Stochastic Parity Games on Lossy Channel Systems

Parosh Aziz Abdulla[1], Lorenzo Clemente[2], Richard Mayr[3], and Sven Sandberg[1]

[1] Uppsala University [2] LaBRI, University of Bordeaux I [3] University of Edinburgh

Abstract. We give an algorithm for solving stochastic parity games with almost-sure winning conditions on *lossy channel systems*, for the case where the 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 [15]. 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., [24,17,19,15]), 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 [21,22,20]. 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 finite-state machines that communicate by asynchronous message passing via unbounded FIFO communication channels. They are also known as communicating finite-state machines (CFSM) [13].

A *Lossy Channel System (LCS)* [7] 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.

A *Probabilistic Lossy Channel System (PLCS)* [10,8] is a probabilistic variant of LCS where, in each computation step, messages are lost from the channels with a given probability. In [5], a *game extension* of PLCS was introduced where the players control transitions in the control graph and message losses are probabilistic.

The original motivation for LCS and PLCS was to capture the behavior of communication protocols; such protocols are designed to operate correctly even if the communication medium is unreliable (i.e., if messages can be lost). However, Channel Systems (aka CFSM) are a very expressive model that can encode the behavior of Turing machines, by storing the content of a Turing tape in a channel [13]. 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. The only requirement is that the system re-visits a certain finite core region (we call this an attractor; see below) with probability one, e.g.,

- 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 [26] section II.B. Like in some works cited in [26], the maximal waiting time in our model is exponentially distributed. In basic PLCS, unlike in [26], 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 finite attractor).
- 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).
- Systems modeling 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. Several algorithms for symbolic model checking of PLCS have been presented [1,23]. Markov decision processes (i.e., $1\frac{1}{2}$-player games) on infinite graphs induced by PLCS were studied in [9], which shows that $1\frac{1}{2}$-player games with almost-sure Büchi objectives are pure memoryless determined and decidable. This result was later generalized to $2\frac{1}{2}$-player games [5], and further extended to generalized Büchi objectives [11]. 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 [9]. (Undecidability and infinite memory requirement are separate results, since decidability does not imply the existence of finite-memory strategies in infinite-state games). If players are restricted to finite-memory strategies, the $1\frac{1}{2}$-player game with positive probability parity objectives (even the more general *Streett objectives*) becomes decidable [9]. Note that the finite-memory case and the infinite-memory one are a priori incomparable problems, and neither subsumes the other. Cf. Section 7.

Non-stochastic (2-player) parity games on infinite graphs were studied in [27], 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 [18,15,14].

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* winning 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 non-trivial generalization of the well-known scheme for non-stochastic games in [27] (see the remark in Section 4). 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 [27]; 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 6), we show that the scheme can be instantiated for lossy channel systems. The instantiation requires the use of a much more involved framework than the classical one for well quasi-ordered transition systems [3] (see the remark in Section 6). 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) [9]. 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.

Full proofs are available in the technical report [4].

2 Preliminaries

Notation. Let $\mathbb{O}$ and $\mathbb{N}$ denote the set of ordinal resp. natural numbers. 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 $f(x) = \perp$ to denote that f is undefined on x, and define $dom(f) := \{x |\ f(x) \neq \perp\}$. We say that f is an *extension* of g if $g(x) = f(x)$ whenever $g(x) \neq \perp$. 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.

[1] In the game community (e.g., [27]) the word *attractor* is used to denote what we call a *force set* in Section 3. In the infinite-state systems community (e.g., [1,6]), the word is used in the same way as we use it in this paper.

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.) $\texttt{Col} : S \to \{0, \ldots, n\}$, where $\texttt{Col}(s)$ is called the *color* of state s. Let $Q \subseteq S$ be a set of states. We use $\overset{\mathcal{G}}{\neg} Q := S - Q$ 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* 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 $F^x_{all}(\mathcal{G})$, $F^x_{finite}(\mathcal{G})$, and $F^x_{\emptyset}(\mathcal{G})$ 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 [12]. First, we define the measure for total strategies, and then we extend it to general (partial) strategies. Let $\Omega^s = sS^\omega$ denote the set of all infinite sequences of states starting from s. Consider a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \mathtt{Col})$, an initial state s, and total strategies f^x and f^{1-x} of Players x and $1-x$. 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 [12]. 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., [16]). Formulas are interpreted on the structure $(S, \longrightarrow)$. We use $\llbracket\varphi\rrbracket^s$ to denote the set of runs starting from s that satisfy the CTL^* path-formula φ. This set is measurable [25], and we just write $\mathcal{P}_{\mathcal{G},s,f^x,f^{1-x}}(\varphi) \sim c$ instead of $\mathcal{P}_{\mathcal{G},s,f^x,f^{1-x}}(\llbracket\varphi\rrbracket^s) \sim c$.

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-Parity* $:= \bigvee_{i \in \{0,\dots,n\} \wedge (i \bmod 2) = x} (\Box\Diamond[S]^{\mathtt{Col}=i} \wedge \Diamond\Box[S]^{\mathtt{Col}\leq i})$.

Winning Sets. For a strategy f^x of Player x, and a set F^{1-x} of strategies of Player $1-x$, we define $W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{\sim c}) := \{s \mid \forall f^{1-x} \in F^{1-x}. \mathcal{P}_{\mathcal{G},s,f^x,f^{1-x}}(\varphi) \sim c\}$. 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 *almost surely (a.s.)* wins from s. If $s \in W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{>0})$, then we say that Player x wins *with positive probability (w.p.p.)*. We define $V^x(f^x, F^{1-x})(\mathcal{G}, \varphi) := \{s \mid \forall f^{1-x} \in F^{1-x}. \mathit{Runs}(\mathcal{G}, s, f^x, f^{1-x}) \subseteq \llbracket\varphi\rrbracket^s\}$. 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., i.e., $V^x(f^x, F^{1-x})(\mathcal{G}, \varphi) \subseteq W^x(f^x, F^{1-x})(\mathcal{G}, \varphi^{=1})$.

Determinacy and Solvability. A game is called *determined*, wrt. a winning condition and two sets F^0, F^1 of strategies of Player 0, resp. Player 1, if, from every state, one of the players x has a strategy $f^x \in F^x$ that wins against all strategies $f^{1-x} \in F^{1-x}$ of the opponent. 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.

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 $\mathcal{P}_{\mathcal{G},s,f^0,f^1}(\Diamond A) = 1$. 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 $\mathcal{P}_{\mathcal{G},s,f^0,f^1}(\Box\Diamond A) = 1$, i.e., the attractor will be visited infinitely often a.s..

Transition Systems. Consider strategies $f^x \in F^x_{\emptyset}$ and $f^{1-x} \in F^{1-x}_{\mathit{finite}}$ 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 $\mathcal{G}, f^{1-x}, f^x$ to be the pair $(S_M, \rightsquigarrow)$ where $S_M = S \times M$, and $\rightsquigarrow \subseteq S_M \times S_M$ such that $(s_1, m_1) \rightsquigarrow (s_2, m_2)$ 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 $\mathcal{G}$ 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 $\rho \in \mathit{Runs}(\mathcal{G}, s, f^x, f^{1-x})$ 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 [27]) for checking reachability winning conditions, and state some of its properties that we use in the later sections. Below, fix a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P, \mathtt{Col})$.

