STOCHASTIC PARITY GAMES ON LOSSY CHANNEL SYSTEMS

PAROSH AZIZ ABDULLA, LORENZO CLEMENTE, RICHARD MAYR, AND SVEN SANDBERG

Uppsala University, Department of Information Technology, Box 337, SE-751 05 Uppsala, Sweden

Département d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgium

University of Edinburgh, School of Informatics, 10 Crichton Street, Edinburgh EH89AB, UK

Uppsala University, Department of Information Technology, Box 337, SE-751 05 Uppsala, Sweden

ABSTRACT. We give an algorithm for solving stochastic parity games with almost-sure winning conditions on *lossy channel systems*, under the constraint that both players are restricted to finite-memory strategies. First, we describe a general framework, where we consider the class of $2\frac{1}{2}$-player games with almost-sure parity winning conditions on possibly infinite game graphs, assuming that the game contains a *finite attractor*. An attractor is a set of states (not necessarily absorbing) that is almost surely re-visited regardless of the players' decisions. We present a scheme that characterizes the set of winning states for each player. Then, we instantiate this scheme to obtain an algorithm for *stochastic game lossy channel systems*.

1. INTRODUCTION

Background. 2-player games can be used to model the interaction of a controller (player 0) who makes choices in a reactive system, and a malicious adversary (player 1) who represents an attacker. To model randomness in the system (e.g., unreliability; randomized algorithms), a third player 'random' is defined who makes choices according to a predefined probability distribution. The resulting stochastic game is called a $2\frac{1}{2}$-player game in the terminology of [CJH03]. The choices of the players induce a run of the system, and the winning conditions of the game are expressed in terms of predicates on runs.

Most classic work on algorithms for stochastic games has focused on finite-state systems (e.g., [Sha53, Con92, dAHK98, CJH03]), but more recently several classes of infinite-state systems have been considered as well. Stochastic games on infinite-state probabilistic recursive systems (i.e., probabilistic pushdown automata with unbounded stacks) were studied in [EY05, EY08, EWY08]. A different (and incomparable) class of infinite-state systems are channel systems, which use unbounded communication buffers instead of unbounded recursion.

Channel Systems consist of nondeterministic finite-state machines that communicate by asynchronous message passing via unbounded FIFO communication channels. They are also known

as communicating finite-state machines (CFSM) [BZ83]. Channel Systems are a very expressive model that can encode the behavior of Turing machines, by storing the content of an unbounded tape in a channel [BZ83]. Therefore, all verification questions are undecidable on Channel Systems.

A *Lossy Channel System (LCS)* [AJ93, Fin94] consists of finite-state machines that communicate by asynchronous message passing via unbounded *unreliable* (i.e., lossy) FIFO communication channels, i.e., messages can spontaneously disappear from channels. The original motivation for LCS is to capture the behavior of communication protocols which are designed to operate correctly even if the communication medium is unreliable (i.e., if messages can be lost). Additionally (and quite unexpectedly at the time), the lossiness assumption makes safety/reachability and termination decidable [AJ93, Fin94], albeit of non-primitive recursive complexity [Sch02]. However, other important verification problems are still undecidable for LCS, e.g., recurrent reachability (i.e., Büchi properties), boundedness, and behavioural equivalences [AJ96, Sch01, May03].

A *Probabilistic Lossy Channel System (PLCS)* [BS03, AR03] is a probabilistic variant of LCS where, in each computation step, each message can be lost independently with a given probability. This solves two limitations of LCS. First, from a modelling viewpoint, probabilistic losses are more realistic than the overly pessimistic setting of LCS where all messages can always be lost at any time. Second, in PLCS almost-sure recurrent reachability properties become decidable (unlike for LCS) [BS03, AR03]. Several algorithms for symbolic model checking of PLCS have been presented [ABRS05, Rab03]. The only reason why certain questions are decidable for LCS/PLCS is that the message loss induces a quasi-order on the configurations, which has the properties of a simulation. Similarly to Turing machines and CFSM, one can encode many classes of infinite-state probabilistic transition systems into a PLCS. Some examples are:

- Queuing systems where waiting customers in a queue drop out with a certain probability in every time interval. This is similar to the well-studied class of queuing systems with impatient customers which practice *reneging*, i.e., drop out of a queue after a given maximal waiting time; see [WLJ10] section II.B. Like in some works cited in [WLJ10], the maximal waiting time in our model is exponentially distributed. In basic PLCS, unlike in [WLJ10], this exponential distribution does not depend on the current number of waiting customers. However, an extension of PLCS with this feature would still be analyzable in our framework (except in the pathological case where a high number of waiting customers increases the customers patience exponentially, because such a system would not necessarily have a so-called *finite attractor*; see below).
- Probabilistic resource trading games with probabilistically fluctuating prices. The given stores of resources are encoded by counters (i.e., channels), which exhibit a probabilistic decline (due to storage costs, decay, corrosion, obsolescence, etc).
- Systems modelling operation cost/reward, which is stored in counters/channels, but probabilistically discounted/decaying over time.
- Systems which are periodically restarted (though not necessarily by a deterministic schedule), due to, e.g., energy depletion or maintenance work.

Due to this wide applicability of PLCS, we focus on this model in this paper. However, our main results are formulated in more general terms referring to infinite Markov chains with a finite attractor; see below.

Previous work. In [BBS07], a non-deterministic extension of PLCS was introduced where one player controls transitions in the control graph and message losses are fully probabilistic. This yields a Markov decision process (i.e., a $1\frac{1}{2}$-player game) on the infinite graphs induced by PLCS. It was

shown in [BBS07] that $1\frac{1}{2}$-player games with *almost-sure* repeated reachability (Büchi) objectives are decidable and pure memoryless determined.

In [AHdA$^+$08], $2\frac{1}{2}$-player games on PLCS are considered, where the players control transitions in the control graph and message losses are probabilistic. Almost-sure Büchi objectives are decidable for this class, and pure memoryless strategies suffice for *both players* [AHdA$^+$08]. Generalized Büchi objectives are also decidable, and finite-memory strategies suffice for the player, while memoryless strategies suffice for the opponent [BS13].

On the other hand, $1\frac{1}{2}$-player games on PLCS with *positive probability* Büchi objectives, i.e., almost-sure co-Büchi objectives from the (here passive) opponent's point of view, can require infinite memory to win and are also undecidable [BBS07]. However, if the player is restricted to finite-memory strategies, $1\frac{1}{2}$-player games with positive probability *parity objectives* (even the more general *Streett objectives*) become decidable and memoryless strategies suffice for the player [BBS07]. Note that the finite-memory case and the infinite-memory one are a priori incomparable problems, and neither subsumes the other. Cf. Section 6.

Non-stochastic (2-player) parity games on infinite graphs were studied in [Zie98], where it is shown that such games are determined, and that both players possess winning memoryless strategies in their respective winning sets. Furthermore, a scheme for computing the winning sets and winning strategies is given. Stochastic games ($2\frac{1}{2}$-player games) with parity conditions on *finite* graphs are known to be memoryless determined and effectively solvable [dAH00, CJH03, CdAH06].

Our contribution. We give an algorithm to decide almost-sure *parity* games for probabilistic lossy channel systems in the case where the players are restricted to finite memory strategies. We do that in two steps. First, we give our result in general terms (Section 4): We consider the class of $2\frac{1}{2}$-player games with almost-sure parity wining conditions on possibly infinite game graphs, under the assumption that the game contains a *finite attractor*. An attractor is a set A of states such that, regardless of the strategies used by the players, the probability measure of the runs which visit A infinitely often is one.[1] Note that this means neither that A is absorbing, nor that every run must visit A. We present a general scheme characterizing the set of winning states for each player. The scheme is a generalization of the well-known scheme for non-stochastic games in [Zie98]. In fact, the constructions are equivalent in the case that no probabilistic states are present. We show correctness of the scheme for games where each player is restricted to a finite-memory strategy. The correctness proof here is more involved than in the non-stochastic case of [Zie98]; we rely on the existence of a finite attractor and the restriction of the players to use finite-memory strategies. Furthermore, we show that if a player is winning against all finite-memory strategies of the other player then he can win using a *memoryless* strategy.

In the second step (Section 5), we show that the scheme can be instantiated for lossy channel systems. The above two steps yield an algorithm to decide parity games in the case when the players are restricted to finite memory strategies. If the players are allowed infinite memory, then the problem is undecidable already for $1\frac{1}{2}$-player games with co-Büchi objectives (a special case of 2-color parity objectives) [BBS07]. Note that even if the players are restricted to finite memory strategies, such a strategy (even a memoryless one) on an infinite game graph is still an infinite object. Thus, unlike for finite game graphs, one cannot solve a game by just guessing strategies and then checking if they are winning. Instead, we show how to effectively compute a finite, symbolic representation of the (possibly infinite) set of winning states for each player as a regular language (Section 5.2), and a finite description of winning strategies (Section 5.3).

---

[1] In the game community (e.g., [Zie98]) the word *attractor* is used to denote what we call a *force set* in Section 3. In the infinite-state systems community (e.g., [ABRS05, AHM07]), the word is used in the same way as we use it in this paper.

2. PRELIMINARIES

Notation. Let $\mathbb{O}$ and $\mathbb{N}$ denote the set of ordinal resp. natural numbers. With α, β, and γ we denote arbitrary ordinals, while with λ we denote limit ordinals. We use $f : X \to Y$ to denote that f is a total function from X to Y, and use $f : X \rightharpoonup Y$ to denote that f is a partial function from X to Y. We write $f(x) = \bot$ to denote that f is undefined on x, and define $dom(f) := \{x : f(x) \neq \bot\}$. We say that f is an *extension* of g if $g(x) = f(x)$ whenever $g(x) \neq \bot$. For $X' \subseteq X$, we use $f|X'$ to denote the restriction of f to X'. We will sometimes need to pick an arbitrary element from a set. To simplify the exposition, we let $select(X)$ denote an arbitrary but fixed element of the nonempty set X.

A *probability distribution* on a countable set X is a function $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. For a set X, we use X^* and X^ω to denote the sets of finite and infinite words over X, respectively. The empty word is denoted by ε.

Games. A *game* (of *rank n*) is a tuple $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \texttt{Col})$ defined as follows. S is a set of *states*, partitioned into the pairwise disjoint sets of *random states* S^R, states S^0 of Player 0, and states S^1 of Player 1. $\longrightarrow \subseteq S \times S$ is the *transition relation*. We write $s \longrightarrow s'$ to denote that $(s, s') \in \longrightarrow$. We assume that for each s there is at least one and at most countably many s' with $s \longrightarrow s'$. The *probability function* $P : S^R \times S \to [0, 1]$ satisfies both $\forall s \in S^R . \forall s' \in S . (P(s, s') > 0 \iff s \longrightarrow s')$ and $\forall s \in S^R . \sum_{s' \in S} P(s, s') = 1$. (The sum is well-defined since we assumed that the number of successors of any state is at most countable.) The *coloring function* is defined as $\texttt{Col} : S \to \{0, \ldots, n\}$, where $\texttt{Col}(s)$ is called the *color* of state s.

Let $Q \subseteq S$ be a set of states. We use $\stackrel{\mathcal{G}}{\neg} Q := S - Q$ to denote the *complement* of Q. Define $[Q]^0 := Q \cap S^0$, $[Q]^1 := Q \cap S^1$, $[Q]^{0,1} := [Q]^0 \cup [Q]^1$, and $[Q]^R := Q \cap S^R$. For $n \in \mathbb{N}$ and $\sim \in \{=, \leq\}$, let $[Q]^{\texttt{Col} \sim n} := \{s \in Q : \texttt{Col}(s) \sim n\}$ denote the sets of states in Q with color $\sim n$.

A *run* ρ in $\mathcal{G}$ is an infinite sequence $s_0 s_1 \cdots$ of states s.t. $s_i \longrightarrow s_{i+1}$ for all $i \geq 0$; $\rho(i)$ denotes s_i. A *path* π is a finite sequence $s_0 \cdots s_n$ of states s.t. $s_i \longrightarrow s_{i+1}$ for all $i : 0 \leq i < n$. We say that ρ (or π) *visits* s if $s = s_i$ for some i. For any $Q \subseteq S$, we use Π_Q to denote the set of paths that end in some state in Q. Intuitively, the choices of the players and the resolution of randomness induce a run $s_0 s_1 \cdots$, starting in some initial state $s_0 \in S$; state s_{i+1} is chosen as a successor of s_i, and this choice is made by Player 0 if $s_i \in S^0$, by Player 1 if $s_i \in S^1$, and it is chosen randomly according to the probability distribution $P(s_i, \cdot)$ if $s_i \in S^R$.

