Stochastic Games with Lossy Channels

Parosh Aziz Abdulla
Uppsala University, Sweden.
parosh@it.uu.se

Noomene Ben Henda
Uppsala University, Sweden.
Noomene.BenHenda@it.uu.se

Luca de Alfaro
University of California, USA.
luca@soe.ucsc.edu

Richard Mayr
NC State University, USA.
mayr@csc.ncsu.edu

Sven Sandberg
Uppsala University, Sweden.
svens@it.uu.se

Abstract

We consider turn-based stochastic games on infinite graphs induced by game probabilistic lossy channel systems (GPLCS), the game version of probabilistic lossy channel systems (PLCS). We study games with Büchi (repeated reachability) objectives and almost-sure winning conditions. These games are pure memoryless determined and, under the assumption that the target set is regular, a symbolic representation of the set of winning states for each player can be effectively constructed. Thus, turn-based stochastic games on GPLCS are decidable. This generalizes the decidability result for PLCS-induced Markov decision processes in [10].

1 Introduction

Background. It is natural to model a reactive system as a 2-player game between the "controller" or player 0, who makes nondeterministic choices of the system, and the "environment" or player 1, who provides malicious inputs to the system. In this model, each state belongs to one of the players, who selects an outgoing transition that determines the next state. Starting in some initial state, the players jointly construct an infinite sequence of states called a *run*. The winning condition is specified as a predicate on runs. Verifying properties of the system corresponds to finding the winner of the game, where the winning condition depends on the property to check.

Systems that have a probabilistic component give rise to stochastic games. These are games where some states belong to "player random", who selects the next state according to a pre-defined probability distribution. Randomness is useful to model stochastic loss of information such as unreliable communication, as well as randomized algorithms.

Previous work on algorithms for stochastic games has mostly focused on finite-state systems (see, e.g., [26, 14, 16, 12]). However, many systems can only be faithfully modeled using infinitely many states. A lot of recent research has therefore been concerned with probabilistic infinite-state models. Probabilistic versions of lossy channel systems [11, 7] and pushdown automata [18, 19] use unbounded queues and stacks, respectively. Probabilistic Petri nets [4] model systems with an unbounded number of processes which run in parallel. The recently introduced Noisy Turing machines [8] model computer memories subject to stochastic errors.

We consider infinite-state stochastic games induced by lossy channel systems (LCS) [1, 10, 24]. LCS consist of finite-state control parts and unbounded channels (queues), i.e., automata where transitions are labeled by send and receive operations. They can model communication protocols such as the sliding window protocol and HDLC [6], where the communication medium is unreliable. In this paper, we introduce *game probabilistic LCS (GPLCS)*. GPLCS are *probabilistic* in the sense that the channels may randomly lose messages; and they are *games* in the sense that the next transition in the control part is selected by one of the players, depending on the current state. We can use player 0 to model nondeterminism in a communication protocol and player 1 to model a malicious cracker trying to break the protocol.

We consider Büchi (repeated reachability) objectives with almost-sure winning conditions. In other words, the goal for player 0 is to guarantee that with probability one, a given set of target states is visited infinitely many times. In the example of the malicious cracker, this corresponds to checking that the system can respond in such a way that it always eventually returns to a "ready state" with probability 1, no matter how the cracker acts.

Related Work. The work closest to ours is [9] where the authors consider the same model. They study GPLCS with simple reachability objectives and different winning conditions; i.e., almost-sure, with positive probability, etc. However, they do not consider GPLCS with Büchi objectives. Previous work on LCS considers several types of nondeterministic [20, 6] and probabilistic systems (Markov chains) [22, 1, 24], as well as Markov decision processes [10] and non-stochastic games [3]. Of these, the work most closely related to ours is [10], which concerns LCS where messages are lost probabilistically and control transitions are taken nondeterministically (i.e., PLCS-induced Markov decision processes). This is a special case of our model in the sense that the game is restricted to only one player. It was shown in [10] that such 1-player Büchi-games are decidable (while coBüchi-games are undecidable). We generalize the decidability result of [10] for PLCS-induced Markov decision processes to 2-player stochastic games. The scheme presented in [10] also differs from ours in the fact that the target set is defined by control-states, while we consider more general *regular sets*. Thus our result is not a direct generalization of [10].

Stochastic games on infinite-state probabilistic recursive systems were studied in [18, 19]. However, recursive systems are incomparable to the GPLCS model considered in this paper.

In [3], a model similar to ours is studied. It differs in that the system is not probabilistic, and instead one of the players controls message losses. For this model, [3] proves that safety games are decidable and parity games (which generalize Büchi games) are undecidable.

Two-player *concurrent* (but non-stochastic) games with infinite state spaces are studied in [17]. Concurrency means that the two players independently and simultaneously select actions, and the next state is determined by the combination of the actions and the current state. [17] describes schemes for computing winning sets and strategies for Büchi games (as well as reachability games and some more general games). The article characterizes classes of games where the schemes terminate, based on properties of certain equivalence relations on states. However, this approach does not work for GPLCS (not even for non-probabilistic LCS), since LCS do not satisfy the necessary preconditions. Unlike the process classes studied in [17], LCS do not have a finite index w.r.t. the equivalences considered in [17].

