u \in Type_{\max}$, we know that $\mathbf{r}^*_u \geq \mathbf{r}^*_v + c_{u,v}$ for any $v \in next(u)$, for otherwise the *max* player would be better off taking the transition to the node v in the first step, and thereafter obtaining $\mathbf{r}^*_v + c_{u,v}$. On the other hand we also have that $\mathbf{r}^*_u \leq \mathbf{r}^*_v + c_{u,v}$ for some $v \in next(u)$, as otherwise no matter what first transition player *max* picks from u, the *min* player has a strategy such that *max* will not be able to obtain expected reward $\mathbf{r}^*_u$.

[8] We note that this assumption would be unnecessary if we were working with SCFG games (in CNF form) with simultaneous expansion. The entire proof would go through for such games even with 0 rewards on rules.

(e) For $u \in Type_{\min}$ we know that $\mathbf{r}^*_u \leq \mathbf{r}^*_v + c_{u,v}$ for all $v \in next(u)$, as otherwise it would be better for the *min* player to take the transition leading to the node v and giving to *max* player expected reward value $\mathbf{r}^*_v + c_{u,v}$ that is lower than $\mathbf{r}^*_u$. However we also have to have that $\mathbf{r}^* \geq \mathbf{r}^*_v + c_{u,v}$ for some $v \in next(u)$, because otherwise player *max* could always obtain expected reward higher than $\mathbf{r}^*_u$ no matter what *min* player does.

3. Note that P is monotonic, and $\mathbf{r}^*$ is a fixed point of P. Since $x^0 = \mathbf{0} \leq \mathbf{r}^*$, it follows by induction on k that $\mathbf{x}^k \leq \mathbf{r}^*$, for all $k \geq 0$.
4. Consider any fixed point $\mathbf{r}'$ of the equation system $P(\mathbf{x})$. We will prove that $\mathbf{r}^* \leq \mathbf{r}'$. 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 $s = \langle \beta, u \rangle$, where u belongs to player 2(*minimizer*) nodes, we choose $\tau^*(s) = \arg\min_{v \in next(u)} \mathbf{r}'_v$ (breaking ties lexicographically).

Lemma 4. *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. Base case $\mathbf{r}^{0,\sigma,\tau^*} = \mathbf{0} \leq \mathbf{r}'$ is trivial.

(a) $u = ex$, then $\mathbf{r}^{k,\sigma,\tau^*}_u = 0 = \mathbf{r}'_u$ for all $k \geq 0$.

(b) $u \in Type_{rand}$ is a random node and after we define a strategy $\sigma'(\theta) = \sigma(\langle \varepsilon, u \rangle \theta)$ we get:

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

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

(c) If $u = (b, en)$ is an entry en of the box b then we claim

$$\mathbf{r}^{k+1,\sigma,\tau^*}_u \leq \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{en} + \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{(b,ex')} + c_u \tag{2}$$

where (b, ex') is the only return port of box b. To see this, note that in any specific trajectory, the total reward gained in $k+1$ steps starting at call port (b, en) is c_u plus the remaining reward, which is split into two parts: that gained in i steps inside box b, and the rest gained in j steps after returning from box b, and such that $i + j = k$. Thus clearly the total expected reward in $k+1$ steps starting at u is no more than c_u plus the expected reward in k steps starting inside box b (i.e., starting at the entry en of $Y(b)$) plus the expected gain in k steps starting at (b, ex'). We now have

$$\max_\rho \mathbf{r}^{k,\rho,\tau^*}_{en} + \max_\rho \mathbf{r}^{k,\rho,\tau^*}_{(b,ex')} + c_u \leq \mathbf{r}'_{en} + \mathbf{r}'_{(b,ex')} + c_u = \mathbf{r}'_u \tag{3}$$

by inductive assumption, and by the fact that $\mathbf{r}'$ is a fixed point of $P(\mathbf{x})$. So, combining equations (2) and (3), we have $\mathbf{r}^{k+1,\sigma,\tau^*}_u \leq \mathbf{r}'_u$.

(d) For $u \in Type_{max}$ we claim

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

because the player has to move to some neighbor v of $\langle \varepsilon, u \rangle$ in one step, and thus it can not gain more that $\mathbf{r}^{k,\sigma',\tau^*}$, where σ' is defined from σ in the same way as for $Type_{rand}$. Thus

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

.

(e) For $u \in Type_{min}$ we know that $\tau^*(u) = \arg\min_{v \in next(u)}(\mathbf{r}'_u + c_{u,v}) = v^*$, so:

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

□

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 *max* player strategy σ, so $\sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u$. Thus for all vertices u:

$$\mathbf{r}_u^* = \sup_\sigma \inf_\tau \mathbf{r}_u^{*,\sigma,\tau} \leq \inf_\tau \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau} \leq \sup_\sigma \mathbf{r}_u^{*,\sigma,\tau^*} \leq \mathbf{r}'_u \tag{4}$$

5. In equation (4) above, choose $\mathbf{r}' = \mathbf{r}^*$. Then we have, for all vertices u,

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

6. We know that $\mathbf{z} = \lim_{k\to\infty} \mathbf{x}^k$ exists in $(0, \infty]$, because it is a monotonically non-decreasing sequence (note some entries may be infinite). In fact we have $\mathbf{z} = \lim_{k\to\infty} P^{k+1}(\mathbf{0}) = P(\lim_{k\to\infty} P^k(\mathbf{0}))$, and thus $\mathbf{z}$ is a fixed point of the equation $P(\mathbf{x}) = \mathbf{x}$. 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 $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{r}^*$.

□

Corollary 1. *In every positive reward 1-RSSG, the* minimizer *has an optimal deterministic Stackless and Memoryless (SM) strategy.*

Proof. It is enough to consider the strategy τ^*, from Part 4 of Theorem 1, when we let $\mathbf{r}' = \mathbf{r}^*$. For then, by equation (1), we have $r_u^* = \sup_\sigma r^{*,\sigma,\tau^*} = \inf_\tau \sup_\sigma r^{*,\sigma,\tau^*}$. □

A.5 Proof of Lemma 1

Lemma 1 *For any $x \in \mathbb{R}^n_{\geq 0}$, $A \in (\mathbb{R}^\infty_{\geq 0})^{n\times n}$ and $b \in (\mathbb{R}^\infty_{\geq 0})^n$, if $x \leq Ax + b$ then*

$$x \leq (\sum_{k=0}^{\infty} A^k) b$$

This holds even if for some indices i we have $b_i = 0$, as long as the entries in such rows i of the matrix A are all zero.