Reachability Properties. Fix a state $s \in S$ and sets of states $Q, Q' \subseteq S$. Let $\mathit{Post}_{\mathcal{G}}(s) := \{s' : s \longrightarrow s'\}$ denote the set of *successors* of s. Extend it to sets of states by $\mathit{Post}_{\mathcal{G}}(Q) := \bigcup_{s \in Q} \mathit{Post}_{\mathcal{G}}(s)$. Note that for any given state $s \in S^R$, $P(s, \cdot)$ is a probability distribution over $\mathit{Post}_{\mathcal{G}}(s)$. Let $\mathit{Pre}_{\mathcal{G}}(s) := \{s' : s' \longrightarrow s\}$ denote the set of *predecessors* of s, and extend it to sets of states as above. We define $\widetilde{\mathit{Pre}_{\mathcal{G}}}(Q) := \overset{\mathcal{G}}{\neg} \mathit{Pre}_{\mathcal{G}}\left(\overset{\mathcal{G}}{\neg} Q\right)$, i.e., it denotes the set of states whose successors *all* belong to Q. We say that Q is *sink-free* if $\mathit{Post}_{\mathcal{G}}(s) \cap Q \neq \emptyset$ for all $s \in Q$, and *closable* if it is sink-free and $\mathit{Post}_{\mathcal{G}}(s) \subseteq Q$ 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.

If $\overset{\mathcal{G}}{\neg} Q$ is closable, we define the *subgame* $\mathcal{G} \ominus Q := (Q', [Q']^0, [Q']^1, [Q']^R, \longrightarrow', P', \mathtt{Col}')$, where $Q' := \overset{\mathcal{G}}{\neg} Q$ 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 $s \in S^R$ since $\overset{\mathcal{G}}{\neg} Q$ is closable. We use $\mathcal{G} \ominus Q_1 \ominus Q_2$ to denote $(\mathcal{G} \ominus Q_1) \ominus Q_2$.

For $x \in \{0, 1\}$, we say that Q is an *x-trap* if it is closable and $\mathit{Post}_{\mathcal{G}}(s) \subseteq Q$ 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 (adapted from [27]) 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 1. *If Q is a $(1-x)$-trap, then there exists a memoryless strategy $f^x \in F^x_{\emptyset}(\mathcal{G})$ for Player x such that $Q \subseteq V^x(f^x, F^{1-x}_{\mathit{all}}(\mathcal{G}))(\mathcal{G}, \Box Q)$.*

Scheme. Given a set $\mathtt{Target} \subseteq S$, we give a scheme for computing a partitioning of S into two sets $\mathit{Force}^x(\mathcal{G}, \mathtt{Target})$ and $\mathit{Avoid}^{1-x}(\mathcal{G}, \mathtt{Target})$ that are winning for Players x and $1-x$. More precisely, we define a memoryless strategy that allows Player x to force the game to $\mathtt{Target}$ w.p.p.; and define a memoryless strategy that allows Player $1-x$ to surely avoid $\mathtt{Target}$.

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:

$$\begin{aligned} \mathcal{R}_0 &:= \mathtt{Target}; \\ \mathcal{R}_{i+1} &:= \mathcal{R}_i \cup [\mathit{Pre}_{\mathcal{G}}(\mathcal{R}_i)]^R \cup [\mathit{Pre}_{\mathcal{G}}(\mathcal{R}_i)]^x \cup [\widetilde{\mathit{Pre}_{\mathcal{G}}}(\mathcal{R}_i)]^{1-x} && \text{if } i+1 \text{ is a successor ordinal;} \\ \mathcal{R}_i &:= \bigcup_{j<i} \mathcal{R}_j && \text{if } i > 0 \text{ is a limit ordinal;} \end{aligned}$$

$$\mathit{Force}^x(\mathcal{G}, \mathtt{Target}) := \bigcup_{i \in \mathbb{O}} \mathcal{R}_i; \qquad \mathit{Avoid}^{1-x}(\mathcal{G}, \mathtt{Target}) := \overset{\mathcal{G}}{\neg}\, \mathit{Force}^x(\mathcal{G}, \mathtt{Target}).$$

First, we show that the iteration above converges (possibly in infinitely many steps). To this end, we observe that $\mathcal{R}_i \subseteq \mathcal{R}_{i+1}$ if $i+1$ is a successor ordinal and $\mathcal{R}_j \subseteq \mathcal{R}_i$ if $j < i$ and i is a limit ordinal. Therefore $\mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \cdots$. Since the sequence is non-decreasing and since the sequence is bounded by S, it will eventually converge. Define α to be the smallest ordinal such that $\mathcal{R}_\alpha = \mathcal{R}_i$ for all $i \geq \alpha$. This gives the following lemma, which also implies that the Avoid^{1-x} set is a trap for Player x. (Lemmas 2 and 3 are adapted from [27], where they are stated in a non-probabilistic setting.)

Lemma 2. *There is an $\alpha \in \mathbb{O}$ such that $\mathcal{R}_\alpha = \bigcup_{i \in \mathbb{O}} \mathcal{R}_i$.*

Lemma 3. *$\mathit{Avoid}^{1-x}(\mathcal{G}, \mathtt{Target})$ is an x-trap.*

The following lemma shows correctness of the construction. In fact, it shows that a winning player also has a memoryless winning strategy.

Lemma 4. *There is a memoryless strategy $\mathit{force}^x(\mathcal{G}, \mathtt{Target}) \in F_{\emptyset}^x(\mathcal{G})$ such that $\mathit{Force}^x(\mathcal{G}, \mathtt{Target}) \subseteq W^x(\mathit{force}^x(\mathcal{G}, \mathtt{Target}), F_{\mathit{all}}^{1-x}(\mathcal{G}))(\mathcal{G}, \Diamond \mathtt{Target}^{>0})$; and a memoryless strategy $\mathit{avoid}^{1-x}(\mathcal{G}, \mathtt{Target}) \in F_{\emptyset}^{1-x}(\mathcal{G})$ such that $\mathit{Avoid}^x(\mathcal{G}, \mathtt{Target}) \subseteq V^{1-x}(\mathit{avoid}^{1-x}(\mathcal{G}, \mathtt{Target}), F_{\mathit{all}}^x(\mathcal{G}))(\mathcal{G}, \Box(\overset{\mathcal{G}}{\neg}\, \mathtt{Target}))$.*

The first claim of the lemma can be proven using transfinite induction on i to show that it holds for each state $s \in \mathcal{R}_i$. The second claim follows from Lemma 3 and Lemma 1.

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.