In [28], a scheme is given to solve non-stochastic *parity* games on infinite state spaces of arbitrary cardinality. The parity condition is more general than the Büchi condition, so the scheme applies to Büchi games too. However, stochastic games are not considered. In fact, if our scheme is instantiated on the special case of non-stochastic Büchi games, it will coincide with the scheme in [28]. Furthermore, [28] does not suggest any class of infinite-state systems for which termination is guaranteed.

Our algorithms are related to the algorithms presented in [16, 15] for solving concurrent games with respect to probability-1 ω-regular properties. However, the proofs in [16, 15] apply only to finite-state games; we will need to develop entirely new arguments to prove the correctness of our approach for GPLCS.

Contribution. We prove that the almost-sure Büchi-GPLCS problem is decidable: we can compute symbolic representations of the winning sets and winning strategies for both players. The symbolic representations are based on regular expressions, and the result holds under the assumption that the set of target states is also regular. The winning strategies are pure memoryless, i.e., the next state depends only on the current state and is not selected probabilistically. Our result generalizes the decidability result for PLCS-induced Markov decision processes (i.e., 1-player games) in [10].

We now give an overview of our method. First, we give a scheme to compute the winning sets in simple reachability games, where the goal of player 0 is to reach a regular set of target states with a *positive probability*. (Note that this differs from almost-sure reachability.) The scheme is based on backward reachability. We prove that the scheme terminates for GPLCS, show how to instantiate it for GPLCS, and prove its correctness.

Next, we give a scheme to construct the winning sets in almost-sure Büchi-games, using the scheme for reachability games as a subroutine. The scheme constructs bigger and bigger sets of states winning for player 1, denoted $X_0 \subseteq X_1 \subseteq \cdots$. The set X_0 is empty and X_1 consists of those states where player 1 can force the game to never reach the target set, with a positive probability. For $i \geq 1$, X_{i+1} consists of three parts. The first part is just X_i, where player 1 wins by induction hypothesis. The second part does not include any target states, and player 0 can choose between two ways to lose. Either the game stays in the second part and thus never reaches the target set, or the game reaches X_i and player 0 loses by induction hypothesis. The third part consists of those states from which player 1 can force the game with a positive probability to the first or second part.

We prove that this scheme terminates for GPLCS (i.e., $\exists i.X_i = X_{i+1}$). We instantiate the scheme for GPLCS using *regular state languages* to represent the infinite sets. Then we prove that player 1 wins when the game starts in $\bigcup_{i\in\mathbb{N}} X_i$, and that player 0 wins otherwise.

Outline. In Section 2, we define stochastic games. In Section 3, we describe GPLCS and show how they induce an infinite-state stochastic game. In Section 4, we show how to construct the winning sets in simple reachability games on GPLCS. In Section 5, we show how to construct the winning sets in Büchi games on GPLCS. Due to space limitations, some proofs are in the appendix; however, the intuitions are given in the main text.

2 Preliminaries

We use $\mathbb{R}, \mathbb{N}$ for the real and natural numbers. If X is a set then X^* and X^ω denote the sets of finite and infinite sequences over X, respectively. The empty word is denoted by ε. For partial functions $f, g : X \rightharpoonup Y$ which have the same value when both are defined, we use $f \cup g$ to denote the smallest function that extends both f and g.

A *probability distribution* on a countable set X is a function $f : X \to [0, 1]$ such that $\sum_{x\in X} f(x) = 1$. 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.

Turn-Based Stochastic Games. A *turn-based stochastic game* (or a *game* for short) is a tuple $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ where:

- S is a countable 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 $(s, s') \in \longrightarrow$. Let $\mathrm{Post}(s) := \{s' : s \longrightarrow s'\}$ denote the set of *successors* of s and extend it to sets $Q \subseteq S$ of states by $\mathrm{Post}(Q) := \bigcup_{s\in Q} \mathrm{Post}(s)$. We assume that games are deadlock-free, i.e., each state has at least one successor ($\forall s \in S.\mathrm{Post}(s) \neq \emptyset$).
- 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 $\forall s \in S^R. \sum_{s'\in S} P(s, s') = 1$. Note that for any given state $s \in S^R$, $P(s, \cdot)$ is a probability distribution over $\mathrm{Post}(s)$.

For any set $Q \subseteq S$ of states, we let $\overline{Q} := S - Q$ denote its complement. We define $[Q]^R := Q \cap S^R$, $[Q]^0 := Q \cap S^0$, $[Q]^1 := Q \cap S^1$, and $[Q]^{01} := Q \cap (S^0 \cup S^1)$.

A *run* ρ in a game is an infinite sequence $s_0 s_1 \cdots$ of states s.t. $s_i \longrightarrow s_{i+1}$ for all $i \geq 0$. We use $\rho(i)$ to denote 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 $i : 0 \leq i < n$. For any $Q \subseteq S$, we use Π_Q to denote the set of paths that end in some state in Q.

Informally, the two players 0 and 1 construct an infinite run $s_0 s_1 \cdots$, starting in some initial state $s_0 \in S$. The state s_{i+1} is chosen as a successor of s_i. Player 0 chooses the successor s_{i+1} if $s_i \in S^0$, player 1 chooses s_{i+1} if $s_i \in S^1$, and the successor s_{i+1} is chosen randomly according to the probability distribution $P(s_i, \cdot)$ if $s_i \in S^R$.