Strategies. For $x \in \{0, 1\}$, a strategy for Player x prescribes the next move, given the current prefix of the run. Formally, a *strategy* of Player x is a partial function $f^x : \Pi_{S^x} \rightharpoonup S$ s.t. $s_n \longrightarrow f^x(s_0 \cdots s_n)$ if $f^x(s_0 \cdots s_n)$ is defined. The strategy f^x prescribes for Player x the next move, given the current prefix of the run. A run $\rho = s_0 s_1 \cdots$ is said to be *consistent* with a strategy f^x of Player x if $s_{i+1} = f^x(s_0 s_1 \cdots s_i)$ whenever $f^x(s_0 s_1 \cdots s_i) \neq \bot$. We say that ρ is *induced* by (s, f^x, f^{1-x}) if $s_0 = s$ and ρ is consistent with both f^x and f^{1-x}. We use $Runs(\mathcal{G}, s, f^x, f^{1-x})$ to denote the set of runs in $\mathcal{G}$ induced by (s, f^x, f^{1-x}). We say that f^x is total if it is defined for every $\pi \in \Pi_{S^x}$.

A strategy f^x of Player x is *memoryless* if the next state only depends on the current state and not on the previous history of the run, i.e., for any path $s_0 \cdots s_n \in \Pi_{S^x}$, we have $f^x(s_0 \cdots s_n) = f^x(s_n)$.

A *finite-memory strategy* updates a finite memory each time a transition is taken, and the next state depends only on the current state and memory. Formally, we define a *memory structure* for Player x as a quadruple $\mathcal{M} = (M, m_0, \tau, \mu)$ satisfying the following properties. The nonempty set M is called the *memory* and $m_0 \in M$ is the *initial memory configuration*. For a current memory configuration m and a current state s, the next state is given by $\tau : S^x \times M \to S$, where $s \longrightarrow \tau(s, m)$. The next memory configuration is given by $\mu : S \times M \to M$. We extend μ to paths by $\mu(\varepsilon, m) = m$ and $\mu(s_0 \cdots s_n, m) = \mu(s_n, \mu(s_0 \cdots s_{n-1}, m))$. The total strategy $strat_{\mathcal{M}} : \Pi_{S^x} \to S$ induced by $\mathcal{M}$ is given by $strat_{\mathcal{M}}(s_0 \cdots s_n) := \tau(s_n, \mu(s_0 \cdots s_{n-1}, m_0))$. A total strategy f^x is said to have *finite memory*

if there is a memory structure $\mathcal{M} = (M, m_0, \tau, \mu)$ where M is finite and $f^x = strat_{\mathcal{M}}$. Consider a run $\rho = s_0 s_1 \cdots \in Runs(\mathcal{G}, s, f^x, f^{1-x})$ where f^{1-x} is induced by $\mathcal{M}$. We say that ρ *visits* the configuration (s, m) if there is an i such that $s_i = s$ and $\mu(s_0 s_1 \cdots s_{i-1}, m_0) = m$.

We use $F_{all}^x(\mathcal{G})$, $F_{finite}^x(\mathcal{G})$, and $F_{\emptyset}^x(\mathcal{G})$ to denote the set of *all*, *finite-memory*, and *memoryless* strategies respectively of Player x in $\mathcal{G}$. Note that memoryless strategies and strategies in general can be partial, whereas for simplicity we only define total finite-memory strategies.

Probability Measures. We use the standard definition of probability measures for a set of runs [Bil86]. First, we define the measure for total strategies, and then we extend it to general (partial) strategies. Consider a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \texttt{Col})$, an initial state s, and total strategies f^x and f^{1-x} of Players x and $1-x$. Let $\Omega^s = sS^{\omega}$ denote the set of all infinite sequences of states starting from s. For a measurable set $\mathfrak{R} \subseteq \Omega^s$, we define $\mathcal{P}_{\mathcal{G}, s, f^x, f^{1-x}}(\mathfrak{R})$ to be the probability measure of $\mathfrak{R}$ under the strategies f^x, f^{1-x}. This measure is well-defined [Bil86]. For (partial) strategies f^x and f^{1-x} of Players x and $1-x$, $\sim \in \{<, \leq, =, \geq, >\}$, a real number $c \in [0,1]$, and any measurable set $\mathfrak{R} \subseteq \Omega^s$, we define $\mathcal{P}_{\mathcal{G}, s, f^x, f^{1-x}}(\mathfrak{R}) \sim c$ iff $\mathcal{P}_{\mathcal{G}, s, g^x, g^{1-x}}(\mathfrak{R}) \sim c$ for all total strategies g^x and g^{1-x} that are extensions of f^x resp. f^{1-x}.

Winning Conditions. The winner of the game is determined by a predicate on infinite runs. We assume familiarity with the syntax and semantics of the temporal logic CTL^* (see, e.g., [CGP99]). Formulas are interpreted on the structure $(S, \longrightarrow)$. We use $[\![\varphi]\!]^s$ to denote the set of runs starting from s that satisfy the CTL^* path-formula φ. This set is measurable [Var85], and we just write $\mathcal{P}_{\mathcal{G}, s, f^x, f^{1-x}}(\varphi) \sim c$ instead of $\mathcal{P}_{\mathcal{G}, s, f^x, f^{1-x}}([\![\varphi]\!]^s) \sim c$.

We will consider games with *parity* winning conditions, whereby Player 1 wins if the largest color that occurs infinitely often in the infinite run is odd, and Player 0 wins if it is even. Thus, the winning condition for Player x can be expressed in CTL^* as

$$x\text{-}Parity := \bigvee_{i \in \{0, \ldots, n\} \wedge (i \bmod 2) = x} \left(\Box \Diamond [S]^{\texttt{Col}=i} \wedge \Diamond \Box [S]^{\texttt{Col} \leq i} \right) .$$

Winning Sets. For a strategy f^x of Player x, and a set F^{1-x} of strategies of Player $1-x$, we define

$$W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{\sim c}) := \{s : \forall f^{1-x} \in F^{1-x}. f^{1-x} \text{ is total} \implies \mathcal{P}_{\mathcal{G}, s, f^x, f^{1-x}}(\varphi) \sim c\}$$

If there is a strategy f^x such that $s \in W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{\sim c})$, then we say that s is a *winning state* for Player x in $\mathcal{G}$ wrt. $\varphi^{\sim c}$ (and f^x is *winning at s*), provided that Player $1-x$ is restricted to strategies in F^{1-x}. Sometimes, when the parameters $\mathcal{G}$, s, F^{1-x}, φ, and $\sim c$ are known, we will not mention them and may simply say that "s is a winning state" or that "f^x is a winning strategy", etc. If $s \in W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{=1})$, then we say that Player x wins from s *almost surely (a.s.)*. If $s \in W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{>0})$, then we say that Player x wins from s *with positive probability (w.p.p.)*.

We also define $V^x(f^x, F^{1-x})(\mathcal{G}, \varphi) := \{s : \forall f^{1-x} \in F^{1-x}. Runs(\mathcal{G}, s, f^x, f^{1-x}) \subseteq [\![\varphi]\!]^s\}$. If $s \in V^x(f^x, F^{1-x})(\mathcal{G}, \varphi)$, then we say that Player x *surely* wins from s. Notice that any strategy that is surely winning from a state s is also winning from s a.s., and any strategy that is winning a.s. is also winning w.p.p., i.e., $V^x(f^x, F^{1-x})(\mathcal{G}, \varphi) \subseteq W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{=1}) \subseteq W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{>0})$.

Determinacy and Solvability. A game is called *determined* wrt. an objective $\varphi^{\sim c}$ and two sets F^0, F^1 of strategies of Player 0, resp. Player 1, if, for every state s, Player x has a strategy $f^x \in F^x$ that is winning against all strategies $g \in F^{1-x}$ of the opponent, i.e., $s \in W^x(f^x, F^{1-x})(\mathcal{G}, \text{cond}_x)$, where $\text{cond}_0 = \varphi^{\sim c}$ and $\text{cond}_1 = \varphi^{\not\sim c}$. By *solving* a determined game, we mean giving an algorithm to compute symbolic representations of the sets of states which are winning for either player and a symbolic representation of the corresponding winning strategies.

Attractors. A set $A \subseteq S$ is said to be an *attractor* if, for each state $s \in S$ and strategies f^0, f^1 of Player 0 resp. Player 1, it is the case that $\mathcal{P}_{G,s,f^0,f^1}(\Diamond A) = 1$. In other words, regardless of where we start a run and regardless of the strategies used by the players, we will reach a state inside the attractor a.s.. It is straightforward to see that this also implies that $\mathcal{P}_{G,s,f^0,f^1}(\Box \Diamond A) = 1$, i.e., the attractor will be visited infinitely often a.s.

Transition Systems. Consider strategies $f^x \in F_\emptyset^x$ and $f^{1-x} \in F_{finite}^{1-x}$ of Player x resp. Player $1 - x$, where f^x is memoryless and f^{1-x} is finite-memory. Suppose that f^{1-x} is induced by memory structure $\mathcal{M} = (M, m_0, \tau, \mu)$. We define the *transition system* $\mathcal{T}$ induced by G, f^{1-x}, f^x to be the pair $(S_M, \leadsto)$ where $S_M = S \times M$, and $\leadsto \subseteq S_M \times S_M$ such that $(s_1, m_1) \leadsto (s_2, m_2)$ if $m_2 = \mu(s_1, m_1)$, and one of the following three conditions is satisfied: (i) $s_1 \in S^x$ and either $s_2 = f^x(s_1)$ or $f^x(s_1) = \bot$, (ii) $s_1 \in S^{1-x}$ and $s_2 = \tau(s_1, m_1)$, or (iii) $s_1 \in S^R$ and $P(s_1, s_2) > 0$.

Consider the directed acyclic graph (DAG) of maximal strongly connected components (SCCs) of the transition system $\mathcal{T}$. An SCC is called a *bottom SCC (BSCC)* if no other SCC is reachable from it. Observe that the existence of BSCCs is not guaranteed in an infinite transition system. However, if G contains a finite attractor A and M is finite then $\mathcal{T}$ contains at least one BSCC, and in fact each BSCC contains at least one element (s_A, m) with $s_A \in A$. In particular, for any state $s \in S$, any run $\rho \in Runs(G, s, f^x, f^{1-x})$ will visit a configuration (s_A, m) infinitely often a.s. where $s_A \in A$ and $(s_A, m) \in B$ for some BSCC B.

3. REACHABILITY

In this section we present some concepts related to checking reachability objectives in games. First, we define basic notions. Then we recall a standard scheme (described e.g. in [Zie98]) for checking reachability winning conditions, and state some of its properties that we use in the later sections. In this section, we do not use the finite attractor property, nor do we restrict the class of strategies in any way. Below, fix a game $G = (S, S^0, S^1, S^R, \longrightarrow, P, \texttt{Col})$.

Reachability Properties. Fix a state $s \in S$ and sets of states $Q, Q' \subseteq S$. Let $Post_G(s) := \{s' : s \longrightarrow s'\}$ denote the set of *successors* of s. Extend it to sets of states by $Post_G(Q) := \bigcup_{s \in Q} Post_G(s)$. Note that for any given state $s \in S^R$, $P(s, \cdot)$ is a probability distribution over $Post_G(s)$. Let $Pre_G(s) := \{s' : s' \longrightarrow s\}$ denote the set of *predecessors* of s, and extend it to sets of states as above. We define $\widetilde{Pre_G}(Q) := \overset{G}{\neg} Pre_G(\overset{G}{\neg} Q)$, i.e., it denotes the set of states whose successors *all* belong to Q. We say that Q is *sink-free* if $Post_G(s) \cap Q \neq \emptyset$ for all $s \in Q$, and *closable* if it is sink-free and $Post_G(s) \subseteq Q$ for all $s \in [Q]^R$. If Q is closable then each state in $[Q]^{0,1}$ has at least one successor in Q, and all the successors of states in $[Q]^R$ are in Q.

For $x \in \{0, 1\}$, we say that Q is an *x-trap* if it is closable and $Post_G(s) \subseteq Q$ for all $s \in [Q]^x$. Notice that S is both a 0-trap and a 1-trap, and in particular it is both sink-free and closable. The following lemma states that, starting from a state inside a set of states Q that is a trap for one player, the other player can surely keep the run inside Q.