Proof. Let $D = \sum_{k=0}^\infty A^k$ and $y = Db$. We have to prove that $x \leq y$. Some of the entries of D can be infinite. Let $R = \{r_1, r_2, \ldots, r_m\}$ be the set of indices of the rows of D that contain at least one ∞ entry. For every $r \in R$, $y_r = \sum_{i=1}^n D_{r,i} b_i$. Since for all i, $b_i > 0$, and for at least one i $D_{r,i}$ is ∞, we have $y_r = \infty$ and so $x_r \leq y_r$ is trivially fulfilled for

every $r \in R$. Now let us construct a new matrix A' by zeroing all the entries of the rows of A that are in R. Similarly let x' be a vector x with zeroed entries x_r where $r \in R$. Let $D' = \sum_{k=0}^{\infty} A'^k$.

We will prove that $x' \leq A'x' + b$. For entries $r \in R$, it is trivial as $(A'x')_r + b_r = 0 + b_r \geq 0 = x'_r$. If $r \notin R$ then $x'_r = x_r$ and

$$(A'x')_r = \sum_{i=1}^{n} A'_{r,i}x'_i = \sum_{\{i|A'_{r,i}>0\}} A'_{r,i}x'_i$$

Proposition 3. *If $A_{i,j} > 0$, and for some k we have that $D_{j,k} = \infty$ then $D_{i,k} = \infty$.*

Proof. We have that $D = I + AD$ and so $D_{i,k} = \delta_{ik} + \sum_{l=1}^{n} A_{i,l}D_{l,k} \geq A_{i,j}D_{j,k} = \infty$. (where δ_{ik} is equal to 1 if $i = k$ and 0 otherwise) □

Suppose that $r \notin R$. If for some i, $x'_i \neq x_i$, then $i \in R$ and we must have $D_{i,j} = \infty$ for some j. If $A'_{r,i} > 0$ then $A_{r,i} = A'_{r,i}$, and from Proposition 3 we get that $D_{r,j} = \infty$, which contradicts the fact that $r \notin R$. Thus for $r \notin R$, and for i such that $A'_{r,i} > 0$, we must have $x'_i = x_i$ and $A'_{r,i} = A_{r,i}$. Thus $(A'x')_r + b_r = (Ax)_r + b_r \geq x_r = x'_r$ for all $r \notin R$. Hence we can conclude that $x'_r \leq (A'x')_r + b_r$ for all r.

We will now prove that $\lim_{k\to\infty} A'^k = 0$. For contradiction, note that if we had $\lim_{k\to\infty}(A'^k)_{i,j} \neq 0$ for some i, j then it must be the case that $D'_{i,j} = \infty$ because $(A'^k)_{i,j} \geq 0$ for all k, and if there is some $\epsilon > 0$ such that for infinitely many k, $(A')^k_{i,j} > \epsilon$, then $D'_{i,j} = \infty$. Since $A' \leq A$, we get that $A'^k \leq A^k$ for any $k \geq 0$ and thus $\sum_{k=0}^{\infty} A'^k \leq \sum_{k=0}^{\infty} A^k$. Thus if $D'_{i,j} = \infty$ then $D_{i,j} = \infty$. Hence all entries in the i-th row of A must have been zeroed to obtained A'. However if the i-th row in A' has all zeroes, then so does the i-th row in A'^k for any k. That contradicts the assumption that $\lim_{k\to\infty}(A'^k)_{i,j} \neq 0$.

By substituting x' by $A'x' + b$ in $x' \leq A'x' + b$, we get that $x' \leq A'x' + b \leq A'(A'x' + b) + b = A'^2x' + A'b + b \leq A'^2(A'x' + b) + A'b + b = A'^3x' + (A'^2 + A' + I)b$ and by iterating we see that $x' \leq A'^{l+1}x' + (\sum_{k=0}^{l}(A')^k)b$ for any $l \geq 0$. As x' is a vector of finite values and $\lim_{k\to\infty} A'^k = 0$ we have $x' \leq (\sum_{k=0}^{\infty}(A')^k)b$ and so also $x' \leq (\sum_{k=0}^{\infty}(A)^k)b = y$. The last fact proves that for $r \notin R$ $x_r = x'_r \leq y_r$, and we can finally conclude that $x \leq y = (\sum_{k=0}^{\infty} A^k)b$.

Now we show that we can also handle the case when for some indices i, $b_i = 0$.

We proceed by induction on the number, d, of indices i such that $b_i = 0$ and the i-th row of A is all zero. For the base case $d = 0$, the claim was already proved. For the inductive case, suppose $d > 0$, and let i be the smallest such index. Since we assume $Ax + b \geq x$, it must be that $x_i = 0$. Let M' denote the matrix obtained by removing the i-th row and the i-th column in some matrix M. Similarly for a vector v by v' denote the vector v with removed i-th entry. Since $x_i = 0$, $M'x' = (Mx)'$ for any matrix M. Also, since the i-th row of A is all zeroes we have that $(A')^k = (A^k)'$ for any $k \geq 0$ and we can also conclude that $\sum_{k=0}^{\infty}(A')^k = (\sum_{k=0}^{\infty} A^k)'$. Now assuming $Ax + b \geq x$ we can see that $(Ax + b)' \geq x'$ and so $A'x' + b' \geq x'$. But it is easy to confirm that A' and b' have the same property as before: if $b'_j = 0$ then the j-th row of A' is all zero. Moreover, there are now $d-1$ such indices. Thus, by inductive hypothesis, $x' \leq (\sum_{k=0}^{\infty}(A')^k)b' = (\sum_{k=0}^{\infty} A^k)'b' = ((\sum_{k=0}^{\infty} A^k)b)'$, and since the inequality is trivial for the i-th position of x, we get that $x \leq ((\sum_{k=0}^{\infty} A^k)b)$. □