By induction on n, we define two sequences of functions $\mathcal{C}_0, \mathcal{C}_1, \ldots$ and $\mathcal{D}_0, \mathcal{D}_1, \ldots$ s.t., for each $n \geq 0$ and game $\mathcal{G}$ of rank at most n, $\mathcal{C}_n(\mathcal{G})$ characterizes the states from which Player x is winning a.s., where $x = n \bmod 2$, and $\mathcal{D}_n(\mathcal{G})$ characterizes the set of states from which Player x is winning w.p.p.. The scheme for $\mathcal{C}_n$ is related to [27]; cf.

the remark at the end of this section. In both cases, we provide a memoryless strategy that is winning for Player x; Player $1-x$ is always restricted to finite-memory.

For the base case, let $\mathcal{C}_0(\mathcal{G}) := S$ and $\mathcal{D}_0(\mathcal{G}) := S$ for any game $\mathcal{G}$ of rank 0. Indeed, from any configuration Player 0 trivially wins a.s./w.p.p. because there is only color 0.

For $n \geq 1$, let $\mathcal{G}$ be a game of rank n. $\mathcal{C}_n(\mathcal{G})$ is defined with the help of two auxiliary transfinite sequences $\{\mathcal{X}_i\}_{i\in\mathbb{O}}$ and $\{\mathcal{Y}_i\}_{i\in\mathbb{O}}$. The construction ensures that $\mathcal{X}_0 \subseteq \mathcal{Y}_0 \subseteq \mathcal{X}_1 \subseteq \mathcal{Y}_1 \subseteq \cdots$, and that the elements of $\mathcal{X}_i, \mathcal{Y}_i$ are winning w.p.p. for Player $1-x$. The construction alternates as follows. In the inductive step, we have already constructed $\mathcal{X}_j$ and $\mathcal{Y}_j$ for all $j < i$. Our construction of $\mathcal{X}_j$ and $\mathcal{Y}_j$ is in three steps:

1. $\mathcal{X}_i$ is the set of states where Player $1-x$ can force the run to visit $\bigcup_{j<i} \mathcal{Y}_j$ w.p.p..
2. Find a set of states where Player $1-x$ wins w.p.p. in $\mathcal{G} \ominus \mathcal{X}_i$.
3. Take $\mathcal{Y}_i$ to be the union of $\mathcal{X}_j$ and the set constructed in step 2.

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

$$\begin{aligned} \mathcal{X}_i &:= \mathit{Force}^{1-x}(\mathcal{G}, \textstyle\bigcup_{j<i} \mathcal{Y}_j), \\ \mathcal{Z}_i &:= \mathit{Force}^{x}(\mathcal{G} \ominus \mathcal{X}_i, [\overset{\mathcal{G}}{\neg}\, \mathcal{X}_i]^{\mathtt{Col}=n}), \\ \mathcal{Y}_i &:= \mathcal{X}_i \cup \mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i), \end{aligned} \qquad \mathcal{C}_n(\mathcal{G}) :=\overset{\mathcal{G}}{\neg} (\textstyle\bigcup_{i\in\mathbb{O}} \mathcal{X}_i).$$

Notice that the subgames $\mathcal{G} \ominus \mathcal{X}_i$ and $\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i$ are well-defined since (by Lemma 3) $\overset{\mathcal{G}}{\neg}\, \mathcal{X}_i$ is closable in $\mathcal{G}$, and $\overset{\mathcal{G}\ominus\mathcal{X}_i}{\neg}\, \mathcal{Z}_i$ is closable in $\mathcal{G} \ominus \mathcal{X}_i$.

We now construct $\mathcal{D}_n(\mathcal{G})$. Assume that we can construct $\mathcal{C}_n(\mathcal{G})$. We will define the transfinite sequence $\{\mathcal{U}_i\}_{i\in\mathbb{O}}$ and the auxiliary transfinite sequence $\{\mathcal{V}_i\}_{i\in\mathbb{O}}$. We again precede the formal definition with an informal explanation of the idea. The construction ensures that $\mathcal{U}_0 \subseteq \mathcal{V}_0 \subseteq \mathcal{U}_1 \subseteq \mathcal{V}_1 \subseteq \cdots$, and that all $\mathcal{U}_i$, $\mathcal{V}_i$ are winning w.p.p. for Player x in $\mathcal{G}$. The construction alternates in a similar manner to the construction of $\mathcal{C}_n$. In the inductive step, we have already constructed $\mathcal{V}_j$ for all $j < i$. We first compute the set of states where Player x can force the play to reach $\mathcal{V}_j$ w.p.p. for some $j < i$. We call this set $\mathcal{U}_i$. It is clear that $\mathcal{U}_i$ is winning w.p.p. for Player x in $\mathcal{G}$, given the induction hypothesis that all $\mathcal{V}_j$ are winning. Then, we find a set of states where Player x wins w.p.p. in $\mathcal{G} \ominus \mathcal{U}_i$. It is clear that $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ is such a set. This set is winning w.p.p. for Player x, because a play starting in $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ either stays in this set and Player x wins with probability 1, or the play leaves $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ and enters $\mathcal{U}_i$ which, as we already know, is winning w.p.p.. We thus take $\mathcal{V}_i$ as the union of $\mathcal{U}_i$ and $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$. We define the sequences formally by

$$\begin{aligned} \mathcal{U}_i &:= \mathit{Force}^{x}(\mathcal{G}, \textstyle\bigcup_{j<i} \mathcal{V}_j), \\ \mathcal{V}_i &:= \mathcal{U}_i \cup \mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i), \end{aligned} \qquad \mathcal{D}_n(\mathcal{G}) := \textstyle\bigcup_{i\in\mathbb{O}} \mathcal{U}_i.$$

By the definitions, for $j < i$ we get $\mathcal{Y}_j \subseteq \mathcal{X}_i \subseteq \mathcal{Y}_i$ and $\mathcal{V}_j \subseteq \mathcal{U}_i \subseteq \mathcal{V}_i$. As in Lemma 2, we can prove that these sequences converge.

Lemma 5. *There are* $\alpha, \beta \in \mathbb{O}$ *such that (i)* $\mathcal{X}_\alpha = \mathcal{Y}_\alpha = \bigcup_{i\in\mathbb{O}} \mathcal{Y}_i$*, (ii)* $\mathcal{C}_n(\mathcal{G}) = \overset{\mathcal{G}}{\neg}\, \mathcal{X}_\alpha$*, (iii)* $\mathcal{U}_\beta = \mathcal{V}_\beta = \bigcup_{i\in\mathbb{O}} \mathcal{V}_i$*, and (iv)* $\mathcal{D}_n(\mathcal{G}) = \mathcal{U}_\beta$*.*

The following lemma shows the correctness of the construction. Recall that we assume that $\mathcal{G}$ is of rank n and that it contains a finite attractor. Let $x = n \bmod 2$.