Lemma 3.1. *If Q is a $(1-x)$-trap, then there exists a memoryless strategy $f^x \in F_\emptyset^x(G)$ for Player x such that $Q \subseteq V^x(f^x, F_{all}^{1-x}(G))(G, \Box Q)$.*

Proof. We define a memoryless strategy f^x of Player x that is surely winning from any state $s \in Q$, i.e., $Q \subseteq V^x(f^x, F_{all}^{1-x}(G))(G, \Box Q)$. For a state $s \in [Q]^x$, we define $f^x(s) = select(Post_G(s) \cap Q)$. This is well-defined since Q is a $(1-x)$-trap. We can now show that any run that starts from a state $s \in Q$ and that is consistent with f^x will surely remain inside Q. Let f^{1-x} be any strategy of

Player $1-x$, and let $s_0s_1\ldots \in Runs(\mathcal{G},s,f^x,f^{1-x})$. We show, by induction on i, that $s_i \in Q$ for all $i \geq 0$. The base case is clear since $s_0 = s \in Q$. For $i > 1$, we consider three cases depending on s_i:

- $s_i \in [S]^x$. By the induction hypothesis we know that $s_i \in Q$, and hence by definition of f^x we know that $s_{i+1} = f^x(s_i) \in Q$.
- $s_i \in [S]^{1-x}$. By the induction hypothesis we know that $s_i \in Q$, and hence $s_{i+1} \in Q$ since Q is a $(1-x)$-trap.
- $s_i \in [S]^R$. By the induction hypothesis we know that $s_i \in Q$, and hence $s_{i+1} \in Q$ since Q is closable. $\qquad\square$

Scheme. Given a set $\texttt{Target} \subseteq S$, we give a scheme for computing a partitioning of S into two sets $Force^x(\mathcal{G},\texttt{Target})$ and $Avoid^{1-x}(\mathcal{G},\texttt{Target})$ s.t. 1) Player x has a memoryless strategy on $Force^x(\mathcal{G},\texttt{Target})$ to force the game to $\texttt{Target}$ w.p.p., and 2) Player $1-x$ has a memoryless strategy on $Avoid^{1-x}(\mathcal{G},\texttt{Target})$ to surely avoid $\texttt{Target}$. The scheme and its correctness is adapted from [Zie98] to the stochastic setting.

First, we characterize the states that are winning for Player x, by defining an increasing set of states each of which consists of winning states for Player x, as follows:

$$\mathcal{R}_0 := \texttt{Target}$$

$$\mathcal{R}_{\alpha+1} := \mathcal{R}_\alpha \cup [Pre_\mathcal{G}(\mathcal{R}_\alpha)]^R \cup [Pre_\mathcal{G}(\mathcal{R}_\alpha)]^x \cup [\widetilde{Pre_\mathcal{G}}(\mathcal{R}_\alpha)]^{1-x}$$

$$\mathcal{R}_\lambda := \bigcup_{\alpha < \lambda} \mathcal{R}_\alpha \qquad \text{(for } \lambda \text{ a limit ordinal)}$$

Clearly, the sequence is non-decreasing, i.e., $\mathcal{R}_\alpha \subseteq \mathcal{R}_\beta$ when $\alpha \leq \beta$, and since the sequence is bounded by S, it converges at some (possibly infinite) ordinal. We state this as a lemma:

Lemma 3.2. *There is a* $\gamma \in \mathbb{O}$ *such that* $\mathcal{R}_\gamma = \bigcup_{\alpha \in \mathbb{O}} \mathcal{R}_\alpha$.

Let γ be the smallest ordinal s.t. $\mathcal{R}_\gamma = \mathcal{R}_{\gamma+1}$ (it exists by the lemma above). We define

$$Force^x(\mathcal{G},\texttt{Target}) := \mathcal{R}_\gamma$$

$$Avoid^{1-x}(\mathcal{G},\texttt{Target}) := {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$$

Lemma 3.3. $Avoid^{1-x}(\mathcal{G},\texttt{Target})$ *is an x-trap.*

Proof. Recall that $Avoid^{1-x}(\mathcal{G},\texttt{Target}) = {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ and $\mathcal{R}_{\gamma+1} \subseteq \mathcal{R}_\gamma$. First, we prove that ${}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ is sink-free. There are two cases to consider:

- $s \in [{}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma]^x \cup [{}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma]^R$. First, $Post_\mathcal{G}(s) \subseteq {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$. Indeed, if not, we would have $Post_\mathcal{G}(s) \cap \mathcal{R}_\gamma \neq \emptyset$, and thus $s \in \mathcal{R}_{\gamma+1} \subseteq \mathcal{R}_\gamma$, which is a contradiction. Second, since S is sink-free, we have $Post_\mathcal{G}(s) \neq \emptyset$, and thus $Post_\mathcal{G}(s) \cap {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma \neq \emptyset$.
- $s \in [{}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma]^{1-x}$. We clearly have $Post_\mathcal{G}(s) \cap {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma \neq \emptyset$, otherwise $Post_\mathcal{G}(s) \subseteq \mathcal{R}_\gamma$, and thus $s \in \mathcal{R}_{\gamma+1} \subseteq \mathcal{R}_\gamma$, which is a contradiction.

Second, when proving sink-freeness above, we showed that $Post_\mathcal{G}(s) \subseteq {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ for any $s \in [{}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma]^R$ which means that ${}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ is closable. Finally, we also showed that $Post_\mathcal{G}(s) \subseteq {}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ for any $s \in [{}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma]^x$, which means that ${}^{\mathcal{G}}_{\neg} \mathcal{R}_\gamma$ is an x-trap, thus concluding the proof. $\qquad\square$

The following lemma shows correctness of the construction. In fact, it shows that a winning player also has a memoryless strategy which is winning against an arbitrary opponent.

Lemma 3.4. *There are memoryless strategies* $force^x(G, \mathtt{Target}) \in F_\emptyset^x(G)$ *for Player x and* $avoid^{1-x}(G, \mathtt{Target}) \in F_\emptyset^{1-x}(G)$ *for Player* $1-x$ *s.t.*

$$Force^x(G, \mathtt{Target}) \subseteq W^x(force^x(G, \mathtt{Target}), F_{all}^{1-x}(G))(G, \Diamond \mathtt{Target}^{>0})$$

$$Avoid^{1-x}(G, \mathtt{Target}) \subseteq V^{1-x}(avoid^{1-x}(G, \mathtt{Target}), F_{all}^x(G))(G, \Box(\overset{G}{\lnot} \mathtt{Target}))$$

Proof. Let $\mathcal{R} = Force^x(G, \mathtt{Target})$. To prove the first claim, we define a memoryless strategy f^x of Player x that is winning from $\mathcal{R}$. For any $s \in [\mathcal{R}]^x$, let α be the unique ordinal s.t. $s \in [\mathcal{R}_{\alpha+1} \setminus \mathcal{R}_\alpha]^x$. Then, we define $f^x(s) := select(Post_G(s) \cap \mathcal{R}_\alpha)$. We show that f^x forces the run to the target set $\mathtt{Target}$ w.p.p. against an arbitrary opponent. Fix a strategy f^{1-x} for Player $1-x$. We show that $\mathcal{P}_{G,s,f^x,f^{1-x}}(\Diamond \mathtt{Target}) > 0$ by transfinite induction. If $s \in \mathcal{R}_0$, then the claim follows trivially. If $s \in \mathcal{R}_{\alpha+1}$, then either $s \in \mathcal{R}_\alpha$ in which case the claim holds by the induction hypothesis, or $s \in \mathcal{R}_{\alpha+1} \setminus \mathcal{R}_\alpha$. In the latter case, there are three sub-cases:

- $s \in [\mathcal{R}_{\alpha+1} \setminus \mathcal{R}_\alpha]^x$. By definition of f^x, we know that $f^x(s) = s'$ for some $s' \in \mathcal{R}_\alpha$. By the induction hypothesis, $\mathcal{P}_{G,s',f^0,f^1}(\Diamond \mathtt{Target}) > 0$, and hence $\mathcal{P}_{G,s,f^0,f^1}(\Diamond \mathtt{Target}) > 0$.
- $s \in [\mathcal{R}_{\alpha+1} \setminus \mathcal{R}_\alpha]^{1-x}$. Let s' be the successor of s chosen by f^{1-x}. By definition of $\mathcal{R}_{\alpha+1}$, we know that $s' \in \mathcal{R}_\alpha$. Then, the proof follows as in the previous case.
- $s \in [\mathcal{R}_{\alpha+1} \setminus \mathcal{R}_\alpha]^R$. By definition of $\mathcal{R}_{\alpha+1}$, there is a $s' \in \mathcal{R}_\alpha$ such that $P(s,s') > 0$. By the induction hypothesis, $\mathcal{P}_{G,s,f^0,f^1}(\Diamond \mathtt{Target}) \geq \mathcal{P}_{G,s',f^0,f^1}(\Diamond \mathtt{Target}) \cdot P(s,s') > 0$.

Finally, if $s \in \mathcal{R}_\lambda$ for a limit ordinal λ, then $s \in \mathcal{R}_\alpha$ for some $\alpha < \lambda$, and the claim follows by the induction hypothesis.

From Lemma 3.3 and Lemma 3.1 it follows that there is a strategy f^{1-x} for Player $1-x$ such that $Avoid^{1-x}(G, \mathtt{Target}) \subseteq V^{1-x}(f^{1-x}, F_{all}^x)(G, \Box(Avoid^{1-x}(G, \mathtt{Target})))$. The second claim follows then from the fact that $\mathtt{Target} \cap Avoid^{1-x}(G, \mathtt{Target}) = \emptyset$. ◻

4. PARITY CONDITIONS

We describe a scheme for solving stochastic parity games with almost-sure winning conditions on infinite graphs, under the conditions that the game has a finite attractor (as defined in Section 2), and that the players are restricted to finite-memory strategies.

We define a sequence of functions $C_0, C_1, \ldots$ Each C_n takes a single argument, a game of rank at most n, and it returns the set of states where Player x wins a.s., with $x = n \bmod 2$. In other words, the player that has the same parity as color n wins a.s. in $C_n(G)$. We provide a memoryless strategy that is winning a.s. for Player x in $C_n(G)$ against any finite-memory strategy of Player $1-x$, and a memoryless strategy that is winning w.p.p. for Player $1-x$ in $\overset{G}{\lnot} C_n(G)$ against any finite-memory strategy of Player x.

The scheme is by induction on n and is related to [Zie98]. In the rest of the section, we make use of the following notion of sub-game. For a closable $\overset{G}{\lnot} Q$, we define the *sub-game* $G \ominus Q := (Q', [Q']^0, [Q']^1, [Q']^R, \longrightarrow', P', \mathtt{Col}')$, where $Q' := \overset{G}{\lnot} Q$ is the new set of states, $\longrightarrow' := \longrightarrow \cap (Q' \times Q')$, $P' := P|([Q']^R \times Q')$, $\mathtt{Col}' := \mathtt{Col}|Q'$. Notice that $P'(s)$ is a probability distribution for any $s \in [Q']^R$ since Q' is closable. We use $G \ominus Q_1 \ominus Q_2$ to denote $(G \ominus Q_1) \ominus Q_2$.

For the base case, let $C_0(G) := S$ for any game G of rank 0. Indeed, from any configuration Player 0 trivially wins a.s. (even surely) because there is only color 0.
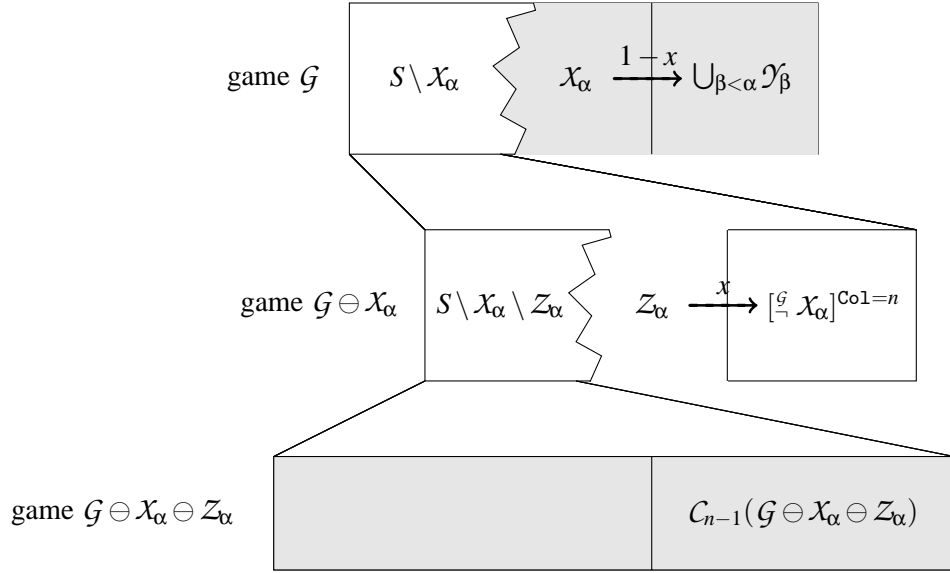
FIGURE 1. The construction of the various sets involved in the inductive step. The grey area is $\mathcal{Y}_\alpha$.