A.6 Proof of Theorem 2

Proof. Let σ be any SM strategy for player 1. Consider $\mathbf{r}^{*,\sigma}_u = \inf_{\tau\in\Psi_2} \mathbf{r}^{*,\sigma,\tau}_u$. (Note that some entries in the vector $\mathbf{r}^{*,\sigma}$ may be ∞.) First, note that if $\mathbf{r}^{*,\sigma} = P(\mathbf{r}^{*,\sigma})$ then $\mathbf{r}^{*,\sigma} = \mathbf{r}^*$. This is because, by Theorem 1, $\mathbf{r}^* \leq \mathbf{r}^{*,\sigma}$, and on the other hand, σ is just one strategy for player 1, and for every vertex u, $\mathbf{r}^*_u = \sup_{\sigma'\in\Psi_1} \mathbf{r}^{*,\sigma'}_u \geq \mathbf{r}^{*,\sigma}_u$. Now we claim that, for all vertices u such that $u \notin Type_{\max}$, $\mathbf{r}^{*,\sigma}_u$ satisfies its equation in $\mathbf{x} = P(\mathbf{x})$. In other words, $\mathbf{r}^{*,\sigma}_u = P_u(\mathbf{r}^{*,\sigma})$. To see this, note that for vertices u of Types $\{0, call, rand\}$, no choice of either player is involved and the equation holds by definition of $\mathbf{r}^{*,\sigma}$ (In particular, the expected reward value at a call u is c_u plus the sum of the expected reward values of the game starting at the entry inside the box, and the game starting at the return port.) For nodes $u \in Type_{\min}$, we have the equation $\mathbf{x}_u = \min_{v\in next(u)} \mathbf{x}_v + c_{u,v}$. But note that the best minimizer can do against strategy σ, starting at $\langle\epsilon, u\rangle$, is to move to a neighboring vertex v such that $v = \arg\min_{v\in next(u)}(\mathbf{r}^{*,\sigma}_v + c_{u,v})$. Thus, the only equations that may fail are those for $u \in Type_{max}$, $\mathbf{x}_u = \max_{v\in next(u)}(\mathbf{x}_v + c_{u,v})$. Suppose $\sigma(u) = v$, for some neighbor v. Clearly then, $\mathbf{r}^{*,\sigma}_u = \mathbf{r}^{*,\sigma}_v + c_{u,v}$. Thus, $\mathbf{r}^{*,\sigma}_u \leq \max_{v'\in next(u)}(\mathbf{r}^{*,\sigma}_{v'} + c_{u,v'})$. Thus equality fails iff there is another vertex $w \neq v$, with $(u, \perp, w) \in \delta$, such that $\mathbf{r}^{*,\sigma}_v + c_{u,v} < \mathbf{r}^{*,\sigma}_w + c_{u,w}$.

Suppose now that the nodes $(u_1, u_2, \ldots u_n)$ are all those nodes where the *SM* strategy σ is not locally optimal, i.e., for $i = 1, 2, \ldots, n$, $\sigma(u_i) = v_i$, and thus $\mathbf{r}^{*,\sigma}_{u_i} = \mathbf{r}^{*,\sigma}_{v_i} + c_{u_i,v_i}$, but there is some w_i such that $\mathbf{r}^{*,\sigma}_{v_i} + c_{u_i,v_i} < \mathbf{r}^{*,\sigma}_{w_i} + c_{u_i,w_i}$. Let $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and similarly define $\mathbf{v}$ and $\mathbf{w}$. Consider now a revised *SM* strategy σ', which is identical to σ, except that $\sigma'(u_i) = w_i$ for all i. Next, consider a parametrized 1-exit RSSG, $A(\mathbf{t})$ where $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, which is identical to A, except that all edges out of vertices u_i are removed, and replaced by a single probability 1 edge labeled by reward t_i, to the exit of the same component node u_i is in. Fixing the value of the vector $\mathbf{t} \in [0, \infty]^n$ determines an 1-RSSG, $A(\mathbf{t})$. Note that if we restrict *SM* strategies σ or σ' to vertices other than those in $\mathbf{u}$, then they both define the same *SM* strategy for the 1-RSSG $A(\mathbf{t})$. Define $r^{*,\sigma,\tau,\mathbf{t}}_z$ to be the expected total reward starting from $\langle\epsilon, z\rangle$ in the Markov chain $M^{z,\sigma,\tau}_{A(\mathbf{t})}$. Now, for each vertex z, define the function $f_z(\mathbf{t}) = \inf_{\tau\in\Psi_2} r^{*,\sigma,\tau,\mathbf{t}}_z$. In other words, $f_z(\mathbf{t})$ is the infimum of the expected rewards, over all strategies of player 2, starting at $\langle\epsilon, z\rangle$ in $A(\mathbf{t})$. This reward is parametrized by $\mathbf{t}$. Now, let $\mathbf{t}^\sigma$ be a vector such that $\mathbf{t}^\sigma_{u_i} = \mathbf{r}^{*,\sigma}_{u_i}$, and observe that $f_z(\mathbf{t}^\sigma) = \mathbf{r}^{*,\sigma}_z$ for every z. This is so because any strategy for minimizing the total reward starting from z would, upon hitting a state $\langle\beta, u_i\rangle$ in some arbitrary context β, be best off minimizing the total expected reward starting from $\langle\beta, u\rangle$ until that context is exited, (and unless the minimizer has a strategy that assures the context is exited with probability 1, the expected reward will be ∞).

Note that, by Corollary 1, in the 1-RSSG reward game on $A(\mathbf{t})$, for any values in vector $\mathbf{t}$, and any start vertex z, minimizer has an optimal SM strategy $\tau_{z,\mathbf{t}}$, such that $\tau_{z,\mathbf{t}} = \arg\min_{\tau\in\Psi_2} r^{*,\sigma,\tau,\mathbf{t}}_z$. Let $g_{(z,\tau)}(\mathbf{t}) = r^{*,\sigma,\tau,\mathbf{t}}_z$. Note that $f_z(\mathbf{t}) = \min_\tau g_{z,\tau}(\mathbf{t})$, where the minimum is over *SM* strategies. Now, note that the function $g_{z,\tau}(\mathbf{t})$ is the expected reward in a positive reward 1-RMC starting from a particular vertex, and it is given by $g_{z,\tau}(\mathbf{t}) = (\lim_{k\to\infty} R^k(\mathbf{0}))_z$ for a linear system $\mathbf{x} = R(\mathbf{x})$ with non-negative coefficients in R, where $R(\mathbf{x}) = A_{\sigma,\tau}\mathbf{x} + b_{\sigma,\tau}(\mathbf{t})$, for some nonnegative matrix $A_{\sigma,\tau}$, and vector $b_{\sigma,\tau}(\mathbf{t})$ which describes the average 1-step rewards from each vertex. All of these 1-step rewards are positive, except that at positions u_i the entry is the variable t_i, i.e., $b_{u_i}(\mathbf{t}) = t_i$. (Note that for all i the u_i'th row vector of $A_{\sigma,\tau}$ is all zero.) Simple iteration then shows