Lemma 6. *There are memoryless strategies* $f_c^x, f_d^x, \in F_\emptyset^x(\mathcal{G})$ *and* $f_c^{1-x}, f_d^{1-x} \in F_\emptyset^{1-x}(\mathcal{G})$ *such that the following properties hold:*
(i) $\mathcal{C}_n(\mathcal{G}) \subseteq W^x(f_c^x, F_{\mathit{finite}}^{1-x}(\mathcal{G}))(\mathcal{G}, x\text{-}\mathit{Parity}^{=1})$.
(ii) $\overset{\mathcal{G}}{\neg}\, \mathcal{C}_n(\mathcal{G}) \subseteq W^{1-x}(f_c^{1-x}, F_{\mathit{finite}}^x(\mathcal{G}))(\mathcal{G}, (1-x)\text{-}\mathit{Parity}^{>0})$.
(iii) $\mathcal{D}_n(\mathcal{G}) \subseteq W^x(f_d^x, F_{\mathit{finite}}^{1-x}(\mathcal{G}))(\mathcal{G}, x\text{-}\mathit{Parity}^{>0})$.
(iv) $\overset{\mathcal{G}}{\neg}\, \mathcal{D}_n(\mathcal{G}) \subseteq W^{1-x}(f_d^{1-x}, F_{\mathit{finite}}^x(\mathcal{G}))(\mathcal{G}, (1-x)\text{-}\mathit{Parity}^{=1})$.

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

f_c^x. For $n \geq 1$, let α be as defined in Lemma 5. Let $\overline{\mathcal{X}_\alpha} := \overset{\mathcal{G}}{\neg}\, \mathcal{X}_\alpha$ and $\overline{\mathcal{Z}_\alpha} := \overset{\mathcal{G}}{\neg}\, \mathcal{Z}_\alpha$. We know that $\mathcal{C}_n(\mathcal{G}) = \overline{\mathcal{X}_\alpha}$. For a state $s \in \mathcal{C}_n(\mathcal{G})$, we define $f_c^x(s)$ depending on the membership of s in one of the following three partitions of $\mathcal{C}_n(\mathcal{G})$: (1) $\overline{\mathcal{X}_\alpha} \cap \overline{\mathcal{Z}_\alpha}$, (2) $\overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}<n}$, and (3) $\overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}=n}$.

1. $s \in \overline{\mathcal{X}_\alpha} \cap \overline{\mathcal{Z}_\alpha}$. Define $\mathcal{G}' := \mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha$. From Lemma 5, we have that $\mathcal{X}_{\alpha+1} - \mathcal{X}_\alpha = \emptyset$. By the construction of $\mathcal{Y}_i$ we have, for arbitrary i, that $\mathcal{D}^{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i) = \mathcal{Y}_i - \mathcal{X}_i$, and by the construction of $\mathcal{X}_{i+1}$, we have that $\mathcal{Y}_i - \mathcal{X}_i \subseteq \mathcal{X}_{i+1} - \mathcal{X}_i$. By combining these facts we obtain $\mathcal{D}^{n-1}(\mathcal{G}') \subseteq \mathcal{X}_{\alpha+1} - \mathcal{X}_\alpha = \emptyset$. Since $\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i$ does not contain any states of color n (or higher), it follows by the induction hypothesis that there is a memoryless strategy $f_1 \in F_\emptyset^x(\mathcal{G}')$ such that $\overset{\mathcal{G}'}{\neg}\, \mathcal{D}_{n-1}(\mathcal{G}') \subseteq W^x(f_1, F_{\mathit{finite}}^{1-x}(\mathcal{G}'))(\mathcal{G}', x\text{-}\mathit{Parity}^{=1})$. We define $f_c^x(s) := f_1(s)$.
2. $s \in \overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}<n}$. Define $f_c^x(s) := \mathit{force}^x(\mathcal{G} \ominus \mathcal{X}_\alpha, [\mathcal{Z}_\alpha]^{\mathtt{Col}=n})(s)$.
3. $s \in \overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}=n}$. By Lemma 3 we know that $\mathit{Post}_{\mathcal{G}}(s) \cap \overline{\mathcal{X}_\alpha} \neq \emptyset$. Define $f_c^x(s) := \mathit{select}(\mathit{Post}_{\mathcal{G}}(s) \cap \overline{\mathcal{X}_\alpha})$.

Let $f^{1-x} \in F_{\mathit{finite}}^{1-x}(\mathcal{G})$ be a finite-memory strategy for Player $1-x$. We show that $\mathcal{P}_{\mathcal{G},s,f_c^x,f^{1-x}}(x\text{-}\mathit{Parity}) = 1$ for any state $s \in \mathcal{C}_n(\mathcal{G})$. First, we show that, any run $s_0 s_1 \cdots \in \mathit{Runs}(\mathcal{G}, s, f_c^x, f^{1-x})$ will always stay inside $\overline{\mathcal{X}_\alpha}$, i.e., $s_i \in \overline{\mathcal{X}_\alpha}$ for all $i \geq 0$. We use induction on i. The base case follows from $s_0 = s \in \overline{\mathcal{X}_\alpha}$. For the induction step, we assume that $s_i \in \overline{\mathcal{X}_\alpha}$, and show that $s_{i+1} \in \overline{\mathcal{X}_\alpha}$. We consider the following cases:

- $s_i \in [\overline{\mathcal{X}_\alpha}]^{1-x} \cup [\overline{\mathcal{X}_\alpha}]^R$. The result follows since $\overline{\mathcal{X}_\alpha}$ is a $(1-x)$-trap in $\mathcal{G}$ (by Lemma 3).
- $s_i \in [\overline{\mathcal{X}_\alpha} \cap \overline{\mathcal{Z}_\alpha}]^x$. We know that $s_{i+1} = f_1(s_i)$. Since $f_1 \in F_\emptyset^x(\mathcal{G} \ominus \mathcal{X}_\alpha \ominus \mathcal{Z}_\alpha)$ it follows that $s_{i+1} \in \overline{\mathcal{X}_\alpha} \cap \overline{\mathcal{Z}_\alpha}$ and in particular $s_{i+1} \in \overline{\mathcal{X}_\alpha}$.
- $s_i \in [\overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}<n}]^x$. We know that $s_{i+1} = \mathit{force}^x(\mathcal{G} \ominus \mathcal{X}_\alpha, [\mathcal{Z}_\alpha]^{\mathtt{Col}=n})(s_i)$. The result follows by the fact that $\mathit{force}^x(\mathcal{G} \ominus \mathcal{X}_\alpha, [\mathcal{Z}_\alpha]^{\mathtt{Col}=n})$ is a strategy in $\mathcal{G} \ominus \mathcal{X}_\alpha$.
- $s_i \in [\overline{\mathcal{X}_\alpha} \cap [\mathcal{Z}_\alpha]^{\mathtt{Col}=n}]^x$. We have $s_{i+1} \in \mathit{Post}_{\mathcal{G}}(s_i) \cap \overline{\mathcal{X}_\alpha}$ and in particular $s_{i+1} \in \overline{\mathcal{X}_\alpha}$.