Strategies. For $\sigma \in \{0, 1\}$, a *strategy* of player σ is a partial function $f^\sigma : \Pi_{S^\sigma} \rightharpoonup S$ s.t. $s_n \longrightarrow f^\sigma(s_0 \cdots s_n)$ if f^σ is defined. The strategy f^σ prescribes for player σ the next move, given the current prefix of the run. We say that f^σ is total if it is defined for every $\pi \in \Pi_{S^\sigma}$.

A strategy f^σ of player σ is *memoryless* if the next state only depends on the current state and not on the previous history of the game, i.e., for any path $s_0 \cdots s_k \in \Pi_{S^\sigma}$, we have $f^\sigma(s_0 \cdots s_k) = f^\sigma(s_k)$. A memoryless strategy of player σ can be regarded simply as a function $f^\sigma : S^\sigma \rightharpoonup S$, such that $s \longrightarrow f^\sigma(s)$ whenever f^σ is defined.

Consider two total strategies f^0 and f^1 of player 0 and 1. A path $\pi = s_0 \cdots s_n$ in $\mathcal{G}$ is said to be *consistent* with f^0 and f^1 if the following holds. For all $0 \leq i \leq n-1$, $s_i \in S^0$ implies $f^0(s_0 \cdots s_i) = s_{i+1}$ and $s_i \in S^1$ implies $f^1(s_0 \cdots s_i) = s_{i+1}$. We define similarly *consistent* runs. In the sequel, whenever the strategies are known from the context, we assume that all mentioned paths and runs are consistent with them.

Probability Measures. We use the standard definition of the probability measure for a set of runs [23]. First, we define the measure for total strategies, and then extend it to general (partial) strategies. We let $\Omega^s = sS^\omega$ denote the set of all infinite sequences of states starting from s. Consider a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$, an initial state s, and total strategies f^0 and f^1 of player 0 and 1. For a measurable set $\mathfrak{R} \subseteq \Omega^s$, we define $\mathcal{P}^s_{f^0,f^1}(\mathfrak{R})$ to be the probability measure of $\mathfrak{R}$ under the strategies f^0, f^1. It is well-known that this measure is well-defined [23]. When the state s is known from context, we drop the superscript and write $\mathcal{P}_{f^0,f^1}(\mathfrak{R})$. For (partial) strategies f^0 and f^1 of player 0 and 1, $\sim\in\{<,\leq,=,\geq,>\}$, and any measurable set $\mathfrak{R}\subseteq\Omega^s$, we define $\mathcal{P}^s_{f^0,f^1}(\mathfrak{R})\sim x$ iff $\mathcal{P}^s_{g^0,g^1}(\mathfrak{R})\sim x$ for all total strategies g^0 and g^1 which are extensions of f^0 resp. f^1. For a single strategy f^σ of player σ, we define $\mathcal{P}^s_{f^\sigma}(\mathfrak{R})\sim x$ iff $\mathcal{P}^s_{f^0,f^1}(\mathfrak{R})\sim x$ for all strategies $f^{1-\sigma}$ of player $(1-\sigma)$. If $\mathcal{P}_{f^0,f^1}(\mathfrak{R})=1$, then we say that $\mathfrak{R}$ happens *almost surely* under the strategies f^0,f^1.

We assume familiarity with the syntax and semantics of the temporal logic *CTL** (see, e.g., [13]). We use $(s\models\varphi)$ to denote the set of runs starting in s that satisfy the *CTL** path-formula φ. We use $\mathcal{P}_{f^0,f^1}(s\models\varphi)$ to denote the measure of $(s\models\varphi)$ under strategies f^0,f^1, i.e., we measure the probability of those runs which start in s, are consistent with f^0,f^1 and satisfy the path-formula φ. This set is measurable by [27].

Traps. For a player $\sigma\in\{0,1\}$ and a set $Q\subseteq S$ of states, we say that Q is a σ-*trap* if player $(1-\sigma)$ has a strategy that forces all runs to stay inside Q. Formally, all successors of states in $[Q]^\sigma\cup[Q]^R$ are in Q and every state in $[Q]^{1-\sigma}$ has some successor in Q.

Winning Conditions. Our main result considers *Büchi*[1] objectives: player 0 wants to visit a given set $F\subseteq S$ infinitely many times. We consider games with *almost-sure* winning condition. More precisely, given an initial state $s\in S$, we want to check whether player 0 has a strategy f^0 such that for all strategies f^1 of player 1, it is the case that $\mathcal{P}_{f^0,f^1}(s\models\Box\Diamond F)=1$.

Determinacy and Solvability. A game is said to be *determined* if, from every state, one of the players has a strategy that wins against all strategies of the opponent. Notice that determinacy implies that there is a partitioning W^0,W^1 of S, such that players 0 and 1 have winning strategies from W^0 and W^1, respectively. A game is *memoryless determined* if it is determined and there are winning strategies which are memoryless. By *solving* a determined game, we mean giving an algorithm to check, for any state $s\in S$, whether $s\in W^0$ or $s\in W^1$.