that $g_{z,\tau}(\mathbf{t}) = \lim_{k\to\infty} R^k(\mathbf{0})_z = ((\sum_{k=0}^{\infty} A_{\sigma,\tau}^k)b(\mathbf{t}))_z$. (Note that if $\lim_{k\to\infty} A_{\sigma,\tau}^k = 0$, then $(\sum_{k=0}^{\infty} A_{\sigma,\tau}^k) = (I - A_{\sigma,\tau})^{-1}$.) Now $g_{z,\tau}(\mathbf{t})$ has the following properties: it is a continuous, nondecreasing, and linear function of $\mathbf{t} \in [0,\infty]^n$, and for $\mathbf{t} \in [0,\infty]^n$, $g_{z,\tau}(\mathbf{t}) \in [0,\infty]$. Specifically, we can think of it as a function $g_{z,\tau}(\mathbf{t}) = \boldsymbol{\alpha}^{z,\tau}\mathbf{t} + \beta^{z,\tau}$, where $\boldsymbol{\alpha}^{z,\tau} = (\alpha_1^{z,\tau}, \alpha_2^{z,\tau}, \ldots, \alpha_n^{z,\tau})$ and $\alpha_i^{z,\tau}, \beta^{z,\tau} \in [0,\infty]$.

Let $\mathbf{g}^\tau(\mathbf{t}) = (g_{w_1,\tau}(\mathbf{t}'), g_{w_2,\tau}(\mathbf{t}'), \ldots, g_{w_n,\tau}(\mathbf{t}'))$ where $\mathbf{t}' = \mathbf{t} + \mathbf{c}^{\mathbf{u},\mathbf{w}}$ and $\mathbf{c}^{\mathbf{u},\mathbf{w}} = (c_{u_1,w_1}, c_{u_2,w_2}, \ldots, c_{u_n,w_n})$. Note $\mathbf{t} \in (-c_{u_1,w_1}, \infty] \times (-c_{u_2,w_2}, \infty] \times \ldots \times (-c_{u_n,w_n}, \infty]$. We can represent $\mathbf{g}^\tau(\mathbf{t})$ as $D^\tau\mathbf{t} + \mathbf{d}^\tau$, where $D^\tau = [\boldsymbol{\alpha}^{w_1,\tau}; \boldsymbol{\alpha}^{w_2,\tau}; \ldots; \boldsymbol{\alpha}^{w_n,\tau}]$ and $\mathbf{d}_j^\tau = \sum_{i=0}^{n} \alpha_i^{w_j,\tau} c_{u,w_i} + \beta^{w_j,\tau}$. Note that if $\mathbf{d}_j^\tau = 0$ then $\boldsymbol{\alpha}^{w_j,\tau} = \mathbf{0}$ and $\beta^{w_j,\tau} = 0$.

Consider function $\mathbf{f}(\mathbf{t}) = \min_\tau \mathbf{g}^\tau(\mathbf{t})$. This is well defined, since whatever the values in $\mathbf{t}$, the *min* player always has, by Corollary 1, an optimal *SM* strategy τ^* in $A(\mathbf{t})$ such that for any strategy σ of the *max* player, and any strategy τ of the *min* player, and all z we have $r_z^{*,\sigma,\tau^*,\mathbf{t}} \le r_z^{*,\sigma,\tau,\mathbf{t}}$. Note that $\mathbf{f}(\mathbf{t}) = (f_{w_1}(\mathbf{t}+\mathbf{c}^{\mathbf{u},\mathbf{w}}), f_{w_2}(\mathbf{t}+\mathbf{c}^{\mathbf{u},\mathbf{w}}), \ldots, f_{w_n}(\mathbf{t}+\mathbf{c}^{\mathbf{u},\mathbf{w}}))$.

Lemma 5. *If $\mathbf{f}(\mathbf{t}) > \mathbf{t}$ for some finite vector $\mathbf{t}$, then for any fixed point $\mathbf{t}^*$ of $\mathbf{f}$, $\mathbf{t} \le \mathbf{t}^*$.*

Proof. Suppose that $\mathbf{t}^*$ is some fixed point of $\mathbf{f}$. Since $\mathbf{f}(\mathbf{t}^*) = \min_\tau \mathbf{g}^\tau(\mathbf{t}^*)$, for some τ^* we have $\mathbf{g}^{\tau^*}(\mathbf{t}^*) = \mathbf{t}^*$. From the fact that $\mathbf{f}(\mathbf{t}) > \mathbf{t}$, we get that for all τ we have $\mathbf{g}^\tau(\mathbf{t}) > \mathbf{t}$. In particular we have $\mathbf{g}^{\tau^*}(\mathbf{t}) > \mathbf{t}$, which means that $D^{\tau^*}\mathbf{t} + \mathbf{d}^{\tau^*} > \mathbf{t}$. Now, for all i, either $\mathbf{d}_i^{\tau^*} = 0$ and the i-th row in D^{τ^*} is all zeroes, or $\mathbf{d}_i^{\tau^*} > 0$, thus from Lemma 1 we can conclude that $\mathbf{t} \le \sum_{k=0}^{\infty}(D^{\tau^*})^k\mathbf{d}^{\tau^*}$. However, letting $h(\mathbf{t}) = \mathbf{g}^{\tau^*}(\mathbf{t}) = D^{\tau^*}\mathbf{t} + \mathbf{d}^{\tau^*}$ be the linear operator on $[0,\infty]^n$, note that the least fixed point solution (in $[0,\infty]^n$) of $h(\mathbf{t})$ is $\mathbf{t}_0 = \lim_{k\to\infty} h^{k+1}(\mathbf{0}) = \lim_{k\to\infty} D^{\tau^*}h^k(\mathbf{0}) + \mathbf{d}^{\tau^*} = \sum_{k=0}^{\infty}(D^{\tau^*})^k\mathbf{d}^{\tau^*}$. Thus, any other fixed point of h has to be greater than $\mathbf{t}_0$ and in particular $\mathbf{t}^* \ge \mathbf{t}_0 \ge \mathbf{t}$. □