Let us again consider a run $\rho \in \mathit{Runs}(\mathcal{G}, s, f^x, f^{1-x})$. 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^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 $\mathcal{T}$. This implies that each state that occurs in B will a.s. be visited infinitely often by ρ. There are two possible cases: (i) There is a configuration $(s_B, m) \in B$ with $\mathtt{Col}(s_B) = n$. Since each state in $\mathcal{G}$ has color at most n, Player x will a.s. win. (ii) There is no configuration $(s_B, m) \in B$ with $\mathtt{Col}(s_B) = n$. This implies that $\{s_B | \ (s_B, m) \in B\} \subseteq \overline{\mathcal{Z}}$, and hence Player x uses the strategy f_1 to win the game.

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

- $s \in \mathcal{X}_\beta - \bigcup_{j<\beta} \mathcal{Y}_j$. Define $f_c^{1-x}(s) := f_1(s) := \mathit{force}^{1-x}(\mathcal{G}, \bigcup_{j<\beta} \mathcal{Y}_j)(s)$.
- $s \in \mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_\beta \ominus \mathcal{Z}_\beta)$. By the induction hypothesis (on n), there is a memoryless strategy $f_2 \in F^{1-x}_{\emptyset}(\mathcal{G})$ of Player $1-x$ such that $s \in W^{1-x}(f_2, F^x_{\mathit{finite}}(\mathcal{G} \ominus \mathcal{X}_\beta \ominus \mathcal{Z}_\beta))(\mathcal{G} \ominus \mathcal{X}_\beta \ominus \mathcal{Z}_\beta, (1-x)\text{-}\mathit{Parity}^{>0})$. Define $f_c^{1-x}(s) := f_2(s)$.

Let $f^x \in F^x_{\mathit{finite}}(\mathcal{G})$ be an arbitrary finite-memory strategy for Player x. We now use induction on i to show that $\mathcal{P}_{\mathcal{G}, s, f_c^{1-x}, f^x}((1-x)\text{-}\mathit{Parity}) > 0$ for any state $s \in \mathcal{Y}_i$. There are three cases:

1. If $s \in \bigcup_{j<i} \mathcal{Y}_j$ then the result follows by the induction hypothesis (on i).
2. If $s \in \mathcal{X}_i - \bigcup_{j<i} \mathcal{Y}_j$ then we know that Player $1-x$, can use f_1 to force the game to $\bigcup_{j<i} \mathcal{Y}_j$ from which she wins w.p.p..
3. If $s \in \mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i)$ 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 $\mathcal{X}_i$; or (ii) there is no such run. In sub-case (i), the run reaches $\mathcal{X}_i$ w.p.p. and then by cases 1 and 2 Player $1-x$ wins w.p.p.. In sub-case (ii), any run stays forever outside $\mathcal{X}_i$. So the game is in effect played on $\mathcal{G} \ominus \mathcal{X}_i$. Notice then that any run from s that is consistent with f^x and f_c^{1-x} stays forever in $\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i$. The reason is that (by Lemma 3) ${}^{\mathcal{G} \ominus \mathcal{X}_i}_{\neg} \mathcal{Z}_i$ is an x-trap in $\mathcal{G} \ominus \mathcal{X}_i$. Since any run remains inside $\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i$, Player $1-x$ wins w.p.p. wrt. $(1-x)$-*Parity* using f_2.

f_d^x. For any state s, let β be the smallest ordinal such that $s \in \mathcal{Y}_\beta$. We define $f_d^x(s)$ by two cases:

- $s \in \mathcal{U}_\beta - \bigcup_{j<\beta} \mathcal{V}_j$. Define $f_d^x(s) := f_1(s) := \mathit{force}^x(\mathcal{G}, \bigcup_{j<\beta} \mathcal{V}_j)(s)$.
- $s \in \mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_\beta)$. By the induction hypothesis (on n), Player x has a winning memoryless strategy f_2 inside $\mathcal{G} \ominus \mathcal{U}_i$. Define $f_d^x(s) := f_2(s)$.

Let $f^{1-x} \in F^{1-x}_{\mathit{all}}(\mathcal{G})$ be an arbitrary strategy for Player $1-x$. We now use induction on i to show that $\mathcal{P}_{\mathcal{G}, s, f_d^x, f^{1-x}}(x\text{-}\mathit{Parity}) > 0$ for any state $s \in \mathcal{V}_i$. There are three cases:

1. If $s \in \bigcup_{j<i} \mathcal{V}_j$ then the result follows by the induction hypothesis (on i).

2. If $s \in \mathcal{U}_i - \bigcup_{j<i} \mathcal{V}_j$ then we know that Player x can use f_1 to force the game to $\bigcup_{j<i} \mathcal{V}_j$ from which she wins w.p.p. by the previous case.
3. If $s \in \mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ then Player x uses f_2. There are now two sub-cases: either (i) there is a run from s consistent with f_d^x and f^{1-x} that reaches $\mathcal{U}_i$; or (ii) there is no such run. In sub-case (i), the run reaches $\mathcal{U}_i$ w.p.p. and then by cases 1 and 2 Player x wins w.p.p.. In sub-case (ii), any run stays forever outside $\mathcal{U}_i$. Hence, Player x wins a.s. wrt. x-*Parity* using f_2.

f_d^{1-x}. By the definition of $\mathcal{U}_i$ we know that $\bigcup_{j<i} \mathcal{V}_j \subseteq \mathcal{U}_i$, and by the definition of $\mathcal{V}_i$ we know that $\mathcal{U}_i \subseteq \mathcal{V}_i$. Thus, $\mathcal{U}_0 \subseteq \mathcal{V}_0 \subseteq \mathcal{U}_1 \subseteq \mathcal{V}_1 \subseteq \cdots$, and hence there is an $\alpha \in \mathbb{O}$ such that $\mathcal{U}_i = \mathcal{V}_i = \mathcal{U}_\alpha$ for all $i \geq \alpha$. This means that $\mathcal{D}_n(\mathcal{G}) = \mathcal{U}_\alpha$ and hence by Lemma 3 we know that $\overset{\mathcal{G}}{\neg}\, \mathcal{D}_n(\mathcal{G})$ is an x-trap. Furthermore, since $\mathcal{V}_\alpha = \mathcal{U}_\alpha \cup \mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_\alpha)$, where the union is disjoint, it follows that $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_\alpha) = \emptyset$ and hence, by the induction hypothesis, Player $1-x$ has a memoryless strategy $f \in F_\emptyset^{1-x}(\mathcal{G})$ that is winning w.p.p. against all finite memory strategies $f^x \in F_{\mathit{finite}}^x(\mathcal{G})$ on all states in $\overset{\mathcal{G}}{\neg}\, \mathcal{U}_\alpha = \overset{\mathcal{G}}{\neg}\, \mathcal{D}_n(\mathcal{G})$. Below, we show that f indeed allows Player $1-x$ to win almost surely.