3 Game Probabilistic Lossy Channel Systems (GPLCS)

A *lossy channel system (LCS)* [6] is a finite-state automaton 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. *Probabilistic lossy channel system (PLCS)* [11, 7, 4] define a probabilistic model for message losses. The standard model assumes that each individual message is lost independently with probability λ in every step, where $\lambda>0$ is a parameter of the system.

We consider *game probabilistic LCS (GPLCS)*, the 2-player game extension of PLCS. The set of states is partitioned into states belonging to player 0 and 1, and the transitions are controlled by the players. The player who owns the current control-state chooses an enabled outgoing transition. However, message losses occur randomly. While our definition of GPLCS (see below) assumes the same model of independent message loss as in [11, 7, 4], this is not necessary for our results. We only require the existence of a finite attractor, in the sense described in Section 5. In fact, many other probabilistic message loss models (e.g., burst disturbances, where groups of messages in close proximity are more often affected) satisfy this attractor condition [5].

The players have conflicting goals: player 0 wants to reach a given set of states infinitely often, and player 1 wants to visit it at most finitely many times. This is called a Büchi objective.

Formally, a GPLCS is a tuple $\mathcal{L}=\left(\mathtt{S},\mathtt{S}^0,\mathtt{S}^1,\mathtt{C},\mathtt{M},\mathtt{T},\lambda\right)$ 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*, and $0<\lambda<1$ is the *loss rate*. 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 $\mathtt{c!m}$ (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).

A GPLCS $\mathcal{L}=\left(\mathtt{S},\mathtt{S}^0,\mathtt{S}^1,\mathtt{C},\mathtt{M},\mathtt{T},\lambda\right)$ induces a game $\mathcal{G}=(S,S^0,S^1,S^R,\longrightarrow,P)$, 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: $S^\sigma=\mathtt{S}^\sigma\times(\mathtt{M}^*)^{\mathtt{C}}\times\{1\}$. 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:

- $\mathtt{s}\xrightarrow{\mathtt{nop}}\mathtt{s}'$ and $\mathtt{x}=\mathtt{x}'$;
- $\mathtt{s}\xrightarrow{\mathtt{c!m}}\mathtt{s}'$, $\mathtt{x}'(\mathtt{c})=\mathtt{x}(\mathtt{c})\mathtt{m}$, and for all $\mathtt{c}'\in\mathtt{C}-\{\mathtt{c}\}$,

[1]Also known as *repeated reachability*or ω-*regular* game

$\mathtt{x}'(\mathtt{c}') = \mathtt{x}(\mathtt{c}')$;

- $\mathtt{s} \xrightarrow{\mathtt{c?m}} \mathtt{s}'$, $\mathtt{x}(\mathtt{c}) = \mathtt{m}\mathtt{x}'(\mathtt{c})$, and for all $\mathtt{c}' \in \mathtt{C} - \{\mathtt{c}\}$, $\mathtt{x}'(\mathtt{c}') = \mathtt{x}(\mathtt{c}')$.

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 $\mathtt{x}, \mathtt{x}' : \mathtt{C} \to \mathtt{M}^*$ 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 lost in all channels, and c is the total number of messages in all channels of $\mathtt{x}'$. See [7] for details.

Every state on 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 avoid deadlocks. Intuitively, this means that the run stays in the same control state and only loses messages.

Observe that the game is *bipartite*: every transition goes from a player state to a probabilistic state or the other way around, i.e., $\longrightarrow \subseteq ((S^0 \cup S^1) \times S^R) \cup (S^R \times (S^0 \cup S^1))$.

Problem Statement. We study the problem Büchi-GPLCS, defined as follows. The game graph is induced by a GPLCS; and we consider the almost-sure Büchi objective: player 0 wants to ensure that a given target set is visited infinitely often with probability one.

4 Reachability Games on GPLCS

We consider the reachability game where the winning condition is to reach a given target set with positive probability. Reachability games on GPLCS (with this and various other winning conditions) have been studied in [9], where the winning sets are expressed in terms of the target set in a variant of the μ-calculus.

Nevertheless, we give below a more ad-hoc scheme for computing the winning set, in order to keep the article self-contained. Furthermore, many definitions and some more detailed results on the structure of the winning sets and strategies will be needed in the following section on Büchi-games.

We give a scheme for characterizing sets of states from which a player can, with a positive probability, force the game into a given set of target states, while preserving a given invariant. We show that the scheme always terminates for GPLCS, and then give a symbolic representation of the winning sets, based on regular languages. The symbolic representation is valid under the assumption that the set of target states is also regular. Finally, we show correctness of the construction by describing the winning strategies. In fact, we show that if a player can win, then a *memoryless* strategy is sufficient to win.

Scheme. Fix a game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ and two sets of states $F, I \subseteq S$, called the *target* and *invariant* sets, respectively. For a player $\sigma \in \{0,1\}$, we give a scheme for constructing the set $\mathrm{Force}^\sigma(I, F)$ of states where player σ can, with a positive probability, force the run to eventually reach F, while also preserving the property that the run will always remain within I (i.e., states outside I are not visited before F).