Now, we know that $\mathbf{f}(\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i = f_{w_i}(\mathbf{t}^\sigma) = \mathbf{r}_{w_i}^{*,\sigma} > \mathbf{r}_{v_i}^{*,\sigma} + c_{u_i,v_i} - c_{u_i,w_i} = \mathbf{r}_{u_i}^{*,\sigma} - c_{u_i,w_i} = (\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i$ which proves that $\mathbf{f}(\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}) > \mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}$. Therefore, by Lemma 5, any fixed point of $\mathbf{f}$ has to be greater or equal to $\mathbf{t}^\sigma - \mathbf{c}^{\mathbf{u},\mathbf{w}}$. Also, if we switch strategy σ to σ', then $\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}}$ is a fixed point of $\mathbf{f}$ because $\mathbf{f}(\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i = f_{w_i}(\mathbf{t}^{\sigma'}) = \mathbf{r}_{w_i}^{*,\sigma'} = \mathbf{r}_{u_i}^{*,\sigma'} - c_{u_i,w_i} = (\mathbf{t}^{\sigma'} - \mathbf{c}^{\mathbf{u},\mathbf{w}})_i$. Thus $\mathbf{t}^\sigma \le \mathbf{t}^{\sigma'}$. Since $\mathbf{f}$ is non-decreasing, then $\mathbf{r}_z^{*,\sigma'} = f_z(\mathbf{t}^{\sigma'}) \ge f_z(\mathbf{t}^\sigma) = \mathbf{r}_z^{*,\sigma}$ for any z, and for $u_1, u_2, \ldots, u_n$ the inequality is strict: $\mathbf{r}_{u_i}^{*,\sigma'} - c_{u_i,w_i} = \mathbf{r}_{w_i}^{*,\sigma'} \ge \mathbf{r}_{w_i}^{*,\sigma} > \mathbf{r}_{v_i}^{*,\sigma} + c_{u_i,v_i} - c_{u_i,w_i} = \mathbf{r}_{u_i}^{*,\sigma} - c_{u_i,w_i}$.

Thus, switching to the new *SM* strategy σ', we get $\mathbf{r}^{*,\sigma'}$ which dominates $\mathbf{r}^{*,\sigma}$, and is strictly greater in some coordinates, including all the u_i's. There are finitely many *SM* strategies, thus repeating this we eventually reach some *SM* strategy σ^* that can't be improved. Thus $\mathbf{r}^{*,\sigma^*} = P(\mathbf{r}^{*,\sigma^*})$, and by our earlier claim $\mathbf{r}^{*,\sigma^*} = \mathbf{r}^*$. Thus, maximizer has an optimal *SM* strategy, arrived at via simultaneous strategy improvement. □

A.7 Proof of Proposition 1

Proposition 1 *Let U be any proper set. (I) The vector $\mathbf{r}^*|_U$ is the LFP of $P|_U$. (II) If $r_u^* = \infty$ for some vertex u in an SCC S of G, then $r_v^* = \infty$ for all $v \in S$. (III) If there is an optimal bounded solution $\mathbf{r}'$ to the* max-LP$|_U$ *then it has to be a fixed point of the max-linear operator $P|_U$. (IV) If max-LP$|_U$ has a bounded optimal feasible solution $\mathbf{r}'$ then $\mathbf{r}' = \mathbf{r}^*|_U$.*

Proof. Part (I) follows immediately from definitions. Part (II) follows by induction on the length of the shortest path from any vertex $v \in S$ to u. In particular, if $x_v = \max\{x_w, \ldots\}$, and $r^*_w = \infty$, then $r^*_v = \infty$, and likewise for other vertex types. For part (III), observe that for each vertex $u \in Type_{\max}$, if $\mathbf{r}'$ is an optimal bounded solution of the max-LP, then at least one of the constraints $x_u \geq x_v + c_{u,v}$ holds *tightly*, i.e., $x_u = x_v + c_{u,v}$. For otherwise, we could decrease the value of x_u, letting $x_u = \max_{v \in next(u)}(x_v + c_{u,v})$, and still satisfy all constraints. The fact that the other types of inequalities are satisfied tightly follows similarly. For part (IV), if max-LP$|_U$ has a feasible bounded solution, then the optimal (minimum) solution $\mathbf{r}'$ is bounded. From part (III), we know $\mathbf{r}'$ is a fixed point of $P|_U$, but then from the objective function of max-LP$|_U$, we know that $\mathbf{r}'$ is the LFP of $P|_U$, so we must have $\mathbf{r}' = \mathbf{r}^*|_U$. □

A.8 Proof of Proposition 2

Proposition 2 *(I)* $\mathbf{r}^* = (\sum_{k=0}^{\infty} A_\tau^k) b_\tau$. *(II) If* $\mathbf{r}^*$ *is finite, then* $\lim_{k\to\infty} A_\tau^k = 0$, *and thus* $(I - A_\tau)^{-1} = \sum_{i=0}^{\infty}(A_\tau)^i$ *exists (i.e., is a finite real matrix).*

Proof. (I): $\mathbf{r}^* = \lim_{k\to\infty}(P'_\tau)^{k+1}(0) = \lim_{k\to\infty} A_\tau (P'_\tau)^k(0) + b_\tau = \lim_{k\to\infty}(\sum_{i=0}^{k}(A_\tau)^k) b_\tau$. (This holds regardless of whether $\mathbf{r}^*$ is finite. We shall use this fact in a subsequent proof.) (II): since $\mathbf{r}^* = P'_\tau(\mathbf{r}^*)$, we have, for any $k \geq 0$, $\mathbf{r}^* = A_\tau^k \mathbf{r}^* + (I + A_\tau + A_\tau^2 + \ldots + A_\tau^{k-1}) b_\tau$. The second part of the right hand side, in the limit, is equal to $\mathbf{r}^*$, thus $A_\tau^k \mathbf{r}^*$ in the limit is an all-zero vector. It follows that the limit of A_τ^k is an all-zero matrix since all the entries/rewards in $\mathbf{r}^*$ are positive (we have already removed 0 entries). □

A.9 Proof of Theorem 5

We first need some preliminary claims. Let W be the set of vertices u such that u-min-LP$'$ is bounded and let S be the minimum *proper* set such that $W \subseteq S$. From min-LP remove all the constraints for variables outside of the set S and remove the variables of $Type_0$ in the same way as before. Call this set of constraints LP$_S$.