For $n \geq 1$, let $\mathcal{G}$ be a game of rank n. In the following, let

$$x = n \bmod 2.$$

$\mathcal{C}_n(\mathcal{G})$ is defined with the help of two auxiliary transfinite sequences of sets of states $\{\mathcal{X}_\alpha\}_{\alpha \in \mathbb{O}}$ and $\{\mathcal{Y}_\alpha\}_{\alpha \in \mathbb{O}}$. The construction ensures that $\mathcal{X}_0 \subseteq \mathcal{Y}_0 \subseteq \mathcal{X}_1 \subseteq \mathcal{Y}_1 \subseteq \cdots$, and that the states of $\mathcal{X}_\alpha, \mathcal{Y}_\alpha$ are winning w.p.p. for Player $1 - x$. We use strong induction, i.e., to construct $\mathcal{X}_\alpha$ we assume that $\mathcal{X}_\beta$ has been constructed for all $\beta < \alpha$, and it suffices to state one unified inductive step rather than distinguishing between base case, successor ordinals and non-zero limit ordinals. In the (unified) inductive step, we have already constructed $\mathcal{X}_\beta$ and $\mathcal{Y}_\beta$ for all $\beta < \alpha$. Our construction of $\mathcal{X}_\alpha$ and $\mathcal{Y}_\alpha$ is in three steps (cf. Figure 1):

(1) $\mathcal{X}_\alpha$ is the set of states where Player $1 - x$ can force the run to visit $\bigcup_{\beta < \alpha} \mathcal{Y}_\beta$ w.p.p.
(2) Find a set of states where Player $1 - x$ wins w.p.p. in the sub-game $\mathcal{G} \ominus \mathcal{X}_\alpha$.
(3) Take $\mathcal{Y}_\alpha$ to be the union of $\mathcal{X}_\alpha$ and the set constructed in step 2.

We next show how to find the winning states in the sub-game $\mathcal{G} \ominus \mathcal{X}_\alpha$ in step 2. We first compute the set of states where Player x can force the play in $\mathcal{G} \ominus \mathcal{X}_\alpha$ to reach a state with color n w.p.p. We call this set $\mathcal{Z}_\alpha$. The sub-game $\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$ does not contain any states of color n. Therefore, this game can be completely solved, using the already constructed function $\mathcal{C}_{n-1}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$. The resulting winning set is winning a.s. in $\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$, hence it is winning w.p.p. We will prove that the states where Player $1 - x$ wins w.p.p. in $\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$ are winning w.p.p. also in $\mathcal{G}$. We thus take $\mathcal{Y}_\alpha$ as the union of $\mathcal{X}_\alpha$ and $\mathcal{C}_{n-1}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$.

We define the sequences formally:

$$\mathcal{X}_\alpha := Force^{1-x}(\mathcal{G}, \bigcup_{\beta < \alpha} \mathcal{Y}_\beta)$$
$$\mathcal{Z}_\alpha := Force^x(\mathcal{G} \ominus \mathcal{X}_\alpha, [\stackrel{\mathcal{G}}{\neg} \mathcal{X}_\alpha]^{\mathtt{Col}=n})$$
$$\mathcal{Y}_\alpha := \mathcal{X}_\alpha \cup \mathcal{C}_{n-1}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$$

Notice that the sub-games $G \ominus X_\alpha$ and $G \ominus X_\alpha \ominus Z_\alpha$ are well-defined, since $\overset{G}{\neg} X_\alpha$ is closable in G (by Lemma 3.3), and $\overset{G \ominus X_\alpha}{\neg} Z_\alpha$ is closable in $G \ominus X_\alpha$.

By the definition, for $\alpha \leq \beta$ we get $\mathcal{Y}_\alpha \subseteq X_\beta \subseteq \mathcal{Y}_\beta$. As in Lemma 3.2, we can prove that this sequence converges:

Lemma 4.1. *There exists a $\gamma \in \mathbb{O}$ such that $X_\gamma = \mathcal{Y}_\gamma = \bigcup_{\alpha \in \mathbb{O}} \mathcal{Y}_\alpha$.*

Let γ be the least ordinal s.t. $X_{\gamma+1} = X_\gamma$ (which exists by the lemma above). We define

$$C_n(G) :=\overset{G}{\neg} X_\gamma \tag{4.1}$$

The following lemma shows the correctness of the construction. Recall that we assume that G is of rank n and that it contains a finite attractor.

Lemma 4.2. *There are memoryless strategies $f_c^x \in F_\emptyset^x(G)$ for Player x and $f_c^{1-x} \in F_\emptyset^{1-x}(G)$ for Player $1-x$ such that the following two properties hold:*

$$C_n(G) \subseteq W^x(f_c^x, F_{finite}^{1-x}(G))(G, x\text{-}Parity^{=1}) \tag{4.2}$$

$$\overset{G}{\neg} C_n(G) \subseteq W^{1-x}(f_c^{1-x}, F_{finite}^x(G))(G, (1-x)\text{-}Parity^{>0}) \tag{4.3}$$

Proof. Using induction on n, we define the strategies f_c^x, f_c^{1-x}, and prove that the strategies are indeed winning.

Construction of f_c^x. For $n \geq 1$, recall that γ is the least ordinal s.t. $X_{\gamma+1} = X_\gamma$ (as defined above), and define $\overline{X_\gamma} :=\overset{G}{\neg} X_\gamma$ and $\overline{Z_\gamma} :=\overset{G}{\neg} Z_\gamma$. By definition, $C_n(G) = \overline{X_\gamma}$. For a state $s \in \overline{X_\gamma}$, we define $f_c^x(s)$ depending on the membership of s in one of the following three partitions of $\overline{X_\gamma}$:

(1) $s \in \overline{X_\gamma} \cap \overline{Z_\gamma}$. Define $G' := G \ominus X_\gamma \ominus Z_\gamma$. By the definition of γ, we have that $X_{\gamma+1} \setminus X_\gamma = \emptyset$. By the construction of $\mathcal{Y}_\alpha$ we have, for an arbitrary α, that $C_{n-1}(G \ominus X_\alpha \ominus Z_\alpha) = \mathcal{Y}_\alpha \setminus X_\alpha$, and by the construction of $X_{\alpha+1}$, we have that $\mathcal{Y}_\alpha \setminus X_\alpha \subseteq X_{\alpha+1} \setminus X_\alpha$. By combining these facts, we obtain $C_{n-1}(G') \subseteq X_{\gamma+1} \setminus X_\gamma = \emptyset$. Since $G \ominus X_\gamma \ominus Z_\gamma$ does not contain any states of color n (or higher), it follows by the induction hypothesis that there is a memoryless strategy $f_1 \in F_\emptyset^x(G')$ such that $\overset{G'}{\neg} C_{n-1}(G') \subseteq W^x(f_1, F_{finite}^{1-x}(G'))(G', x\text{-}Parity^{>0})$. We define $f_c^x(s) := f_1(s)$. (Later, we will prove that in fact f_1 is winning a.s.)

(2) $s \in \overline{X_\gamma} \cap [Z_\gamma]^{\texttt{Col}<n}$. Define $f_c^x(s) := force^x(G \ominus X_\gamma, [Z_\gamma]^{\texttt{Col}=n})(s)$.

(3) $s \in \overline{X_\gamma} \cap [Z_\gamma]^{\texttt{Col}=n}$. By Lemma 3.3 we know that $Post_G(s) \cap \overline{X_\gamma} \neq \emptyset$. Define $f_c^x(s) := select(Post_G(s) \cap \overline{X_\gamma})$.

Correctness of f_c^x. Let $f^{1-x} \in F_{finite}^{1-x}(G)$ be a finite-memory strategy for Player $1-x$. We show that $\mathcal{P}_{G,s,f_c^x,f^{1-x}}(x\text{-}Parity) = 1$ for any state $s \in C_n(G)$.

First, we give a straightforward proof that any run $s_0 s_1 \cdots \in Runs(G, s, f_c^x, f^{1-x})$ will always stay inside $\overline{X_\gamma}$, i.e., $s_i \in \overline{X_\gamma}$ for all $i \geq 0$. We use induction on i. The base case follows from $s_0 = s \in \overline{X_\gamma}$. For the induction step, we assume that $s_i \in \overline{X_\gamma}$, and show that $s_{i+1} \in \overline{X_\gamma}$. We consider the following cases:

- $s_i \in [\overline{X_\gamma}]^{1-x} \cup [\overline{X_\gamma}]^R$. The result follows since $\overline{X_\gamma}$ is a $(1-x)$-trap in G (by Lemma 3.3).
- $s_i \in [\overline{X_\gamma} \cap \overline{Z_\gamma}]^x$. We know that $s_{i+1} = f_1(s_i)$. Since $f_1 \in F_\emptyset^x(G \ominus X_\gamma \ominus Z_\gamma)$ it follows that $s_{i+1} \in \overline{X_\gamma} \cap \overline{Z_\gamma}$, and in particular $s_{i+1} \in \overline{X_\gamma}$.
- $s_i \in [\overline{X_\gamma} \cap [Z_\gamma]^{\texttt{Col}<n}]^x$. We know that $s_{i+1} = force^x(G \ominus X_\gamma, [Z_\gamma]^{\texttt{Col}=n})(s_i)$. The result follows by the fact that $force^x(G \ominus X_\gamma, [Z_\gamma]^{\texttt{Col}=n})$ is a strategy in $G \ominus X_\gamma$.
- $s_i \in [\overline{X_\gamma} \cap [Z_\gamma]^{\texttt{Col}=n}]^x$. We have $s_{i+1} \in Post_G(s_i) \cap \overline{X_\gamma}$, and in particular $s_{i+1} \in \overline{X_\gamma}$.

We now prove the main claim. This is where we need the assumption of finite attractor and finite-memory strategies. Let us again consider a run $\rho \in Runs(\mathcal{G}, s, f_c^x, f^{1-x})$. We show that ρ is a.s. winning for Player x with respect to x-*Parity* in $\mathcal{G}$. Let f^{1-x} be induced by a memory structure $\mathcal{M} = (M, m_0, \tau, \mu)$. Let $\mathcal{T}$ be the transition system induced by $\mathcal{G}$, f_c^x, and f^{1-x}. As explained in Section 2, ρ will a.s. visit a configuration $(s_A, m) \in B$ for some BSCC B in $\mathcal{T}$. Since there exists a finite attractor, each state that occurs in B will a.s. be visited infinitely often by ρ. Let $n_{\max}$ be the maximal color occurring among the states of B. There are two possible cases:

- $n_{\max} = n$. Since each state in $\mathcal{G}$ has color at most n, Player x will a.s. win.
- $n_{\max} < n$. This implies that $\{s_B : (s_B, m) \in B\} \subseteq \overline{\mathcal{Z}_\gamma}$, and hence Player x uses the strategy f_1 to win the game in $\mathcal{G} \ominus \mathcal{X}_\gamma \ominus \mathcal{Z}_\gamma$ w.p.p.Then, either (i) $n_{\max} \bmod 2 = x$ in which case all states inside B are almost sure winning for Player x; or (ii) $n_{\max} \bmod 2 = 1 - x$ in which case all states inside B are almost sure losing for Player x. The result follows from the fact that case (ii) gives a contradiction since all states in $\mathcal{G} \ominus \mathcal{X}_\gamma \ominus \mathcal{Z}_\gamma$ (including those in B) are winning for Player x w.p.p.

Construction of f_c^{1-x}. We define a strategy f_c^{1-x} such that, for all α, the following inclusion holds: $\mathcal{X}_\alpha \subseteq \mathcal{Y}_\alpha \subseteq W^{1-x}(f_c^{1-x}, F_{finite}^x(\mathcal{G}))(\mathcal{G}, (1-x)\text{-}Parity^{>0})$. The result then follows from the definition of $C_n(\mathcal{G})$. The inclusion $\mathcal{X}_\alpha \subseteq \mathcal{Y}_\alpha$ holds by the definition of $\mathcal{Y}_\alpha$. For any state $s \in^{\mathcal{G}}_{\neg} C_n(\mathcal{G})$, we define $f_c^{1-x}(s)$ as follows. Let α be the smallest ordinal such that $s \in \mathcal{Y}_\alpha$. Such an α exists by the well-ordering of ordinals and since $^{\mathcal{G}}_{\neg} C_n(\mathcal{G}) = \bigcup_{\beta \in \mathbb{O}} \mathcal{X}_\beta = \bigcup_{\beta \in \mathbb{O}} \mathcal{Y}_\beta$. Now there are two cases:

- $s \in \mathcal{X}_\alpha \setminus \bigcup_{\beta < \alpha} \mathcal{Y}_\beta$. Define $f_c^{1-x}(s) := f_1(s) := force^{1-x}(\mathcal{G}, \bigcup_{\beta < \alpha} \mathcal{Y}_\beta)(s)$.
- $s \in C_{n-1}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$. By the induction hypothesis (on n), there is a memoryless strategy $f_2 \in F_0^{1-x}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$ of Player $1-x$ such that $s \in W^{1-x}(f_2, F_{finite}^x(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha))(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha, (1-x)\text{-}Parity^{=1})$. Define $f_c^{1-x}(s) := f_2(s)$.