The idea of the scheme is to perform backward reachability analysis using the basic operations Pre^σ and $\widetilde{\mathrm{Pre}}^\sigma$, defined as follows. Given $\sigma \in \{0, 1, R\}$ and a set $Q \subseteq S$ of states, let $\mathrm{Pre}^\sigma(Q) := \{s \in S^\sigma : \exists s' \in Q. s \longrightarrow s'\}$ denote the set of states of player σ where it is possible to go to Q in the next step. Define $\widetilde{\mathrm{Pre}}^\sigma(Q) := S^\sigma - \mathrm{Pre}^\sigma(\overline{Q})$ to be the set of states where player σ cannot avoid going to Q in the next step.

The construction is inductive. For $\sigma \in \{0, 1\}$, we define two sequences $\{D_i\}_{i \in \mathbb{N}} : D_0 \subseteq D_1 \subseteq \cdots$ and $\{E_i\}_{i \in \mathbb{N}} : E_0 \subseteq E_1 \subseteq \cdots$ of sets of states as follows:

$$D_0 := [F]^R \cap I \quad D_{i+1} := \left(D_i \cup \mathrm{Pre}^R(E_i)\right) \cap I$$
$$E_0 := [F]^{01} \cap I$$
$$E_{i+1} := \left(E_i \cup \mathrm{Pre}^\sigma(D_i) \cup \widetilde{\mathrm{Pre}}^{1-\sigma}(D_i)\right) \cap I.$$

We let $\mathrm{Force}^\sigma(I, F) := \bigcup_{i \geq 0} D_i \cup E_i$. Intuitively, the set D_i contains those states in S^R from which player σ can force the game to F with positive probability (while remaining in I) within i steps. The set E_i contains the states in $S^0 \cup S^1$ satisfying the same property[1] .

Below, we instantiate the above described scheme for GPLCS. In the rest of this section, we consider the game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ induced by a GPLCS $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda)$.

Termination. We recall from [21] that the relation $\preceq$ is a *well quasi-ordering*, i.e., for each infinite sequence $w_0, w_1, w_2, \ldots$ of words over $\mathtt{M}$, there are $j < k$ such that $w_j \preceq w_k$. A set $U \subseteq \mathtt{M}^*$ is said to be *upward closed* if $w \in U$ implies that $w' \in U$ for each $w' \succeq w$.

[1] We remark that it would be possible to define only one sequence, not separating player states from random states. In later proofs, it will be technically convenient to have the sequence $\{D_i\}_{i \in \mathbb{N}}$ defined, since $\{D_i\}_{i \in \mathbb{N}}$ has properties not shared by $\{E_i\}_{i \in \mathbb{N}}$, which are used to show termination.

A *channel language* L is a mapping from $\mathtt{C}$ to $2^{\mathtt{M}^*}$. In other words, L maps each channel to a language over $\mathtt{M}$. We say that L is *upward closed* resp. *regular* if $L(\mathtt{c})$ is upward closed resp. regular for each $\mathtt{c} \in \mathtt{C}$. A *state language* L is of the form $(\mathtt{s}, L')$ where $\mathtt{s} \in \mathtt{S}$ and L' is a channel language. We say that L is upward closed (regular) if L' is upward closed (regular). We generalize the definitions above to finite sets M of state languages by M upward closed (regular) if each $L \in M$ is upward closed (regular). The well quasi-ordering property carries directly over from words (mentioned earlier) to our finite sets of state languages (when required to hold for every control-state).

To prove termination of the scheme, we show that sets D_i are "almost" upward closed in the sense that they are closely related to other sets which are upward closed. More precisely, we consider the sequence $D'_0 \subseteq D'_1 \subseteq \cdots$ of sets of states where $D'_0 := [F]^R$ and $D'_{i+1} := \mathrm{Pre}^R(E_i)$. Since $\mathrm{Pre}^R(Q)$ is upward closed for any set Q of states, it follows that D'_i is upward closed for each $i > 0$. Upward closedness, together with the well quasi-ordering of $\preceq$, implies that there is a j such that $D'_j = D'_{j+1} = \cdots$. We also observe that $D_i = (D'_0 \cup D'_1 \cup \cdots \cup D'_i) \cap I$. This means that $D_{j+1} = D_j$ and consequently $E_{j+2} = E_{j+1}$. Hence, we have the following lemma.

Lemma 4.1 *For any GPLCS and sets $F, I \subseteq S$ of states, the sequences $\{D_i\}_{i\in\mathbb{N}}$ and $\{E_i\}_{i\in\mathbb{N}}$ converge.*

Forms of Winning Sets. The above termination argument relied on upward closedness of the sets D'_i. In fact, we can derive more information about the structure of the winning sets for games induced by GPLCS. Assuming that the sets F and I are regular state languages, it follows that each set D_i or E_i is also a regular state language. This follows from the fact that regular state languages are closed under the application of Pre^σ and the Boolean operations. Since the scheme terminates (by Lemma 4.1), the winning set $Q := \mathrm{Force}^\sigma(I, F)$ is also regular. Furthermore, if I and F are upward closed then $[Q]^R$ is also upward closed. This follows from the fact that $\mathrm{Pre}^R(Q)$ is upward closed for any set Q and that the class of upward closed sets is closed under intersection and union. We summarize these properties as properties (1)–(2) of the following lemma. (Properties (2)–(5) are not needed until the next section).