Proposition 4. *For any two vectors* $\mathbf{x} = [x_1, x_2, \ldots, x_n], \mathbf{y} = [y_1, y_2, \ldots, y_n]$ *and vector* $\mathbf{z} = \max(\mathbf{x}, \mathbf{y}) = [\max(x_1, y_1), \max(x_2, y_2), \ldots, \max(x_n, y_n)]$, *and subset* $A \subseteq \{1, 2, \ldots, n\}$, *and constants* $p_{ij} \geq 0, c_{i,j} \geq 0$ *we have that:*

1. *if vectors* $\mathbf{x}, \mathbf{y}$ *fulfil a linear constraint* $\widetilde{x}_i \leq \sum_{j\in A} p_{ij}(\widetilde{x}_j + c_{i,j})$ *then so does* $\mathbf{z}$
2. *if vectors* $\mathbf{x}, \mathbf{y}$ *fulfil a constraint* $\widetilde{x}_i \leq \min_{j\in A}(\widetilde{x}_j + c_{i,j})$ *then so does* $\mathbf{z}$

Proof. 1. Function *max* is monotonic, hence if $x_i \leq x_j$ and $y_i \leq y_j$, then $\max(x_i, y_i) \leq \max(x_j, y_j)$. Thus $\max(x_i, y_i) \leq \max\big(\sum_{j\in A} p_{ij}(x_j + c_{i,j}), \sum_{j\in A} p_{ij}(y_j + c_{i,j})\big)$ based on the fact that they fulfil the underlying constraint. However we know that for all j we have that $x_j \leq \max(x_j, y_j) = z_j$ and $y_j \leq \max(x_j, y_j) = z_j$, hence $\sum_{j\in A} p_{ij}(x_j + c_{i,j}) \leq \sum_{j\in A} p_{ij}(z_j + c_{i,j})$ and $\sum_{j\in A} p_{ij}(y_j + c_{i,j}) \leq \sum_{j\in A} p_{ij}(z_j + c_{i,j})$, which means that $z_i = \max(x_i, y_i) \leq \max\big(\sum_{j\in A} p_{ij}(x_j + c_{i,j}), \sum_{j\in A} p_{ij}(y_j + c_{i,j})\big) \leq \sum_{j\in A} p_{ij}(z_j + c_{i,j})$

2. Again we know that $\max(x_i, y_i) \leq \max\big(\min_{j\in A}(x_j + c_{i,j}), \min_{j\in A}(y_j + c_{i,j})\big)$ and for all j we have $x_j + c_{i,j} \leq z_j + c_{i,j}$ and $y_j + c_{i,j} \leq z_j + c_{i,j}$. We also know that the *min* function is monotonic, hence $\min_{j\in A}(x_j + c_{i,j}) \leq \min_{j\in A}(z_j + c_{i,j}) \geq \min_{j\in A}(y_j + c_{i,j})$. This means that $z_i = \max(x_i, y_i) \leq \max\big(\min_{j\in A}(x_j + c_{i,j}), \min_{j\in A}(y_j + c_{i,j})\big) \leq \min_{j\in A}(z_j + c_{i,j})$.

□

Corollary 2. *For any two feasible solutions* $\mathbf{x}, \mathbf{y}$ *to* LP_S *we have that* $\mathbf{z} = \max(\mathbf{x}, \mathbf{y}) = [\max_i(\mathbf{x}_i, \mathbf{y}_i)]$ *(vector with entries being the maximum of the respective entries in* $\mathbf{x}$ *and* $\mathbf{y}$*) is a feasible solution to* LP_S *as well.*

Theorem 5 *In a minimizing reward 1-RMDP, for all vertices* u*, the value* $\mathbf{r}^*_u$ *is finite iff* u*-min-LP*$'$ *is feasible and bounded. (And thus, combining this with Lemma 3, we can compute the value of minimizing reward 1-RMDPs in P-time.*

Proof. ($\Rightarrow$) First let us show that for any u if $\mathbf{r}^*_u$ is finite, then u-min-LP$'$ has to be feasible and bounded. Feasibility is easy as an all zero vector $\mathbf{0}$ fulfills all the constraints in u-min-LP$'$.

Now pick the optimal *SM* strategy τ for the *min* player that yields the optimal reward vector $\mathbf{r}^*$ and take any feasible vector $\mathbf{x}$. From the u-min-LP$'$ we can see that $\mathbf{x} \leq A_\tau \mathbf{x} + b_\tau$ (because this is just a subset of the constraints). Since we removed all zero reward nodes ie. exits of components, then all entries of b_τ are positive and from Lemma 1 we can get that $\mathbf{x} \leq (\sum_{k=0}^{\infty} A_\tau^k) b_\tau$. However by Proposition 2 (I) (which holds regardless of whether $\mathbf{r}^*$ is finite) this means that $\mathbf{x} \leq \mathbf{r}^*$ for any feasible $\mathbf{x}$.

For contradiction, assume u-min-LP$'$ was feasible but unbounded. Then there would exist a sequence of feasible vectors $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$ such that $\lim_{k\to\infty} \mathbf{x}^k_u = \infty$. But we know that $\mathbf{x}^k \leq \mathbf{r}^*$ for all k, thus $\mathbf{r}^*_u$ would have to be infinite, contradicting our assumption.

($\Leftarrow$) Now let us show that if u-min-LP$'$ is feasible and bounded then $\mathbf{r}^*_u$ has to be finite. Consider an LP with LP_S constraints and with objective: $maximize \sum_{u\in W} x_u$. Call it W-min-LP and for any optimal solution $\mathbf{x}^*$ denote by $\overline{\mathbf{x}}^*$ the vector filled with values from $\mathbf{x}^*$ for $u \in W$ and ∞ for all $u \in S \setminus W$. Notice that $\overline{\mathbf{x}}^*$ is unique, because if two different optimal vectors $\mathbf{x}$ and $\mathbf{x}'$ differ at a value of some variable $x_u \in W$ then $\max(\mathbf{x}, \mathbf{x}')$ would also be feasible thanks to Corollary 2, and this contradicts optimality.

Lemma 6. *The vector* $\overline{\mathbf{x}}^*$ *is a fixed point of* $P|_S$*.*

Proof. Since for every variable $u \in W$, u-min-LP$'$ is bounded, and we removed from u-min-LP$'$ only the constraints that these variables do not depend on (even in a transitive way), the maximum value of x_u can not possibly increase after we remove these constraints, because that would mean x_u could have obtained a higher value in u-min-LP$'$. Hence the LP W-min-LP is feasible and bounded.