Correctness of f_c^{1-x}. Let $f^x \in F_{finite}^x(\mathcal{G})$ be a finite-memory strategy for Player x. We now use induction on α to show that $\mathcal{P}_{\mathcal{G}, s, f_c^{1-x}, f^x}((1-x)\text{-}Parity) > 0$ for any state $s \in \mathcal{Y}_\alpha$. There are three cases:

(1) If $s \in \bigcup_{\beta < \alpha} \mathcal{Y}_\beta$, then $s \in \mathcal{Y}_\beta$ for some $\beta < \alpha$ and the result follows by the induction hypothesis on β.
(2) If $s \in \mathcal{X}_\alpha \setminus \bigcup_{\beta < \alpha} \mathcal{Y}_\beta$, then we know that Player $1-x$ can use f_1 to force the game w.p.p. to $\bigcup_{\beta < \alpha} \mathcal{Y}_\beta$ from which she wins w.p.p.
(3) If $s \in C_{n-1}(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$, then Player $1-x$ uses f_2. There are now two sub-cases: either (i) there is a run from s consistent with f^x and f_c^{1-x} that reaches $\mathcal{X}_\alpha$; or (ii) there is no such run.
  In sub-case (i), the run reaches $\mathcal{X}_\alpha$ w.p.p. Then, by cases 1 and 2, Player $1-x$ wins w.p.p.
  In sub-case (ii), all runs stay forever outside $\mathcal{X}_\alpha$. So the game is in effect played on $\mathcal{G} \ominus \mathcal{X}_\alpha$. Notice then that any run from s that is consistent with f^x and f_c^{1-x} stays forever in $\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$. The reason is that (by Lemma 3.3) $^{\mathcal{G} \ominus \mathcal{X}_\alpha}_{\neg} \mathcal{Z}_\alpha$ is an x-trap in $\mathcal{G} \ominus \mathcal{X}_\alpha$. Since all runs remain inside $\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$, Player $1-x$ wins w.p.p. (even a.s.) wrt. $(1-x)\text{-}Parity$ using f_2. $\square$

The following theorem follows immediately from the previous lemmas.

Theorem 4.3. *Stochastic parity games with almost sure winning conditions on infinite graphs are memoryless determined, provided there exists a finite attractor and the players are restricted to finite-memory strategies.*

Remark. We can compute both the a.s. winning set and the w.p.p. winning set for both players as follows. Let $n_{\max}$ be the maximal color occurring in the game. Then:

- Player x wins a.s. in $C_{n_{\max}}(\mathcal{G})$ and w.p.p. in $\stackrel{\mathcal{G}}{\dashv} C_{n_{\max}+1}(\mathcal{G})$;
- Player $1-x$ wins a.s. in $C_{n_{\max}+1}(\mathcal{G})$ and w.p.p. in $\stackrel{\mathcal{G}}{\dashv} C_{n_{\max}}(\mathcal{G})$.

5. APPLICATION TO LOSSY CHANNEL SYSTEMS

5.1. **Lossy channel systems.** A *lossy channel system (LCS)* is a finite-state machine equipped with a finite number of unbounded fifo channels (queues) [AJ93]. The system is *lossy* in the sense that, before and after a transition, an arbitrary number of messages may be lost from the channels. We consider *stochastic game-LCS (SG-LCS)*: each individual message is lost independently with probability λ in every step, where $\lambda > 0$ is a parameter of the system. The set of control states is partitioned into states belonging to Player 0 and 1. The player who owns the current control state chooses an enabled outgoing transition.

Formally, a SG-LCS of rank n is a tuple $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda, \mathtt{Col})$ where $\mathtt{S}$ is a finite set of *control states* partitioned into control states $\mathtt{S}^0, \mathtt{S}^1$ of Player 0 and 1; $\mathtt{C}$ is a finite set of *channels*, $\mathtt{M}$ is a finite set called the *message alphabet*, $\mathtt{T}$ is a set of *transitions*, $0 < \lambda < 1$ is the *loss rate*, and $\mathtt{Col} : \mathtt{S} \to \{0, \ldots, n\}$ is the *coloring* function. Each transition $\mathtt{t} \in \mathtt{T}$ is of the form $\mathtt{s} \stackrel{\mathtt{op}}{\longrightarrow} \mathtt{s}'$, where $\mathtt{s}, \mathtt{s}' \in \mathtt{S}$ and op is one of the following three forms: $\mathtt{c}!\mathtt{m}$ (send message $\mathtt{m} \in \mathtt{M}$ in channel $\mathtt{c} \in \mathtt{C}$), $\mathtt{c}?\mathtt{m}$ (receive message $\mathtt{m}$ from channel $\mathtt{c}$), or $\mathtt{nop}$ (do not modify the channels).

The SG-LCS $\mathcal{L}$ induces a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \mathtt{Col})$, where $S = \mathtt{S} \times (\mathtt{M}^*)^{\mathtt{C}} \times \{0, 1\}$. That is, each state in the game (also called a *configuration*) consists of a control state, a function that assigns a finite word over the message alphabet to each channel, and one of the symbols 0 or 1. States where the last symbol is 0 are random: $S^R = \mathtt{S} \times (\mathtt{M}^*)^{\mathtt{C}} \times \{0\}$. The other states belong to a player according to the control state: $S^x = \mathtt{S}^x \times (\mathtt{M}^*)^{\mathtt{C}} \times \{1\}$. Transitions out of states of the form $s = (\mathtt{s}, \mathtt{x}, 1)$ model transitions in $\mathtt{T}$ leaving control state $\mathtt{s}$. On the other hand, transitions leaving configurations of the form $s = (\mathtt{s}, \mathtt{x}, 0)$ model message losses. More precisely, transitions are defined as follows:

- If $s = (\mathtt{s}, \mathtt{x}, 1), s' = (\mathtt{s}', \mathtt{x}', 0) \in S$, then we have $s \longrightarrow s'$ iff $\mathtt{s} \stackrel{\mathtt{op}}{\longrightarrow} \mathtt{s}'$ is a transition in $\mathtt{T}$ and (i) if $\mathtt{op} = \mathtt{nop}$, then $\mathtt{x} = \mathtt{x}'$; (ii) if $\mathtt{op} = \mathtt{c}!\mathtt{m}$, then $\mathtt{x}\mathtt{c} = w$ and $\mathtt{x}' = \mathtt{x}[\mathtt{c} \mapsto w \cdot \mathtt{m}]$ (iii) if $\mathtt{op} = \mathtt{c}?\mathtt{m}$, then $\mathtt{x}\mathtt{c} = \mathtt{m} \cdot w$ and $\mathtt{x}' = \mathtt{x}[\mathtt{c} \mapsto w]$, where the notation $\mathtt{x}[\mathtt{c} \mapsto w]$ represents the channel assignment which is the same as $\mathtt{x}$ except that it maps $\mathtt{c}$ to the word $w \in M^*$.
- To model message losses, we introduce the subword ordering $\preceq$ on words: $x \preceq y$ iff x is a word obtained by removing zero or more messages from arbitrary positions of y. This is extended to channel contents $\mathtt{x}, \mathtt{x}' \in (\mathtt{M}^*)^{\mathtt{C}}$ by $\mathtt{x} \preceq \mathtt{x}'$ iff $\mathtt{x}(\mathtt{c}) \preceq \mathtt{x}'(\mathtt{c})$ for all channels $\mathtt{c} \in \mathtt{C}$, and to configurations $s = (\mathtt{s}, \mathtt{x}, i), s' = (\mathtt{s}', \mathtt{x}', i') \in S$ by $s \preceq s'$ iff $\mathtt{s} = \mathtt{s}', \mathtt{x} \preceq \mathtt{x}'$, and $i = i'$. For any $s = (\mathtt{s}, \mathtt{x}, 0)$ and any $\mathtt{x}' \preceq \mathtt{x}$, there is a transition $s \longrightarrow (\mathtt{s}, \mathtt{x}', 1)$. The probability of random transitions is given by $P((\mathtt{s}, \mathtt{x}, 0), (\mathtt{s}, \mathtt{x}', 1)) = a \cdot \lambda^{c-b} \cdot (1 - \lambda)^c$, where a is the number of ways to obtain $\mathtt{x}'$ by losing messages in $\mathtt{x}$, b is the total number of messages in all channels of $\mathtt{x}$, and c is the total number of messages in all channels of $\mathtt{x}'$ (see [ABRS05] for details).

Every configuration of the form $(\mathtt{s}, \mathtt{x}, 0)$ has at least one successor, namely $(\mathtt{s}, \mathtt{x}, 1)$. If a configuration $(\mathtt{s}, \mathtt{x}, 1)$ does not have successors according to the rules above, then we add a transition $(\mathtt{s}, \mathtt{x}, 1) \longrightarrow (\mathtt{s}, \mathtt{x}, 0)$, to ensure that the induced game is sink-free.

Finally, for a configuration $s = (\mathbf{s}, \mathbf{x}, i)$, we define $\mathtt{Col}(s) := \mathtt{Col}(\mathbf{s})$. Notice that the graph of the game is bipartite, in the sense that a configuration in S^R has only transitions to configurations in $[S]^{0,1}$, and vice versa.

We say that a set of channel contents $\mathtt{X} \subseteq (\mathtt{M}^*)^{\mathtt{C}}$ is *regular* if it is a finite union of sets of the form $\mathtt{Y} \subseteq (\mathtt{M}^*)^{\mathtt{C}}$ where $\mathtt{Y}(\mathtt{c})$ is a regular subset of $\mathtt{M}^*$ for every $c \in \mathtt{C}$ (this coincides with the notion of recognisable subset of $(\mathtt{M}^*)^{\mathtt{C}}$; cf. [Ber79]). We extend the notion of regularity to a set of configurations $P \subseteq S$ by saying that P is *regular* iff, for every control state $\mathbf{s} \in \mathtt{S}$ and $i \in \{0,1\}$, there exists a regular set of channel contents $\mathtt{X}_{\mathbf{s},\mathbf{i}} \subseteq (\mathtt{M}^*)^{\mathtt{C}}$ s.t. $P = \{(\mathbf{s}, \mathbf{x}, i) : \mathbf{s} \in \mathtt{S}, i \in \{0,1\}, \mathbf{x} \in \mathtt{X}_{\mathbf{s},\mathbf{i}}\}$.

In the qualitative *parity game problem* for SG-LCS, we want to characterize the sets of configurations where Player x can force the x-*Parity* condition to hold a.s., for both players.

5.2. **From scheme to algorithm.** We transform the scheme of Section 4 into an algorithm for deciding the a.s. parity game problem for SG-LCS. Consider an SG-LCS $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda, \mathtt{Col})$ and the induced game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \mathtt{Col})$ of some rank n. Furthermore, assume that the players are restricted to finite-memory strategies. We show the following.

Theorem 5.1. *The sets of winning configurations for Players 0 and 1 are effectively computable as regular sets of configurations. Furthermore, from each configuration, memoryless strategies suffice for the winning player.*

In the statement of the theorem, "effectively" means that a finite description of the regular sets of winning configurations is computable. We give the proof in several steps. First, we show that the game induced by an SG-LCS contains a finite attractor (Lemma 5.2). Then, we show that the scheme in Section 3 for computing winning configurations wrt. reachability objectives is guaranteed to terminate (Lemma 5.4). Furthermore, we show that the scheme in Section 4 for computing winning configurations wrt. a.s. parity objectives is guaranteed to terminate (Lemma 5.7). Notice that Lemmas 5.4 and 5.7 imply that for SG-LCS our transfinite constructions stabilize below ω (the first infinite ordinal). Finally, we show that each step in the above two schemes can be performed using standard operations on regular languages (Lemmas 5.11 and 5.12).

Finite attractor. In [ABRS05] it was shown that any Markov chain induced by a Probabilistic LCS contains a finite attractor. The proof can be carried over in a straightforward manner to the current setting. More precisely, the finite attractor is given by $A = (\mathtt{S} \times \boldsymbol{\varepsilon} \times \{0,1\})$ where $\boldsymbol{\varepsilon}(\mathtt{c}) = \varepsilon$ for each $c \in \mathtt{C}$. In other words, A is given by the set of configurations in which all channels are empty. The proof relies on the observation that if the number of messages in some channel is sufficiently large, it is more likely that the number of messages decreases than that it increases in the next step. This gives the following.

Lemma 5.2. $\mathcal{G}$ *contains a finite attractor.*

Termination of Reachability Scheme. For a set of configurations $Q \subseteq S$, we define the *upward closure* of Q by $Q{\uparrow} := \{s : \exists s' \in Q . s' \preceq s\}$. A set $U \subseteq Q \subseteq S$ is said to be *Q-upward-closed* (or *Q-u.c.* for short) if $(U{\uparrow}) \cap Q = U$. We say that U is *upward closed* if it is *S*-u.c.