Fix a finite-memory strategy $f^x \in F_{\mathit{finite}}^x(\mathcal{G})$. Let f^x be induced by a memory structure $\mathcal{M} = (M, m_0, \tau, \mu)$. Consider a run $\rho \in \mathit{Runs}(\mathcal{G}, s, f, f^x)$. Then, ρ will surely stay inside $\mathcal{G} \ominus \mathcal{U}_\alpha$. The reason is that $\overset{\mathcal{G}}{\neg}\, \mathcal{U}_\alpha$ is a trap for Player x by Lemma 3, and that f is a strategy defined inside $\mathcal{G} \ominus \mathcal{U}_\alpha$. Let $\mathcal{T}$ be the transition system induced by $\mathcal{G}$, f^x, and f. As explained in Section 2, ρ will a.s. visit a configuration $(s_A, m) \in B$ for some BSCC B in $\mathcal{T}$. This implies that each configuration in B will a.s. be visited infinitely often by ρ. Let n be the maximal color occurring among the states of B. Then, either (i) $n \bmod 2 = x$ in which case all states inside B are almost sure losing for Player $1-x$; or (ii) $n \bmod 2 = 1-x$ in which case all states inside B are almost sure winning for Player $1-x$. The result follows from the fact that case (i) gives a contradiction since all states in $\overset{\mathcal{G}}{\neg}\, \mathcal{U}_\alpha = \overset{\mathcal{G}}{\neg}\, \mathcal{D}_n(\mathcal{G})$ (including those in B) are winning for Player $1-x$ w.p.p.. Define $f_d^{1-x}(s) := f(s)$.

The following theorem follows immediately from the previous lemmas.

Theorem 1. *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. The scheme for $\mathcal{C}_n$ is adapted from the well-known scheme for non-stochastic games in [27]; in fact, the constructions are equivalent in the case that no probabilistic states are present. Our contribution *to the scheme* is: (1) $\mathcal{C}_n$ is a non-trivial extension of the scheme in [27] to handle probabilistic states; (2) we introduce the alternation between $\mathcal{C}_n$ and $\mathcal{D}_n$; (3) the construction of $\mathcal{D}_n$ is new and has no counterpart in the non-stochastic case of [27].

5 Lossy Channel Systems

A *lossy channel system (LCS)* [7] is a finite-state machine equipped with a finite number of unbounded fifo channels (queues). 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 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 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 $\mathtt{Col} : S \rightarrow \{0, \ldots, n\}$ is the *coloring* function. Each transition $\mathtt{t} \in \mathtt{T}$ is of the form $\mathtt{s} \xrightarrow{\mathtt{op}} \mathtt{s}'$, where $\mathtt{s}, \mathtt{s}' \in \mathtt{S}$ and $\mathtt{op}$ is one of the following three forms: $\mathtt{c!m}$ (send message $\mathtt{m} \in \mathtt{M}$ in channel $\mathtt{c} \in \mathtt{C}$), $\mathtt{c?m}$ (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 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 state $\mathtt{s}$. On the other hand, transitions leaving states of the form $s = (\mathtt{s}, \mathtt{x}, 0)$ model message losses. If $s = (\mathtt{s}, \mathtt{x}, 1), s' = (\mathtt{s}', \mathtt{x}', 0) \in S$, then there is a transition $s \longrightarrow s'$ in the game iff one of the following holds: (i) $\mathtt{s} \xrightarrow{\mathtt{nop}} \mathtt{s}'$ and $\mathtt{x} = \mathtt{x}'$; (ii) $\mathtt{s} \xrightarrow{\mathtt{c!m}} \mathtt{s}'$, $\mathtt{x}'(\mathtt{c}) = \mathtt{x}(\mathtt{c})\mathtt{m}$, and for all $\mathtt{c}' \in \mathtt{C} - \{\mathtt{c}\}$, $\mathtt{x}'(\mathtt{c}') = \mathtt{x}(\mathtt{c}')$; and (iii) $\mathtt{s} \xrightarrow{\mathtt{c?m}} \mathtt{s}'$, $\mathtt{x}(\mathtt{c}) = \mathtt{m}\mathtt{x}'(\mathtt{c})$, and for all $\mathtt{c}' \in \mathtt{C} - \{\mathtt{c}\}$, $\mathtt{x}'(\mathtt{c}') = \mathtt{x}(\mathtt{c}')$. Every state of the form $(\mathtt{s}, \mathtt{x}, 0)$ has at least one successor, namely $(\mathtt{s}, \mathtt{x}, 1)$. If a state $(\mathtt{s}, \mathtt{x}, 1)$ 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. 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 states $\mathtt{x}, \mathtt{x}' : \mathtt{C} \rightarrow \mathtt{M}^*$ 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 game states $s = (\mathtt{s}, \mathtt{x}, i), s' = (\mathtt{s}', \mathtt{x}', i') \in S$ 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 $\mathtt{x}'$ such that $\mathtt{x}' \preceq \mathtt{x}$, there is a transition $s \longrightarrow (\mathtt{s}, \mathtt{x}', 1)$. The probability of random transitions is given by $P((\mathtt{s}, \mathtt{x}, 0), (\mathtt{s}, \mathtt{x}', 1)) = a \cdot \lambda^b \cdot (1 - \lambda)^c$, where a is the number of ways to obtain $\mathtt{x}'$ by losing messages in $\mathtt{x}$, b is the total number of messages needed to be lost in all channels in order to obtain $\mathtt{x}'$ from $\mathtt{x}$, and c is the total number of messages in all channels of $\mathtt{x}'$ (see [1] for details). Finally, for a state $s = (\mathtt{s}, \mathtt{x}, i)$, we define $\mathtt{Col}(s) := \mathtt{Col}(\mathtt{s})$. Notice that the graph of the game is bipartite, in the sense that a state in S^R has only transitions to states in $[S]^{0,1}$, and vice versa.

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.

6 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 2. *The sets of winning states for Players* 0 *and* 1 *are effectively computable as regular languages. Furthermore, from each state, memoryless strategies suffice for the winning player.*

We give the proof in several steps. First, we show that the game induced by an SG-LCS contains a finite attractor (Lemma 7). Then, we show that the scheme in Section 3 for computing winning states wrt. reachability objectives is guaranteed to terminate (Lemma 9). Furthermore, we show that the scheme in Section 4 for computing winning states wrt. a.s. parity objectives is guaranteed to terminate (Lemma 15). Notice that Lemmas 9 and 15 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 16 and 17).

Finite attractor. In [1] 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 $\mathtt{c} \in \mathtt{C}$. In other words, A is given by the set of states 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 7. *$\mathcal{G}$ contains a finite attractor.*

Termination of Reachability Scheme. For a set of states $Q \subseteq S$, we define the *upward closure* of Q by $Q\!\uparrow := \{s \mid \exists s' \in Q.\, s' \preceq s\}$. 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 8. *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 $\alpha \in \mathbb{N}$ such that $Q_i = Q_\alpha$ for all $i \geq \alpha$.*

Now, we can show termination of the reachability scheme.