Lemma 4.2 *Let $Q = \mathit{Force}^\sigma(I, F)$. Then:*

(1) If F and I are regular then Q is regular.

(2) If F and I are upward closed then $[Q]^R$ is upward closed.

(3) Let $s \in I - Q$. If $s \in S^\sigma \cup S^R$, then $\mathit{Post}(s) \subseteq \overline{Q}$. If $s \in S^{1-\sigma}$, then $\mathit{Post}(s) \cap \overline{Q} \neq \emptyset$.

(4) $\mathit{Force}^\sigma(Q, F) = Q$.

(5) $\overline{\mathit{Force}^\sigma(S, F)}$ is a σ-trap.

Correctness. First, we describe a partial memoryless winning strategy $\underline{\mathrm{force}}^\sigma(I, F)$ for player σ from the states in $[\mathrm{Force}^\sigma(I, F)]^\sigma$. Recall that a memoryless strategy can simply be described as a function that assigns one successor to each state. We define a sequence $e_0 \subseteq e_1 \subseteq e_2 \subseteq \cdots$ of strategies for player σ. Let $e_0 := \emptyset$ and define e_{i+1} as follows:

- If $e_i(s)$ is defined then $e_{i+1}(s) := e_i(s)$.
- If $e_i(s)$ is undefined and $s \in [E_{i+1} - E_i]^\sigma$ then $e_{i+1}(s) := \mathit{select}(\mathrm{Post}(s) \cap D_i)$.

Let $\underline{\mathrm{force}}^\sigma(I, F) := \bigcup_{i\geq 0} e_i$. From the definitions, we derive the following lemma.

Lemma 4.3 *In any GPLCS, for any $I, F \subseteq S$, $\sigma \in \{0, 1\}$, and $s \in \mathit{Force}^\sigma(I, F)$, there exists an $\epsilon_s > 0$ such that $\mathcal{P}_{\underline{\mathit{force}}^\sigma(I,F)}(s \models \Diamond F) \geq \epsilon_s$.* [2]

Proof. We recall the construction of the force sets and use induction on i to prove that $\forall i \in \mathbb{N}$ and for any state $s \in (D_i \cup E_i)$ the following holds: There exists an $\epsilon_s > 0$ such that for any extension f^σ of the $\underline{\mathrm{force}}^\sigma(I, F)$ and any strategy $f^{1-\sigma}$ of the opponent, $\mathcal{P}_{\underline{\mathrm{force}}^\sigma(I,F), f^{1-\sigma}}(s \models \Diamond Q) \geq \epsilon_s$. Observe that $\forall i. D_i \cap E_i = \emptyset$.

The base case $s \in (D_0 \cup E_0) \subseteq F$ holds trivially (take $\epsilon_s := 1$).

Now assume that the claim holds for $i \in \mathbb{N}$. Consider s to be in $D_{i+1} \cup E_{i+1}$. In the case $s \in (D_i \cup E_i)$ the claim already holds by induction hypothesis. The remaining cases are described below.

Case $s \in D_{i+1} - D_i$: This implies that $s \in \mathrm{Pre}^R(E_i)$. Thus there is a state $s' \in E_i$ such that $s \longrightarrow s'$ and $\epsilon_{s'} > 0$ by induction hypothesis. We define $\epsilon_s := P(s, s') * \epsilon_{s'} > 0$.

Case $s \in E_{i+1} - E_i$: This implies one of the following two cases.

- If $s \in [S]^\sigma$ then $s \in \mathrm{Pre}^\sigma(D_i)$. Thus there is a state $s' \in D_i$ which is chosen as successor state to s by the $\underline{\mathrm{force}}^\sigma(I, F)$ strategy, i.e., $s \longrightarrow s'$ and $s' = \underline{\mathrm{force}}^\sigma(I, F)(s)$. By induction hypothesis $\epsilon_{s'} > 0$. So we obtain $\epsilon_s := \epsilon_{\underline{\mathrm{force}}^\sigma(I,Q)(s)} = \epsilon_{s'} > 0$.

[2] The weaker statement, i.e., $\mathcal{P}_{\underline{\mathrm{force}}^\sigma(I,F)}(s \models \Diamond F) > 0$, suffices for correctness. However, this stronger version is needed in the sequel.

- If $s \in [S]^{1-\sigma}$ then $s \in \widetilde{\text{Pre}}^{1-\sigma}(D_i)$. It follows that $\text{Post}(s) \subseteq D_i$. The set $\text{Post}(s)$ is finite, since the system is finitely branching. Furthermore, by induction hypothesis, $\epsilon_{s'} > 0$ for all $s' \in D_i$. Thus we obtain $\epsilon_s := \min_{s' \in \text{Post}(s)}(\epsilon_{s'}) > 0$.

The main result follows since for any $s \in \text{Force}^\sigma(I, Q)$, there exists a finite minimal $i \in \mathbb{N}$ such that $s \in (D_i \cup E_i)$. □

In the sequel, we use $\text{Force}^\sigma(F)$ to denote $\text{Force}^\sigma(S, F)$, i.e., we do not mention I in case it is equal to S. We define $\underline{\text{force}}^\sigma(F)$ analogously.