Lemma 5.3. *If $Q_0 \subseteq Q_1 \subseteq \cdots$, and for all i it holds that $Q_i \subseteq Q$ and Q_i is Q-u.c., then there is an $j \in \mathbb{N}$ such that $Q_i = Q_j$ for all $i \geq j$.*

Proof. By Higman's lemma [Hig52], there is a $j \in \mathbb{N}$ s.t. $Q_i{\uparrow} = Q_j{\uparrow}$ for all $i \geq j$. Hence, $Q_i{\uparrow} \cap Q = Q_j{\uparrow} \cap Q$ for all $i \geq j$. Since all Q_i are Q-u.c., $Q_i{\uparrow} \cap Q = Q_i$ for all $i \geq j$. So $Q_i = Q_j$ for all $i \geq j$. $\square$

Now, we can show termination of the reachability scheme.

Lemma 5.4. *There exists a finite $j \in \mathbb{N}$ such that $\mathcal{R}_i = \mathcal{R}_j$ for all $i \geq j$.*

Proof. First, we show that $[\mathcal{R}_i \setminus \mathtt{Target}]^R$ is $(\overset{\mathcal{G}}{\rightsquigarrow} \mathtt{Target})$-u.c. for all $i \in \mathbb{N}$. We use induction on i. For $i = 0$ the result is trivial since $\mathcal{R}_i \setminus \mathtt{Target} = \emptyset$. For $i > 0$, suppose that $s = (\mathtt{s}, \mathtt{x}, 0) \in [\mathcal{R}_i]^R \setminus \mathtt{Target}$. This means that $s \longrightarrow (\mathtt{s}, \mathtt{x}', 1) \in \mathcal{R}_{i-1}$ for some $\mathtt{x}' \preceq \mathtt{x}$, and hence $s' \longrightarrow (\mathtt{s}, \mathtt{x}', 1)$ for all s' s.t. $s \preceq s'$.

By Lemma 5.3, there is a $j' \in \mathbb{N}$ such that $[\mathcal{R}_i]^R \setminus \mathtt{Target} = [\mathcal{R}_{j'}]^R \setminus \mathtt{Target}$ for all $i \geq j'$. Since $\mathcal{R}_i \supseteq \mathtt{Target}$ for all $i \geq 0$ it follows that $[\mathcal{R}_i]^R = [\mathcal{R}_{j'}]^R$ for all $i \geq j'$.

Since the graph of $\mathcal{G}$ is bipartite (as explained in Section 5.1), $[Pre_{\mathcal{G}}(\mathcal{R}_i)]^x = [Pre_{\mathcal{G}}([\mathcal{R}_i]^R)]^x$ and $[\widetilde{Pre_{\mathcal{G}}}(\mathcal{R}_i)]^{1-x} = [\widetilde{Pre_{\mathcal{G}}}([\mathcal{R}_i]^R)]^{1-x}$. Since $[\mathcal{R}_i]^R = [\mathcal{R}_{j'}]^R$ for all $i \geq j'$, we have $[Pre_{\mathcal{G}}(\mathcal{R}_i)]^x = [Pre_{\mathcal{G}}([\mathcal{R}]^R_{j'})]^x \subseteq \mathcal{R}_{j'+1}$ and $[\widetilde{Pre_{\mathcal{G}}}(\mathcal{R}_i)]^{1-x} = [\widetilde{Pre_{\mathcal{G}}}([\mathcal{R}]^R_{j'})]^{1-x} \subseteq \mathcal{R}_{j'+1}$. It then follows that $\mathcal{R}_i = \mathcal{R}_j$ for all $i \geq j := j' + 1$. $\qquad\square$

Termination of Parity Scheme. We prove that the scheme from Section 4 terminates under the condition that the reachability sets are computable and that there exists a finite attractor. This suffices since, by the part above, the reachability scheme terminates, thus yielding computability of the reachability set. However, here we prove termination of the parity scheme with no further assumption on the reachability sets other than their computability.

We first prove two immediate auxiliary lemmas.

Lemma 5.5. *A closable set intersects every attractor.*

Proof. In any closable set, the players can choose strategies that force the game to remain in the set surely. The lemma now follows since an attractor is visited almost surely by any run, and this would be impossible if the attractor did not have any element in the set. $\qquad\square$

Lemma 5.6. $C_n(\mathcal{G})$ is a $(1-x)$-trap.

Proof. $C_0(\mathcal{G})$ is trivially a $(1-x)$-trap. For $i \geq 1$, the result follows immediately from the definition of $C_n(\mathcal{G})$ in Eq 4.1 as the complement of a force set (by Lemma 3.3). $\qquad\square$

Lemma 5.7. *There is a finite $j \in \mathbb{N}$ such that $X_i = X_j$ for all $i \geq j$.*

Proof. We will prove the claim by showing that $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$ in the definition of $\mathcal{Y}_i$ contains an element from the attractor, and that the $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$ sets constructed in different steps i are disjoint. First, $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$ is an x-trap by Lemma 5.6. Hence it is closable, and therefore Lemma 5.5 implies that it contains an element from the attractor. Second, by the definition of the $\ominus$ operator, X_i and $\mathcal{G} \ominus X_i \ominus Z_i$ are disjoint. Since $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i) \subseteq S \setminus X_i \setminus Z_i$, it follows that $\mathcal{Y}_i$ is the *disjoint* union of X_i and $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$. Then, the definition of X_i implies that $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i) \subseteq \mathcal{Y}_i \setminus \bigcup_{j<i} \mathcal{Y}_j$. Hence, if $j \neq i$, $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$ and $C_{n-1}(\mathcal{G} \ominus X_j \ominus Z_j)$ are disjoint. Since all $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$ sets are disjoint, and each of them contains at least one element of the attractor, and the attractor is finite, the algorithm terminates in at most $|A|$ steps. $\qquad\square$

Computability. Regular languages of configurations are effectively closed under the operations of upward-closure, predecessor, set-theoretic union, intersection, and complement [ABD08]. For completeness, we show these properties below.

Lemma 5.8. *If P is a regular set of configurations, then its upward-closure $P\!\uparrow$ is effectively regular.*

Proof. A regular set P of configurations is by definition of the form

$$P = \{(\mathtt{s}, \mathtt{x}, i) : \mathtt{s} \in \mathtt{S}, i \in \{0,1\}, \mathtt{x} \in \mathtt{X}_{\mathtt{s},\mathtt{i}}\}$$

where the $\mathtt{X}_{\mathtt{s},\mathtt{i}}$'s are regular sets of channel contents. It thus suffices to show that $\mathtt{X}\!\uparrow := \{\mathtt{x} : \exists \mathtt{x}' \in \mathtt{X}.\, \mathtt{x}' \preceq \mathtt{x}\}$ is an effectively regular set of channel contents when $\mathtt{X}$ is a regular set of channel contents. By definition, $\mathtt{X}$ is a finite union of sets of the form $\mathtt{Y} \subseteq (\mathtt{M}^*)^{\mathtt{C}}$ with $\mathtt{Y}(\mathtt{c})$ regular for every $c \in \mathtt{C}$. Let $\mathtt{X}\!\uparrow$ be the union of the $\mathtt{Y}\!\uparrow$, where, for every $c \in \mathtt{C}$, a finite automaton recognizing $\mathtt{Y}\!\uparrow(c)$ is obtained from a finite automaton recognizing $\mathtt{Y}(\mathtt{c})$ by adding a self-loop labeled with M on every state thereof. $\square$

Lemma 5.9. *If P, Q are regular sets of configurations, then $P \cup Q$, $P \cap Q$, and $S \setminus P$ are effectively regular sets of configurations.*

Proof. The proof is very similar to the one in the previous lemma, by exploiting the fact that regular languages are closed under the operations of union, intersection, and complement. $\square$

Lemma 5.10. *If P is a regular set of configurations, then $Pre_G(P)$ is an effectively regular set of configurations.*

Proof. Let P be a regular set of configurations. By a case analysis on which transition is taken, we can write

$$Pre_G(P) = \bigcup_{\mathtt{t} \in \mathtt{T}} Pre_G(P, \mathtt{t}) \cup Pre_G^R(P)$$

where

$$Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{nop}} \mathtt{s}'\right) := \{(\mathtt{s}, \mathtt{x}, 1) : (\mathtt{s}', \mathtt{x}, 0) \in P\}$$

$$Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{c!m}} \mathtt{s}'\right) := \{(\mathtt{s}, \mathtt{x}, 1) : (\mathtt{s}', \mathtt{x}', 0) \in P.\, \mathtt{x}'(c) = w \cdot \mathtt{m}, \mathtt{x} = \mathtt{x}[\mathtt{c} \mapsto w]\}$$

$$Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{c?m}} \mathtt{s}'\right) := \{(\mathtt{s}, \mathtt{x}, 1) : (\mathtt{s}', \mathtt{x}', 0) \in P.\, \mathtt{x} = \mathtt{x}'[\mathtt{c} \mapsto \mathtt{m} \cdot \mathtt{x}(\mathtt{c})]\}$$

$$Pre_G^R(P) := \{(\mathtt{s}, \mathtt{x}, 0) : (\mathtt{s}', \mathtt{x}', 1) \in P.\, \mathtt{x}' \preceq \mathtt{x}\} = \{(\mathtt{s}, \mathtt{x}', 0) : (\mathtt{s}', \mathtt{x}', 1) \in P\}\!\uparrow$$

Then, $Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{nop}} \mathtt{s}'\right)$ is clearly effectively regular, $Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{c!m}} \mathtt{s}'\right)$ is regular, because regular languages are effectively closed under (right) quotients, $Pre_G\left(P, \mathtt{s} \xrightarrow{\mathtt{c?m}} \mathtt{s}'\right)$ is regular, because regular language are effectively closed under (left) concatenation with single symbols, and $Pre_G^R(P)$ is effectively regular by Lemma 5.8. $\square$

The lemmas above show that all operations used in computing $Force^x(G, \mathtt{Target})$ effectively preserve regularity. Thus we obtain the following lemma.

Lemma 5.11. *If $\mathtt{Target}$ is regular, then $Force^x(G, \mathtt{Target})$ is effectively regular.*

Lemma 5.12. *For each n, $C_n(G)$ is effectively regular.*

Proof. The set S is regular, and hence $C_0(G) = S$ is effectively regular. The result for $n > 0$ follows from Lemma 5.11 and from the fact that the rest of the operations used to build $C_n(G)$ are those of set complement and union. $\square$

5.3. **Construction of regular winning strategies.** In this section, we show that the memoryless winning strategies constructed in Theorem 5.1 can be finitely represented as a (finite) list of rules with regular guards on the channel contents. This representation can be easily turned in a more low-level one, e.g., a finite automaton with output reading the channel contents and outputting the rule do be played next, but for the ease of presentation we have chosen a more high-level description.

Preliminaries. Let $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda, \mathtt{Col})$ be a SG-LCS. A *(memoryless) regular SG-LCS strategy* $\mathtt{f}$ for Player x is a finite list of guarded rules $\{\mathtt{s}_i, X_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i'\}_{i=1}^n$, where the *guard* $X_i \subseteq (\mathtt{M}^*)^{\mathtt{C}}$ is a regular set of channel contents and $\mathtt{s}_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i'$ is a transition in $\mathtt{T}$ s.t. $\mathtt{s}_i \in \mathtt{S}^x$ and:

- If $\mathtt{op}_i = \mathtt{c}?\mathtt{m}$, every $\mathtt{x} \in X_i$ has $\mathtt{m}$ as the first symbol of $\mathtt{x}(\mathtt{c})$.
- Guards for the same control state are disjoint; i.e., for each i, j, if $\mathtt{s}_i = \mathtt{s}_j$ then $X_i \cap X_j = \emptyset$.

The *domain* of a regular SG-LCS strategy $\mathtt{f}$ is

$$dom(\mathtt{f}) = \{(\mathtt{s}, \mathtt{x}) : \text{ there exists a guarded rule } \mathtt{s}, X \overset{\mathtt{op}}{\longrightarrow} \mathtt{s}' \in \mathtt{f} \text{ s.t. } \mathtt{x} \in X\}$$

Intuitively, the rule $(\mathtt{s}_i, X_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i')$ should be applied from control state $\mathtt{s}_i$ if the channel contents belong to the guard X_i. Formally, let $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \mathtt{Col})$ be the game induced by $\mathcal{L}$. The *(partial, memoryless) induced strategy* $\underline{\mathtt{f}}$ of a regular SG-LCS strategy $\mathtt{f}$ is defined, for every $(\mathtt{s}, \mathtt{x}) \in dom(\mathtt{f})$, as $\underline{\mathtt{f}}(\mathtt{s}, \mathtt{x}, 1) = (\mathtt{s}_i', \mathtt{x}', 0)$, where $\mathtt{s}_i, X_i \overset{\mathtt{op}}{\longrightarrow} \mathtt{s}_i'$ is the unique guarded rule in $\mathtt{f}$ such that $\mathtt{s}_i = \mathtt{s}$ and $\mathtt{x} \in X_i$, and $\mathtt{x}'$ is the unique channel contents s.t. $(\mathtt{s}, \mathtt{x}, 1) \longrightarrow (\mathtt{s}', \mathtt{x}', 0)$ in the game $\mathcal{G}$. If $(\mathtt{s}, \mathtt{x}) \notin dom(\mathtt{f})$, then $\underline{\mathtt{f}}(\mathtt{s}, \mathtt{x}, 1) = \bot$.