Now we show that for an optimal solution $\mathbf{x}^*$ no constraint with a variable $x_u \in W$ on the left hand side can hold tightly (i.e., with equality) when there is some variable from $S \setminus W$ on the right hand side. Let us take some optimal solution $\mathbf{x}^*$ to W-min-LP. Let $S \setminus W = \{v_1, v_2, \ldots, v_n\}$ be the set of unbounded variables, i.e., v_i-min-LP is unbounded. We know that for each of them there is a sequence of feasible solutions $\mathbf{x}^{v_i}_{\mathbf{1}}, \mathbf{x}^{v_i}_{\mathbf{2}}, \mathbf{x}^{v_i}_{\mathbf{3}}, \ldots$ to v_i-min-LP(the bold subscripts denote the position in this sequence, not inside the vector), such that the value of entry x_{v_i} in this sequence of

vectors is nondecreasing and becomes arbitrarily large. If we project this sequence to the variables in S then $\mathbf{x_1}^{v_i}|_S, \mathbf{x_2}^{v_i}|_S, \mathbf{x_3}^{v_i}|_S, \ldots$ is a sequence of feasible solutions to W-min-LP, such that v_i becomes arbitrarily large. Now construct a sequence of vectors $\mathbf{x'_i} = \max(\mathbf{x}^*, \mathbf{x_i}^{v_1}|_S, \mathbf{x_i}^{v_2}|_S, \ldots, \mathbf{x_i}^{v_n}|_S)$. By Corollary 2 we know that all vectors in this sequence are feasible solutions to W-min-LP. We also know that all of them are optimal solutions, because we always take the maximum of the entries, including in the optimal solution $\mathbf{x}^*$. So we obtain as high a value of objective function $\sum_{u \in W} x_u$ as before, and we can not improve this value as it would contradict the assumption that $\mathbf{x}^*$ was optimal. Now notice three things:

1. Since every variable $x_u \in W$ is bounded, at some point in this sequence, we will reach a point such that the r.h.s. of any constraint which involves some variable $x_u \in S \setminus W$ will be larger than the highest possible value of all variables in W. This means that at that point there can not be a constraint that holds with equality such that $x_u \in W$ is the l.h.s. and where there is a variable from $S \setminus W$ on the r.h.s.
2. For all k, for every $x_u \in W$ there has to be some constraint with x_u on the l.h.s. such that $\mathbf{x'_k}$ satisfies this constraint tightly, with equality, because otherwise we can increase the value of x_u without altering the value of any other variables, to obtain a larger value for the objective, which contradicts the optimality of $\mathbf{x'_k}$.
3. All variables $x_v \in S \setminus W$ become arbitrarily large in this sequence, thus it can not be the case that there are only variables from W on the r.h.s. in any constraint with x_v on the l.h.s. (that would force this variable to be bounded).

Using these facts, we can see that for a large enough k, from the vector $\mathbf{x'_k}$ we can construct a vector $\overline{\mathbf{x}}^*$ which a fixed point of $P|_S$. We do so by setting all variables in $S \setminus W$ to ∞, and leaving the variables in W unchanged from $\mathbf{x'_k}$. The claim that $\overline{\mathbf{x}}^*$ is a fixed point of $P|_S$ follows because for every variable $x_u \in W$ of type $Type_{rand}$ or $Type_{call}$, $\mathbf{x'_k}$ satisfies the correlated constraint with x_u on the l.h.s. with equality, and this can only be the case if the r.h.s. of that constraint contains only variables in W, and thus $\overline{\mathbf{x}}^*$ also satisfies this constraint with equality. Likewise, for variables x_u in W of type $Type_{min}$, for $\mathbf{x'_k}$ all constraints such that x_u is the l.h.s. and there is at least one variable from $S \setminus W$ on the r.h.s., must hold with strict inequality. Hence, since equality must hold in $\mathbf{x'_k}$ for one of the constraints involving x_u on the l.h.s., there must exist one such constraint such that the r.h.s. only involves variables in W. Thus, equality also holds for these constraints for $\overline{\mathbf{x}}^*$ for these variables. Thus $\overline{\mathbf{x}}^*$ satisfies the corresponding *min* equation in $P|_S$. Also for variables in $x_v \in S \setminus W$ all the equations in $P|_S$ will clearly be fulfilled after setting their values to ∞, because both sides of the equations correlated to x_v have at least one variable from $S \setminus W$, and that makes both sides have value ∞. □

Now finally we can finish the proof of the theorem using the previous lemma. Since we know that $\mathbf{r}^*|_S$ is the LFP of the operator $P|_S$, it has to be the case that $\mathbf{r}^*|_S \leq \overline{\mathbf{x}}^*$, which means that for all $u \in W$ we have that $\mathbf{r}^*_u|_S \leq \overline{\mathbf{x}}^*_u = \mathbf{x}^*_u$, which is finite. □

A.10 Proof of Theorem 7

Proof. Consider the standard 1-RMC from [11], depicted in Figure 2. From the entry, *en*, this 1-RMC goes with probability p_1 to a sequence of two boxes labeled by the same

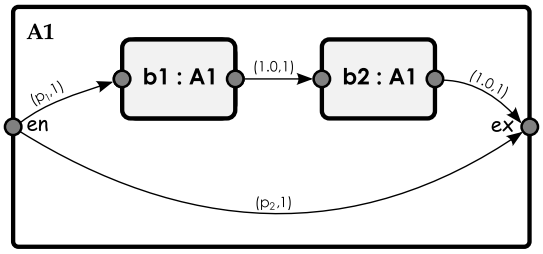

Fig. 2. Standard 1-RMC gadget used in proof of Theorem 7

component and with probability p_2 goes to the exit. We assume $p_1 + p_2 = 1$. As shown in ([11], 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, and a 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. We can do this in polynomial time. (If u is among these nodes, we would already be done, but assume it is not.) The revised SSG will have two designated terminal nodes, the old terminal node, labeled "1", and another terminal node labeled "0". From every node v in the revised SSG which does not carry full probability on its outedges, we direct all the "residual" probability to "0", i.e., we add an edge from v to "0" with probability $p_{v,\text{"0"}} = 1 - \sum_w p_{v,w}$, where the sum is over all remaining nodes w in the SSG. In the resulting finite SSG, we know that if the max player plays with an optimal memoryless strategy (which it has), and the min player plays arbitrarily with a memoryless strategy, there is 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. Note also that, importantly, the "expected time" that it takes for the probability to exit the system when max player plays optimally is finite (because there are no "null recurrent" nodes in a finite Markov chain).