5 Büchi-Games on GPLCS

In this section we consider the Büchi-GPLCS problem. We give a scheme for characterizing the winning sets in almost-sure Büchi games, and then instantiate the scheme for GPLCS. In a similar manner to Section 4, we first show that the scheme always terminates for GPLCS, and then describe the winning sets using a symbolic representation based on regular languages. Again, the symbolic representation is valid under the assumption that the set of final states is also regular. We show the correctness of the construction by describing the memoryless winning strategies. Observe that this implies that Büchi-GPLCS are memoryless determined and solvable. Throughout this section, we fix a GPLCS $\mathcal{L} = \left(\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda\right)$ and the induced game $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$. Take $F \subseteq S$; we consider the Büchi goal for player 0 consisting in visiting F infinitely often.

Scheme. We define a sequence $\{X_i\}_{i\in\mathbb{N}} : X_0 \subseteq X_1 \subseteq \cdots$ of sets of states which are winning for player 1 with a positive probability. In the definition of $\{X_i\}_{i\in\mathbb{N}}$, we use an auxiliary sequence $\{M_i\}_{i\in\mathbb{N}} : M_0 \supseteq M_1 \supseteq \cdots$ of sets of states. The construction is inductive where $X_0 := \emptyset$, $M_0 := S$ and

$$M_{i+1} := \text{Force}^0(\overline{X_i}, F) \qquad X_{i+1} := \text{Force}^1(\overline{M_{i+1}})$$

for each $i \geq 0$. Intuitively, the set X_i consists of states "already classified as losing for player 0". We add states iteratively to these sets. We define M_{i+1} such that $\overline{M_{i+1}}$ is the set of states where player 0 cannot reach F with positive probability while staying always in $\overline{X_i}$. Finally, we claim that the winning states for player 0 are given by $W^0 := \bigcap_{i\geq 0} M_i$, and thus complementarily, the winning states for player 1 are given by $W^1 := \overline{W^0} = \bigcup_{i\geq 0} X_i$.

This property holds by the definitions and will be used later in this section.

Lemma 5.1 $X_0 \subseteq \overline{M_1} \subseteq X_1 \subseteq \overline{M_2} \subseteq X_2 \subseteq \cdots$

The following lemma shows that this construction terminates.

Lemma 5.2 *The sequence $\{X_i\}_{i\in\mathbb{N}}$ converges for any set $F \subseteq S$ of states.*

Proof. (Sketch; details in the appendix) Consider the sequence in Lemma 5.1. We perform the proof in four steps; namely, we show that (i) there is a K such that $[X_K]^R = [X_{K+1}]^R$; (ii) $X_{K+1} = \overline{M_{K+1}}$; (iii) $M_{K+1} = M_{K+2}$; (iv) $X_{K+1} = X_{K+2}$.

(i) We show that each $[X_i]^R$ is upward closed, using induction on i. The base case is trivial since $X_0 = \emptyset$. For the induction step we let $Y := \left[\overline{M_{i+1}}\right]^{01} \cup [X_i]^R$. Using the definitions of X_i, X_{i+1}, and M_{i+1}, it can be shown that $X_{i+1} = \text{Force}^1(Y)$. Since $[X_i]^R$ is upward closed by the induction hypothesis it follows by Lemma 4.2(2) that $[X_{i+1}]^R$ is upward closed. From this and well quasi-ordering of $\preceq$, we get $\exists K. [X_K]^R = [X_{K+1}]^R$. We will use K in the rest of the analysis below.

(ii) From Lemma 5.1 and the fact that $[X_K]^R = [X_{K+1}]^R$, we know that $[X_K]^R = \left[\overline{M_{K+1}}\right]^R = [X_{K+1}]^R$. This is used to show that $\text{Pre}^R(\overline{M_{K+1}})$, $\text{Pre}^1(\overline{M_{K+1}})$, $\widetilde{\text{Pre}}^0(\overline{M_{K+1}}) \subseteq \overline{M_{K+1}}$ which by the definition of X_{K+1} implies $X_{K+1} \subseteq \overline{M_{K+1}}$. Hence, $X_{K+1} = \overline{M_{K+1}}$, by Lemma 5.1.

(iii) Since $M_{K+2} = \text{Force}^0(\overline{X_{K+1}}, F)$ and $X_{K+1} = \overline{M_{K+1}}$, we have that $M_{K+2} = \text{Force}^0(M_{K+1}, F)$. From Lemma 4.2(4) and the fact that $M_{K+1} = \text{Force}^0(\overline{X_K}, F)$, it follows that $M_{K+2} = M_{K+1}$.

(iv) $X_{K+2} = \text{Force}^1(\overline{M_{K+2}}) = \text{Force}^1(\overline{M_{K+1}}) = X_{K+1}$.