Given two regular SG-LCS strategies $\mathtt{f}_0, \mathtt{f}_1$ with disjoint domains, their *union* $\mathtt{f}_0 \cup \mathtt{f}_1$ is the regular SG-LCS strategy obtained by concatenating the lists of guarded rules of $\mathtt{f}_0$ and $\mathtt{f}_1$.

Given two sets of configurations $Q, Q' \subseteq S$, a *selection function* from Q to Q' is any function $f : Q \mapsto Q'$ s.t., for every $(\mathtt{s}, \mathtt{x}) \in Q$,

$$f(\mathtt{s}, \mathtt{x}) \in \big(Post_{\mathcal{G}}(\mathtt{s}, \mathtt{x}) \cap Q'\big)$$

In other words, a selection function picks a legal successor in Q' for every configuration in Q.

Construction. The rest of this section is devoted to the construction of regular winning strategies for both players, as summarised by the following theorem.

Theorem 5.13. *Memoryless winning strategies for both players are effectively computable as regular SG-LCS strategies.*

We begin by showing that, if the set of selection functions is non-empty, then there are simple selection functions induced by *regular* SG-LCS strategies.

Lemma 5.14. *Let $Q, Q' \subseteq S$ be two regular sets of configurations. If there exists a selection function from Q to Q', then there exists a regular SG-LCS strategy $\mathtt{f}$ s.t. $\underline{\mathtt{f}}$ is a selection function from Q to Q'.*

Proof. Let f be a selection function from Q to Q'; in particular, the set $Post_{\mathcal{G}}(\mathtt{s}, \mathtt{x}) \cap Q'$ is non-empty for each $(\mathtt{s}, \mathtt{x}) \in Q$. Let $\mathtt{T} = \{\mathtt{s}_0 \overset{\mathtt{op}_0}{\longrightarrow} \mathtt{s}_0', \ldots, \mathtt{s}_k \overset{\mathtt{op}_k}{\longrightarrow} \mathtt{s}_k'\}$ be the finitely many transitions of $\mathcal{L}$. For every $i \in \{0, \ldots, k\}$, let P_i be the set of predecessors of Q' in Q via transition $\mathtt{s}_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i'$, i.e.,

$$P_i = Pre_{\mathcal{G}}\left(Q', \mathtt{s}_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i'\right) \cap Q = \{(\mathtt{s}_i, \mathtt{x}) \in Q : \text{ there exists } (\mathtt{s}_i', \mathtt{x}') \in Q' \cdot (\mathtt{s}_i, \mathtt{x}) \overset{\mathtt{op}_i}{\longrightarrow} (\mathtt{s}_i', \mathtt{x}')\}$$

Since Q, Q' are regular, $Pre_{\mathcal{G}}\left(Q', \mathtt{s}_i \overset{\mathtt{op}_i}{\longrightarrow} \mathtt{s}_i'\right)$ is regular (cf. Lemma 5.10), and thus P_i is regular too. Consider the sequence of (regular) sets $Q_0 = P_0$, and, for $0 < i \leq k$, $Q_i = P_i \setminus \bigcup_{0 \leq j < i} Q_j$, and let $Q_{i_0}, \ldots, Q_{i_h}$ be the subsequence of non-empty sets. Then, $\{Q_{i_0}, \ldots, Q_{i_h}\}$ is a (regular) partition of Q:

The sets are disjoint by definition, and each $(\mathtt{s},\mathtt{x}) \in Q$ belongs to some Q_{i_j} since $Post_G(\mathtt{s},\mathtt{x}) \cap Q'$ is non-empty. Let $\{X_{i_0}, \ldots, X_{i_h}\} \subseteq 2^{(\mathtt{M}^*)^\mathtt{C}}$ be the set of regular channel contents s.t., for $0 \le j \le h$, Q_{i_j} is of the form $\{(\mathtt{s}_{i_j},\mathtt{x}) : \mathtt{x} \in X_{i_j}\}$. Let $\mathtt{f}$ be the following regular SG-LCS strategy:

$$\{s_{i_0}, X_{i_0} \xrightarrow{\mathtt{op}_{i_0}} \mathtt{s}'_{i_0}, \ldots, s_{i_h}, X_{i_h} \xrightarrow{\mathtt{op}_{i_h}} \mathtt{s}'_{i_h}\} \tag{5.1}$$

By definition, $\underline{\mathtt{f}}$ is a selection function from Q to Q'. $\qquad\square$

In the next lemma, we show that regular SG-LCS strategies suffice to keep the game in regular traps.

Lemma 5.15. *If Q is a $(1-x)$-trap and regular, then there exists a regular SG-LCS strategy $\mathtt{f}$ for Player x such that $Q \subseteq V^x(\underline{\mathtt{f}}, F^{1-x}_{all}(G))(G, \square Q)$.*

Proof. By Lemma 3.1, there exists a memoryless strategy f^x for Player x with the required property. Moreover, by inspecting the proof of the lemma, we can see that f^x is defined as $f^x(s) = select(Post_G(s) \cap Q)$ for every configuration $s \in [Q]^x$, i.e., f^x is a selection function from $[Q]^x$ to Q, and, in fact, any such selection function can be taken. By Lemma 5.14, there exists a regular SG-LCS strategy $\mathtt{f}$ s.t. the induced strategy $\underline{\mathtt{f}}$ is a selection function from $[Q]^x$ to Q. $\qquad\square$

The following lemma shows that there are regular SG-LCS strategies for the reachability and safety objective (cf. Lemma 3.4).

Lemma 5.16. *Let $\mathtt{Target} \subseteq S$ be a regular set of configurations. There exist regular SG-LCS strategies $\mathtt{force}^x(G,\mathtt{Target})$ for Player x and $\mathtt{avoid}^{1-x}(G,\mathtt{Target})$ for Player $1-x$ s.t.*

$$Force^x(G,\mathtt{Target}) \subseteq W^x(\underline{\mathtt{force}^x(G,\mathtt{Target})}, F^{1-x}_{all}(G))(G, \Diamond \mathtt{Target}^{>0})$$

$$Avoid^{1-x}(G,\mathtt{Target}) \subseteq V^{1-x}(\underline{\mathtt{avoid}^{1-x}(G,\mathtt{Target})}, F^x_{all}(G))(G, \square(\stackrel{G}{\neg}\mathtt{Target}))$$

Proof. We first show a regular SG-LCS strategy for Player x for the reachability objective. Consider the sequence of sets $\mathcal{R}_0, \mathcal{R}_1, \ldots$ constructed in Section 3. By Lemma 5.4, there exists $j \in \mathbb{N}$ s.t. $\forall i > j, \mathcal{R}_i = \mathcal{R}_j$. Moreover, since $\mathcal{R}_i$ is built starting from the regular set $\mathtt{Target}$ and according to regularity-preserving operations (union, predecessor, and complement; cf. Lemmas 5.9 and 5.10), $\mathcal{R}_i$ is regular for every $0 \le i \le j$. Consider the sequence of regular sets $R_0 = \mathcal{R}_0$ and $R_i = \mathcal{R}_i \setminus \mathcal{R}_{i-1}$ for every $0 < i \le j$. Recall the definition of $force^x(G,\mathtt{Target})$ in the proof of Lemma 3.4: For every $0 < i \le j, force^x(G,\mathtt{Target})$ was uniformly defined on R_i as

$$force^x(G,\mathtt{Target})(s) = select(Post_G(s) \cap \mathcal{R}_{i-1}).$$

Therefore, there exists a selection function from R_i to $\mathcal{R}_{i-1}$, for every $0 < i \le j$. Since the R_i's and $\mathcal{R}_i$'s are regular, by Lemma 5.14, there exists a regular SG-LCS strategy f_i with domain R_i inducing such a selection function. Since the R_i's are disjoint, and since any selection function is correct, take as $\mathtt{force}^x(G,\mathtt{Target})$ the union strategy $f_0 \cup \cdots \cup f_j$. Since the actual choice of the selection function is irrelevant, we conclude that

$$Force^x(G,\mathtt{Target}) \subseteq W^x(\underline{\mathtt{force}^x(G,\mathtt{Target})}, F^{1-x}_{all}(G))(G, \Diamond \mathtt{Target}^{>0})$$

We conclude the proof by providing the required regular SG-LCS strategy for Player $1-x$ for the safety objective. By Lemma 3.3, $Avoid^{1-x}(G,\mathtt{Target})$ is an x-trap. Since $Avoid^{1-x}(G,\mathtt{Target})$ is regular, by Lemma 5.15 there exists a regular SG-LCS strategy $\mathtt{avoid}^{1-x}(G,\mathtt{Target})$ such that $Avoid^{1-x}(G,\mathtt{Target}) \subseteq V^{1-x}(\underline{\mathtt{avoid}^{1-x}(G,\mathtt{Target})}, F^x_{all}(G))(G, \square(\stackrel{G}{\neg}\mathtt{Target}))$. $\qquad\square$

To conclude the proof of Theorem 5.13, we show that regular SG-LCS strategies suffice for the parity objective (cf. Lemma 4.2).

Lemma 5.17. *There are regular SG-LCS strategies $\mathtt{f}_c^x$ for Player x and $\mathtt{f}_c^{1-x}$ for Player $1-x$ such that*

$$C_n(\mathcal{G}) \ \subseteq \ W^x(\underline{\mathtt{f}}_c^x, F_{\text{finite}}^{1-x}(\mathcal{G}))(\mathcal{G}, x\text{-}Parity^{=1})$$

$$\stackrel{\mathcal{G}}{\dashv} C_n(\mathcal{G}) \ \subseteq \ W^{1-x}(\underline{\mathtt{f}}_c^{1-x}, F_{\text{finite}}^{x}(\mathcal{G}))(\mathcal{G}, (1-x)\text{-}Parity^{>0})$$

Proof. We define regular SG-LCS strategies $\mathtt{f}_c^x$ for Player x and $\mathtt{f}_c^{1-x}$ for Player $1-x$ by induction on $n \geq 1$. By inspecting the proof of Lemma 4.2, we note that winning strategies for both players are constructed according to a case analysis on disjoint regular domains, for which winning regular SG-LCS strategies exist either by induction hypothesis, or by Lemma 5.16 (for reachability). Recall that, by Lemma 5.7, there exists $i \in \mathbb{N}$ s.t. $X_j = X_i$ for every $j > i$. Moreover, all the sets $X_j, \mathcal{Y}_j, Z_j$ involved in the construction are regular for every $0 \leq j \leq i$ since they are constructed starting from regular sets and according to regularity-preserving operations (boolean operations, cf. Lemma 5.9; force-sets, cf. Lemma 5.16).

Construction of $\mathtt{f}_c^x$. Define the two regular sets of configurations $\overline{X_j} :=_\dashv^{\mathcal{G}} X_j$ and $\overline{Z_j} :=_\dashv^{\mathcal{G}} Z_j$. By definition, $C_n(\mathcal{G}) = \overline{X_j}$. Following Lemma 4.2, we define $\mathtt{f}_c^x(s)$ depending on the membership of s in one of the following three partitions of $\overline{X_j}$:

$$\{\overline{X_j} \cap \overline{Z_j}, \ \ \overline{X_j} \cap [Z_j]^{\text{Col}<n}, \ \ \overline{X_j} \cap [Z_j]^{\text{Col}=n}\}$$

In the first case, note that $\mathcal{G} \ominus X_j \ominus Z_j$ does not contain any configurations of color $\geq n$ (cf. Lemma 4.2). Thus, by the induction hypothesis, there is a regular SG-LCS strategy $\mathtt{f}_j$ for Player x in $\mathcal{G} \ominus X_j \ominus Z_j$ such that the induced strategy has domain $\overline{X_j} \cap \overline{Z_j}$. In the second case, let $\mathtt{f}_2$ be the regular SG-LCS strategy $\texttt{force}^x(\mathcal{G} \ominus X_j, [Z_j]^{\text{Col}=n})$, for which the induced strategy has domain $\overline{X_j} \cap [Z_j]^{\text{Col}<n}$ (it exists by Lemma 5.16). Finally, in the third case, the strategy $select(Post_{\mathcal{G}}(\cdot) \cap \overline{X_j})$ witnesses the existence of a selection function from $\overline{X_j} \cap [Z_j]^{\text{Col}=n}$ to $\overline{X_j}$. Let $\mathtt{f}_3$ be a regular SG-LCS strategy inducing a selection function from $\overline{X_j} \cap [Z_j]^{\text{Col}=n}$ to $\overline{X_j}$ (it exists by Lemma 5.14). Then, $\mathtt{f}_c^x$ is defined as the union of the three previously constructed strategies:

$$\mathtt{f}_c^x := \mathtt{f}_1 \cup \mathtt{f}_2 \cup \mathtt{f}_3$$

Since the actual choice of selection function is irrelevant, $\mathtt{f}_c^x$ induces a correct strategy by the same arguments as in the proof of Lemma 4.2, i.e., $C_n(\mathcal{G}) \subseteq W^x(\underline{\mathtt{f}}_c^x, F_{\text{finite}}^{1-x}(\mathcal{G}))(\mathcal{G}, x\text{-}Parity^{=1})$.