Another way to put this fact is as follows: consider the resulting SSG to be a finite reward SSG with reward 1 on each transition, and switch the role of the max and min player, and now the goal of the max player is to maximize the total reward before termination (at either exit), and that of the min player is to minimize it. Translating the above to this setting, the "cleaned up" SSG has the property that the min player has a memoryless strategy using which, no matter what the maximizer does, the total reward will be finite: we will terminate, at "0" or at "1", in finite expected time (because there are no "null recurrent" nodes in finite Markov chains, and both players have optimal memoryless strategies).

Now, take the remaining finite SSG, call it G'. Just put a copy of G' at the entry of the component A_1 of the 1-RMC, identifying the entry en with the initial node, u, of G'. Take every edge that is directed into the terminal node "1" of G, and instead direct it to the exit ex of the component A_1. Next, take every edge that is directed into the terminal "0" node and direct it to the first call, (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 $q_u^* \leq 1/2$ in the finite SSG G' for terminating at the terminal "1" iff the value $r_u^* = \infty$ for expected reward value in the resulting reward 1-RSSG, A (recall: with the role min and max reversed, and with all transitions having reward 1).

The reason is as follows: we know that in A the minimizer has at least one SM strategy that obtains finite reward inside any copy of G', and it must play one such strategy each time it goes through G' if it wants to avoid payoff ∞.

Now, there are only a finite number of SM strategies for minimizer inside G' which yield a finite expected reward (after an optimal response by the maximizer). Let $D \in [0, \infty)$ be the maximum finite expected reward among those SM strategies. Also, no matter what the two players do, we know we will earn reward at least 1, each time we go through G'. So, each time going through G' we accumulate a reward $D' \in [1, D]$. So, from the point of view of trying to make sure the total expected reward is finite, it is really of no relevance what the specific value of D' is when we go through G'. Rather, what is important is whether we "visit" a copy of G', i.e., a copy of the entry u, infinitely often.

Now, to make sure that that the expected number of times u is visited is finite, the minimizer must in fact maximize the probability of terminating at "1", and thus minimize the probability of termination at "0". In addition, the minimizer must also make sure that the expected reward inside G' is finite, but this we know it can do while maximizing the probability of terminating at "1". Thus, the total reward $r_u^* = \infty$ precisely when the value of the SSG termination game G' is $\leq 1/2$.

□

A.11 Proof of Theorem 8

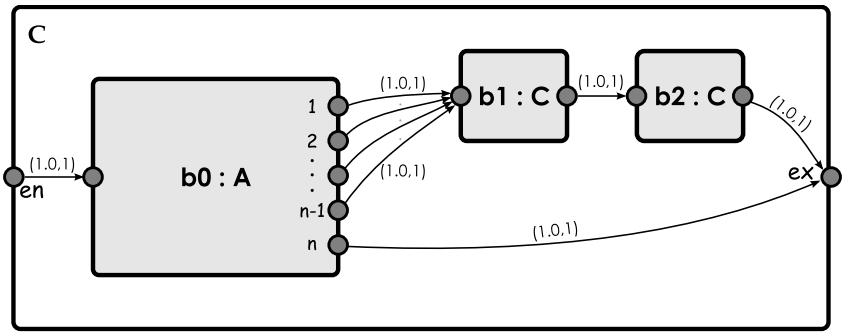

Fig. 3. Multi-exit reward RMDP: undecidability

Proof. We will use the construction of an component named A in the proof of Theorem 6 in [12]. This single-entry n-exit component relates RMDPs with n exits with Probabilistic Finite Automata (PFA) with n states. More precisely the supremum probability

of termination at the n-th exit starting at the entry of A is equal to the supremum probability with which the correlated PFA accepts some word. It was proved in [1] that deciding whether a given PFA with 46 states accepts any word with probability greater than $\frac{1}{2}$ is undecidable. This means it is undecidable to resolve whether the supremum probability of termination at the n-th exit ($n = 46$) in the correlated RMDP A is greater than $\frac{1}{2}$.

To prove that it is also undecidable to resolve whether the reward at a given node is finite or not, we will combine the RMDP A with a gadget 1-RMDP C, as can be seen at Fig. 3. Let us denote by p the supremum probability of termination at the n-th exit of the component A labeling box B. We will argue that $p > 1/2$ iff the infimum total reward for the reward 1-RMDP C is finite.

We will need the following observation about the component A. Namely, for any strategy that yields probability > 0 of exiting from the n-th exit of component A, it must be the case that the total probability of exiting from one of the exits of component A is 1. It is easy to verify this fact based on the structure of component A given in [12].

Now, first suppose $p > 1/2$. It follows from the previously mentioned fact that in the reward game the minimizer has a strategy with which to exit from A with probability 1, and simultaneously to exit from the n-th exit with probability $> 1/2$. Therefore, note that component C, under an optimal strategy played inside box B, acts like our favorite gadget in which the probability of exiting directly is $p > 1/2$. For this gadget, with $p > 1/2$ we know that the resulting expected time until termination is finite.

Moreover, the component A has the property that if $p > 1/2$, then the corresponding PFA accepts a word w with probability $> 1/2$, and we can use word w as a strategy σ_w in A such that starting at the entry of A, the strategy σ_w will exit A with probability 1, exit from the n-the exit with probability $p > 1/2$, and exit from A in finite expected time $2|w|$. Thus the expected time taken until termination inside A, i.e., inside the box B is finite, and hence the total expected time until termination starting at the entry of C is finite.

Next suppose that the infimum total reward is finite, but that $p \leq 1/2$. Then in C we either stay inside a copy of $B(A)$ with non-zero probability, in which case the total reward is infinite, or else we exit from the n-th exit with probability $p \leq 1/2$ and we exit from the other exits with probability $\geq 1/2$. It follows easily from the properties of the gadget in C that the expected termination time is infinite in such a case. Thus if we can decide whether the optimal reward at the entry of C is finite or not, we can also decide whether the termination probability at the n-th exit of B is greater than $\frac{1}{2}$, which we know is undecidable. □