Lemma 9. *There exists an $\alpha \in \mathbb{N}$ such that $\mathcal{R}_i = \mathcal{R}_\alpha$ for all $i \geq \alpha$.*

Proof. First, we show that $[\mathcal{R}_i - \mathtt{Target}]^R$ is $(\overset{\mathcal{G}}{\neg}\, \mathtt{Target})$-u.c. for all $i \in \mathbb{N}$. We use induction on i. For $i = 0$ the result is trivial since $\mathcal{R}_i - \mathtt{Target} = \emptyset$. For $i > 0$, suppose that $s = (\mathtt{s}, \mathtt{x}, 0) \in [\mathcal{R}_i]^R - \mathtt{Target}$. 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 $s \preceq s'$.

By Lemma 8, there is an $\alpha' \in \mathbb{N}$ such that $[\mathcal{R}_i]^R - \mathtt{Target} = [\mathcal{R}_{\alpha'}]^R - \mathtt{Target}$ for all $i \geq \alpha'$. Since $\mathcal{R}_i \supseteq \mathtt{Target}$ for all $i \geq 0$ it follows that $[\mathcal{R}_i]^R = [\mathcal{R}_{\alpha'}]^R$ for all $i \geq \alpha'$.

Since the graph of $\mathcal{G}$ is bipartite (as explained in Section 5), we have $[Pre_{\mathcal{G}}(\mathcal{R}_i)]^x = [Pre_{\mathcal{G}}([\mathcal{R}_i]^R)]^x$ and $[\widetilde{Pre}_{\mathcal{G}}(\mathcal{R}_i)]^{1-x} = [\widetilde{Pre}_{\mathcal{G}}([\mathcal{R}_i]^R)]^{1-x}$. Since $[\mathcal{R}_i]^R = [\mathcal{R}_{\alpha'}]^R$ for all $i \geq \alpha'$, we thus have $[Pre_{\mathcal{G}}(\mathcal{R}_i)]^x = [Pre_{\mathcal{G}}([\mathcal{R}]^R_{\alpha'})]^x \subseteq \mathcal{R}_{\alpha'+1}$ and $[\widetilde{Pre}_{\mathcal{G}}(\mathcal{R}_i)]^{1-x} = [\widetilde{Pre}_{\mathcal{G}}([\mathcal{R}]^R_{\alpha'})]^{1-x} \subseteq \mathcal{R}_{\alpha'+1}$. It then follows that $\mathcal{R}_i = \mathcal{R}_\alpha$ for all $i \geq \alpha := \alpha' + 1$.

Termination of Parity Scheme. We use several auxiliary lemmas. The following lemma states that sink-freeness is preserved by the reachability scheme.

Lemma 10. *If* `Target` *is sink-free then* $Force^x(\mathcal{G}, \texttt{Target})$ *is sink-free.*

Lemma 11. *If* `Target` *is sink-free then* $[Force^x(\mathcal{G}, \texttt{Target})]^R$ *is upward closed.*

Lemma 12. *Let* $\{Q_i\}_{i\in\mathbb{O}}$ *and* $\{Q'_i\}_{i\in\mathbb{O}}$ *be sequences of sets of states such that (i) Each* Q'_i *is sink-free; (ii)* $Q_i = Q'_i \cup Force^x(\mathcal{G}, \bigcup_{j<i} Q_j)$*; (iii)* Q'_i *and* $Force^x(\mathcal{G}, \bigcup_{j<i} Q_j)$ *are disjoint for all i. Then, there is an* $\alpha \in \mathbb{N}$ *such that* $Q_i = Q_\alpha$ *for all* $i \geq \alpha$.

To apply Lemma 12, we prove the following two lemmas.

Lemma 13. $\mathcal{C}_n(\mathcal{G})$ *is a* $(1-x)$*-trap.*

Proof. $\mathcal{C}_0(\mathcal{G})$ is trivially a $(1-x)$-trap. For $i \geq 1$, the result follows immediately from Lemma 5 and Lemma 3.

Lemma 14. *For any game of rank n both* $\mathcal{C}_n(\mathcal{G})$ *and* $\mathcal{D}_n(\mathcal{G})$ *are sink-free.*

Proof. If $n = 0$, then by definition $\mathcal{C}_n(\mathcal{G}) = \mathcal{D}_n(\mathcal{G}) = S$, which is sink-free by assumption. Next, assume $n \geq 1$. By Lemma 13 we know that $\mathcal{C}_n(\mathcal{G})$ is a $(1-x)$-trap and hence also sink-free. To prove the claim for $\mathcal{D}_n(\mathcal{G})$, we use induction on i and prove that both $\mathcal{U}_i$ and $\mathcal{V}_i$ are sink-free. Assume $\mathcal{U}_j$ and $\mathcal{V}_j$ are sink-free for all $j < i$. Then $\bigcup_{j<i} \mathcal{V}_j$ is sink-free, and hence $\mathcal{U}_i$ is sink-free by Lemma 10. Since $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ and $\mathcal{U}_i$ are sink-free, it follows that $\mathcal{V}_i$ is sink-free.

Now, we apply Lemma 12 to prove that the sequences $\{\mathcal{X}_i\}_{i\in\mathbb{O}}$ and $\{\mathcal{U}_i\}_{i\in\mathbb{O}}$ terminate. First, by Lemma 14 we know that $\mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i)$ is sink-free. We know that $\mathcal{X}_i$ and $\mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i)$ are disjoint since $\mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i) \subseteq^{\mathcal{G}}_{\neg} (\mathcal{X}_i \cup \mathcal{Z}_i)$. Hence, we can apply Lemma 12 with $Q_i = \mathcal{Y}_i$, $Q'_i = \mathcal{D}_{n-1}(\mathcal{G} \ominus \mathcal{X}_i \ominus \mathcal{Z}_i)$, and conclude that $\{\mathcal{Y}_i\}_{i\in\mathbb{O}}$ terminates, and hence $\{\mathcal{X}_i\}_{i\in\mathbb{O}}$ terminates. Second, by Lemma 14 we know that $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ is sink-free. Since $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i) \subseteq^{\mathcal{G}}_{\neg} \mathcal{U}_i$, we know that $\mathcal{U}_i$ and $\mathcal{C}_n(\mathcal{G} \ominus \mathcal{U}_i)$ are disjoint. Hence, we can apply Lemma 12 with $Q_i = \mathcal{V}_i$, $Q'_i = \mathcal{C}_{n-1}(\mathcal{G} \ominus \mathcal{U}_i)$, and conclude that $\{\mathcal{V}_i\}_{i\in\mathbb{O}}$ terminates, and hence $\{\mathcal{U}_i\}_{i\in\mathbb{O}}$ terminates. This gives the following lemma.

Lemma 15. *There is an* $\alpha \in \mathbb{N}$ *such that* $\mathcal{X}_i = \mathcal{X}_\alpha$ *for all* $i \geq \alpha$*. There is a* $\beta \in \mathbb{N}$ *such that* $\mathcal{U}_i = \mathcal{U}_\beta$ *for all* $i \geq \beta$.