Construction of $\mathtt{f}_c^{1-x}$. Recall that $\stackrel{\mathcal{G}}{\dashv} C_n(\mathcal{G}) = \mathcal{Y}_j = X_j$, and, for every $0 \leq i \leq j$, $\mathcal{Y}_i = X_i \cup C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$. For every $1 \leq i \leq j$, let $\mathtt{f}_i^1$ be the regular SG-LCS strategy $\texttt{force}^{1-x}(\mathcal{G}, \mathcal{Y}_{i-1})$ with domain $X_i \setminus \mathcal{Y}_{i-1}$ (it exists by Lemma 5.16). By the induction hypothesis, there is also a regular SG-LCS strategy $\mathtt{f}_i^2$ such that the induced strategy has domain $C_{n-1}(\mathcal{G} \ominus X_i \ominus Z_i)$, which is winning a.s. for Player $1-x$ on this domain. Then, $\mathtt{f}_c^{1-x}$ is defined as

$$\mathtt{f}_c^{1-x} := \mathtt{f}_1^1 \cup \mathtt{f}_1^2 \cup \cdots \cup \mathtt{f}_j^1 \cup \mathtt{f}_j^2$$

By reasoning as in the proof of Lemma 4.2, $\mathtt{f}_c^{1-x}$ induces a correct strategy, i.e., $\stackrel{\mathcal{G}}{\dashv} C_n(\mathcal{G}) \subseteq W^{1-x}(\underline{\mathtt{f}}_c^{1-x}, F_{\text{finite}}^x(\mathcal{G}))(\mathcal{G}, (1-x)\text{-}Parity^{>0})$. $\square$
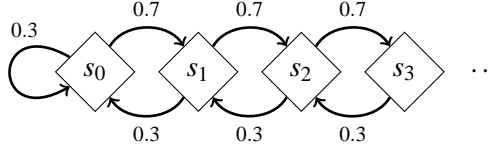
FIGURE 2. Finite attractor requirement.

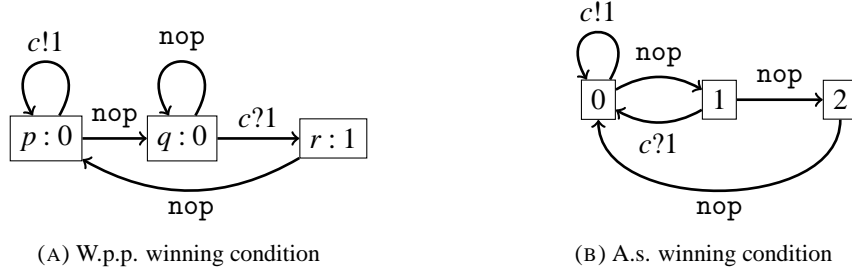(A) W.p.p. winning condition

(B) A.s. winning condition

FIGURE 3. Infinite memory helps Player 1.

6. CONCLUSIONS AND DISCUSSION

We have presented a scheme for solving stochastic games with a.s. and w.p.p. parity winning conditions under the two requirements that (i) the game contains a finite attractor and (ii) both players are restricted to finite-memory strategies. We have shown that this class of games is memoryless determined. The method is instantiated to prove decidability of a.s. and w.p.p. parity games induced by lossy channel systems.

The two above requirements are both necessary for our method. To see why our scheme fails if the game lacks a **finite attractor**, consider the game in Figure 2 (a variant of the Gambler's ruin problem). All states are random, i.e., $S^0 = S^1 = \emptyset$, and $\texttt{Col}(s_0) = 1$ and $\texttt{Col}(s_i) = 0$ when $i > 0$. The probability to go right from any state is 0.7 and the probability to go left (or to make a self-loop in s_0) is 0.3. This game does not have any finite attractor. It can be shown that the probability to reach s_0 infinitely often is 0 for all initial states. However, our construction will classify all states as winning for Player 1. More precisely, the construction of $C_1(\mathcal{G})$ converges after one iteration, with $\mathcal{Z}_\alpha = S$ and $\mathcal{X}_\alpha = \mathcal{Y}_\alpha = \emptyset$ for all α, and $C_1(\mathcal{G}) = S$. Intuitively, the problem is that even if the force-set of $\{s_0\}$ (which is the entire set of states) is visited infinitely many times, the probability of visiting $\{s_0\}$ infinitely often is still zero, since the probability of returning to $\{s_0\}$ gets smaller and smaller. Such behavior is impossible in a game graph that contains a finite attractor.

Our scheme also fails when the players are not both restricted to **finite-memory strategies**. Solving a game under a finite-memory restriction is a different problem from when arbitrary strategies are allowed (not a sub-problem). In fact, it was shown in [BBS07] that for arbitrary strategies, the problem is undecidable. We show two simple examples of stochastic games on LCSs where the two problems yield different results (see also [BBS07]). In one case, we show that infinite memory is more powerful for Player 1 with a w.p.p. objective (cf. Figure 3a), while in the other case infinite memory helps w.r.t. an a.s. objective (cf. Figure 3b). In both cases, Player 0 does not play in the game, thus the memory allowed to her is irrelevant.

First, we show that infinite memory is more powerful for w.p.p. objectives. In Figure 3a, Player 1 plays on control states p, q, and r. Player 1's objective is to visit state r infinitely often

w.p.p.. To ensure this, from state p Player 1 pumps up the channel to a sufficiently large size k (which can be done a.s. for any k given enough time), and then she goes to the risk state q. If each message can be lost independently with probability $\frac{1}{2}$, the probability that all messages are lost, and thus that Player 1 is stuck forever in q, is 2^{-k}. Otherwise, with probability $1 - 2^{-k}$ Player 1 can visit r once, and then go back to p. The strategy of Player 1 is to realise an infinite sequence $k_0 < k_1 < \cdots$ s.t. the probability of visiting state r infinitely often, which is $\prod_{i=0}^{\infty}(1 - 2^{-k_i})$, can be made strictly positive. Clearly, if Player 1 has infinite memory, then she can realize such a sequence by distinguishing different visits to control state p and same channel contents. On the other side, if Player 1 is restricted to finite memory, then either the game eventually stays forever in p (which is losing), or the infinite sequence $k_0, k_1, \ldots$ is upper-bounded by some finite n, which makes the infinite product above equal to 0. In both cases, Player 1 loses if she has only finite memory.

Notice that Player 1 wins not only w.p.p., but even limit-sure in this example. In other words, for every $\varepsilon > 0$ there is an infinite-memory strategy s.t. the parity objective is satisfied with probability $\geq \varepsilon$. We don't know whether there are examples where a similar phenomenon can be reproduced under finite-memory/memoryless strategies.

We now show that infinite memory is more powerful for a.s. objectives. An example similar to the previous case can be given for the a.s. winning mode with a 3-color parity condition. In Figure 3b, Player 1 controls states 0, 1, and 2, whose color equals their name. Thus, the objective of Player 1 is to a.s. visit state 1 infinitely often and state 2 only finitely often. The strategy is similar as in the previous example: Player 1 tries to pump up the channel in state 0, and then she goes to the risk state 1. From here, with low probability all messages are lost, and the penalty is to visit state 2 once. Otherwise, the game can go back directly to state 0 without visiting state 2. In both cases, the game restarts afresh from state 0. An analysis as in the previous example shows that, if Player 1 is restricted to finite memory, then the probability of visiting state 2 from state 1 can be bounded from below. This implies that, whenever state 1 is visited infinitely often, then so is state 2 a.s., and so Player 1 is losing. On the other hand, there is an infinite-memory strategy for Player 1 s.t. the probability of visiting state 2 for n times goes to 0 as n goes to infinity, which implies that the probability of visiting state 2 only finitely often is 1.

As future work, we will consider extending our framework to (fragments of) probabilistic extensions of other models such as Petri nets and noisy Turing machines [AHM07].

REFERENCES

[ABD08]   P. A. Abdulla, A. Bouajjani, and J. D'Orso. Monotonic and downward closed games. *Journal of Logic and Computation*, 18(1):153–169, 2008.

[ABRS05]   P. A. Abdulla, N. Bertrand, A. Rabinovich, and Ph. Schnoebelen. Verification of probabilistic systems with faulty communication. *Information and Computation*, 202(2):105–228, 2005.

[ACMS13]   P. A. Abdulla, L. Clemente, R. Mayr, and S. Sandberg. Stochastic parity games on lossy channel systems. In *Proc. of QEST'13*, LNCS, pages 338–354, Berlin, Heidelberg, 2013. Springer-Verlag.

[AHdA⁺08] P. A. Abdulla, N. B. Henda, L. de Alfaro, R. Mayr, and S. Sandberg. Stochastic games with lossy channels. In *FOSSACS*, volume 4962 of *LNCS*, 2008.

[AHM07]   P. A. Abdulla, N. B. Henda, and R. Mayr. Decisive Markov chains. *Logical Methods in Computer Science*, 3, 2007.

[AJ93]   P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. In *LICS*, pages 160–170, 1993.

[AJ96]   P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.

[AR03]   P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *FOSSACS*, volume 2620 of *LNCS*, pages 39–53, 2003.

[BBS07]   C. Baier, N. Bertrand, and Ph. Schnoebelen. Verifying nondeterministic probabilistic channel systems against ω-regular linear-time properties. *ACM Trans. on Comp. Logic*, 9, 2007.

[Ber79]   J. Berstel. *Transductions and Context-Free Languages*. B. G. Teubner Stuttgart, 1979.

[Bil86]   P. Billingsley. *Probability and Measure*. Wiley, New York, NY, 1986. Second Edition.

[BS03]   N. Bertrand and Ph. Schnoebelen. Model checking lossy channels systems is probably decidable. In *FOS-SACS*, volume 2620 of *LNCS*, pages 120–135, 2003.

[BS13]   N. Bertrand and Ph. Schnoebelen. Solving stochastic Büchi games on infinite arenas with a finite attractor. In *QAPL'13*, pages 116–131, March 2013.

[BZ83]   D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 2(5):323–342, April 1983.

[CdAH06]   K. Chatterjee, L. de Alfaro, and T. Henzinger. Strategy improvement for concurrent reachability games. In *QEST*, pages 291–300. IEEE Computer Society Press, 2006.

[CGP99]   E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, Dec. 1999.

[CJH03]   K. Chatterjee, M. Jurdziński, and T. Henzinger. Simple stochastic parity games. In *CSL*, volume 2803 of *LNCS*, pages 100–113. Springer Verlag, 2003.

[Con92]   A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, February 1992.

[dAH00]   L. de Alfaro and T. Henzinger. Concurrent omega-regular games. In *LICS*, pages 141–156, Washington - Brussels - Tokyo, June 2000. IEEE.

[dAHK98]   L. de Alfaro, T. Henzinger, and O. Kupferman. Concurrent reachability games. In *FOCS*, pages 564–575. IEEE Computer Society Press, 1998.

[EWY08]   K. Etessami, D. Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. In *ICALP*, volume 5125 of *LNCS*. Springer, 2008.

[EY05]   K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *ICALP*, volume 3580 of *LNCS*, pages 891–903, 2005.

[EY08]   K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. *LMCS*, 4, 2008.

[Fin94]   A. Finkel. Decidability of the termination problem for completely specified protocols. *Distributed Computing*, 7(3):129–135, 1994.

[Hig52]   G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3)*, 2(7):326–336, 1952.

[May03]   R. Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 297(1-3):337–354, March 2003.

[Rab03]   A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *ICALP*, volume 2719 of *LNCS*, pages 1008–1021, 2003.

[Sch01]   Ph. Schnoebelen. Bisimulation and other undecidable equivalences for lossy channel systems. In N. Kobayashi and B. C. Pierce, editors, *Theoretical Aspects of Computer Software*, volume 2215 of *LNCS*, pages 385–399. Springer, 2001.

[Sch02]   Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[Sha53]   L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, October 1953.

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

[WLJ10]   K. Wang, N. Li, and Z. Jiang. Queueing system with impatient customers: A review. In *IEEE International Conference on Service Operations and Logistics and Informatics (SOLI)*, pages 82–87. IEEE, 2010.

[Zie98]   W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS*, 200:135–183, 1998.