Computability. For a given regular set $\mathcal{R}$, the set $Pre_{\mathcal{G}}(\mathcal{R})$ is effectively regular [2], i.e., computable as a regular language. The following lemma then follows from the fact that the other operations used in computing $Force^x(\mathcal{G}, \texttt{Target})$ are those of set complement and union, which are effective for regular languages.

Lemma 16. *If* `Target` *is regular then* $Force^x(\mathcal{G}, \texttt{Target})$ *is effectively regular.*

Lemma 17. *For each n, both* $\mathcal{C}_n(\mathcal{G})$ *and* $\mathcal{D}_n(\mathcal{G})$ *are effectively regular.*

Proof. The set S is regular, and hence $\mathcal{C}_0(\mathcal{G}) = \mathcal{D}_0(\mathcal{G}) = S$ is effectively regular. The result for $n > 0$ follows from Lemma 16 and from the fact that the rest of the operations used to build $\mathcal{C}_n(\mathcal{G})$ and $\mathcal{D}_n(\mathcal{G})$ are those of set complement and union.

Remark. Although we use Higman's lemma for showing termination of our fixpoint computations, our proof differs significantly from the standard ones for well quasi-ordered transition systems [3]. For instance, the generated sets are in general not upward closed wrt. the underlying ordering $\preceq$. Therefore, we need to use the notion of Q-upward closedness for a set of states Q. More importantly, we need to define new (and much more involved) sufficient conditions for the termination of the computations (Lemma 12), and to show that these conditions are satisfied (Lemma 14).

7 Conclusions and Discussion

We have presented a scheme for solving stochastic games with a.s. 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. 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 1 (a) (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 $i > 0$.

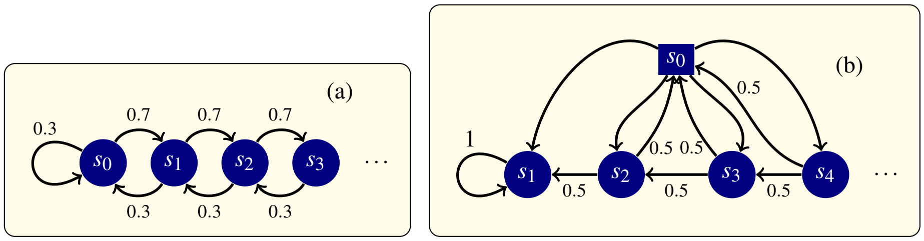

Fig. 1. (a) Finite attractor requirement. (b) Finite strategy requirement.

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 $\mathcal{C}_1(\mathcal{G})$ converges after one iteration with $\mathcal{Z}_i = S, \mathcal{X}_i = \emptyset$ for all i and $\mathcal{C}_1(\mathcal{G}) = S$. 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.

We restrict both players to **finite-memory strategies**. This is a different problem from when arbitrary strategies are allowed (not a sub-problem). In fact, it was shown in [9] that for arbitrary strategies, the problem is undecidable. Figure 1 (b) gives an example of a game graph where the two problems yield different results (see also [9]). Player 1 controls s_0, whereas $s_1, s_2, \ldots$ are random; $\texttt{Col}(s_0) = 0, \texttt{Col}(s_1) = 2, \texttt{Col}(s_i) =$

1 if $i \geq 2$. The transition probabilities are $P(s_1, s_1) = 1$ and $P(s_n, s_{n-1}) = P(s_n, s_0) = \frac{1}{2}$ when $n \geq 2$. Player 1 wants to ensure that the highest color that is seen infinitely often is odd, and thus wants to avoid state s_1 (which has color 2). If the players can use arbitrary strategies, then although player 1 cannot win with probability 1, he can win with a probability arbitrarily close to 1 using an infinite-memory strategy: player 1 goes from s_0 to s_{k+i} when the play visits s_0 for the i'th time. Then player 1 wins with probability $\prod_{i=1}^{\infty}(1 - 2^{-k-i+1})$, which can be made arbitrarily close to 1 for sufficiently large k. In particular, player 0 does not win a.s. in this case. On the other hand, if the players are limited to finite-memory strategies, then no matter what strategy player 1 uses, the play visits s_1 infinitely often with probability 1, so player 0 wins almost surely; this is also what our algorithm computes.

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 [6].

References

1. 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.
2. P. A. Abdulla, A. Bouajjani, and J. d'Orso. Deciding monotonic games. In *CSL*, volume 2803 of *LNCS*, pages 1–14, 2003.
3. P. A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160:109–127, 2000.
4. P. A. Abdulla, L. Clemente, R. Mayr, and S. Sandberg. Stochastic parity games on lossy channel systems. Technical Report EDI-INF-RR-1416, University of Edinburgh, 2013. http://arxiv.org/abs/1305.5228 and http://www.inf.ed.ac.uk/publications/report/1416.html.
5. 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.
6. P. A. Abdulla, N. B. Henda, and R. Mayr. Decisive Markov chains. *Logical Methods in Computer Science*, 3, 2007.
7. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. In *LICS*, pages 160–170, 1993.
8. P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *FOSSACS*, volume 2620 of *LNCS*, pages 39–53, 2003.
9. 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.
10. N. Bertrand and Ph. Schnoebelen. Model checking lossy channels systems is probably decidable. In *FOSSACS*, volume 2620 of *LNCS*, pages 120–135, 2003.
11. N. Bertrand and Ph. Schnoebelen. Solving stochastic büchi games on infinite arenas with a finite attractor. In *Proceedings of the 11th International Workshop on Quantitative Aspects of Programming Languages (QAPl'13)*, Roma, Italy, March 2013. To appear.
12. P. Billingsley. *Probability and Measure*. Wiley, New York, NY, 1986. Second Edition.
13. D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 2(5):323–342, April 1983.
14. K. Chatterjee, L. de Alfaro, and T. Henzinger. Strategy improvement for concurrent reachability games. In *QEST*, pages 291–300. IEEE Computer Society Press, 2006.
15. K. Chatterjee, M. Jurdziński, and T. Henzinger. Simple stochastic parity games. In *CSL*, volume 2803 of *LNCS*, pages 100–113. Springer Verlag, 2003.
16. E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, Dec. 1999.
17. A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, February 1992.
18. L. de Alfaro and T. Henzinger. Concurrent omega-regular games. In *LICS*, pages 141–156, Washington - Brussels - Tokyo, June 2000. IEEE.
19. L. de Alfaro, T. Henzinger, and O. Kupferman. Concurrent reachability games. In *FOCS*, pages 564–575. IEEE Computer Society Press, 1998.
20. K. Etessami, D. Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. In *ICALP*, volume 5125 of *LNCS*. Springer, 2008.
21. K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *ICALP*, volume 3580 of *LNCS*, pages 891–903, 2005.
22. K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. *LMCS*, 4, 2008.
23. A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *ICALP*, volume 2719 of *LNCS*, pages 1008–1021, 2003.
24. L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, October 1953.
25. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338, 1985.
26. 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.
27. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS*, 200:135–183, 1998.