□

Forms of Winning Sets. From Lemma 4.2(1), it follows that if F is regular then each X_i and M_i is regular. From Lemma 5.2 we get the following:

Lemma 5.3 *If F is regular then W^0 and W^1 are regular.*

Winning Strategy for Player 1. We define a sequence $\{x_i\}_{i\in\mathbb{N}}$ of strategies for player 1, such that $x_0 \subseteq x_1 \subseteq \cdots$. For each i, the strategy $x_i : [X_i]^1 \to S$ is memoryless and winning for player 1 from states in X_i. The sequence $\{x_i\}_{i\in\mathbb{N}}$ converges to a memoryless strategy $w^1 := \bigcup_{i\in\mathbb{N}} x_i$ for player 1 which is winning from states in W^1. We define the sequence using induction on i. We will also motivate why the strategy is winning for player 1. Define $x_0 := \emptyset$. For all $i \geq 0$, we define $x_{i+1}(s)$ by case analysis. By Lemma 5.1, we know that $X_i \subseteq \overline{M_{i+1}} \subseteq X_{i+1}$. There are three cases, reflecting the membership of s in these three sets:

(i) If $s \in X_i$ then $x_{i+1}(s) := x_i(s)$. Here, we know by the induction hypothesis that a winning strategy x_i for player 1 has already been defined in s.

(ii) If $s \in \overline{M_{i+1}} - X_i$ then $x_{i+1}(s) := \mathit{select}(\mathrm{Post}(s) \cap \overline{M_{i+1}})$. The idea is that player 1 uses a strategy which guarantees that any run either (A) will stay in $\overline{M_{i+1}} - X_i$; or (B) will eventually enter X_i. In (A), player 1 wins since $\overline{M_{i+1}} - X_i$ does not have any states in F by the definition of M_{i+1}. In (B), player 1 wins by the induction hypothesis.

    More precisely, we observe that player 1 selects a successor of s which belongs to $\overline{M_{i+1}}$. Such a successor exists by the following argument. First, observe that (by set operations) $\overline{M_{i+1}} - X_i = \overline{X_i} - M_{i+1}$. The result follows by instantiating Lemma 4.2(3) with $I = \overline{X_i}$ and $Q = M_{i+1}$. By the same argument, for each $s' \in \left[\overline{M_{i+1}}\right]^R \cup \left[\overline{M_{i+1}}\right]^0$, all successors of s' belong to $\overline{M_{i+1}}$. This guarantees that either (A) or (B) holds.

(iii) If $s \in X_{i+1} - \overline{M_{i+1}}$ then $x_{i+1}(s) := \underline{\mathrm{force}}^1(\overline{M_{i+1}})(s)$. Since, by definition, $X_{i+1} = \mathrm{Force}^1(\overline{M_{i+1}})$, player 1 can use $\underline{\mathrm{force}}^1(\overline{M_{i+1}})$ to take the game with a positive probability to $\overline{M_{i+1}}$ (Lemma 4.3). From there, player 1 wins as described above.

Now, consider a state $s \in W^1$. By definition, we know that $s \in X_i$ for some $i \geq 0$. This means that $w^1 = x_i$ is winning for player 1 from s according to the above argument. Hence:

Lemma 5.4 *For each $s \in W^1$, $\mathcal{P}_{w^1}(s \models \neg\Box\Diamond F) > 0$.*

Winning Strategy for Player 0. In this paragraph, we define a memoryless strategy w^0 and we prove that it is winning.

To describe how w^0 is defined, we rely on two auxiliary results on games induced by GPLCS. First, we recall the definition of *an attractor.* A set $A \subseteq S$ is called an attractor if $\mathcal{P}(s \models \Diamond A) = 1$ for any $s \in S$. In other words, from any state $s \in S$, A is almost surely visited regardless of the strategies of the players. The following result was shown in [11, 7, 4] for *probabilistic LCS*, where moves in the control graph are taken probabilistically instead of by two competing players. The results straightforwardly generalize to GPLCS.

Lemma 5.5 *Let $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda)$ be a GPLCS and let $\mathcal{G}$ be the game induced by $\mathcal{L}$. The set $A = (\mathtt{S} \times \varepsilon \times \{0,1\})$ is a finite attractor in $\mathcal{G}$.*

The second result follows from Lemma 5.5 and Lemma 4.3 as described below.

Lemma 5.6 *Let $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ be a game induced by a GPLCS. For any $Q, I \subseteq S$ and $\sigma \in \{0,1\}$, the following holds: For any $s \in \mathit{Force}^\sigma(I,Q)$, $\mathcal{P}_{\underline{\mathit{force}}^\sigma(I,Q)}(s \models \Box \mathit{Force}^\sigma(I,Q) \wedge \neg\Box\Diamond Q) = 0$.*

Proof. Given $Q, I \subseteq S$, $\sigma \in \{0,1\}$, and $s \in \mathrm{Force}^\sigma(I,Q)$. We assume that player σ uses an extension f^σ of the $\underline{\mathrm{force}}^\sigma(I,Q)$ strategy and player $(1-\sigma)$ uses a strategy $f^{1-\sigma}$. By Lemma 5.5, the game has a finite attractor A. By definition of the attractor, almost all runs must visit A infinitely often. We define $A' := A \cap \mathrm{Force}^\sigma(I,Q)$.

If $A' = \emptyset$, then $\mathcal{P}_{f^\sigma, f^{1-\sigma}}(s \models \Box\mathrm{Force}^\sigma(I,Q) \wedge \neg\Box\Diamond Q) \leq \mathcal{P}_{f^\sigma, f^{1-\sigma}}(s \models \neg\Diamond A) = 0$, where the inequality follows from the assumption and the equality from the definition of an attractor.

Consider now the case where $A' \neq \emptyset$. By Lemma 4.3 and finiteness of A (and thus A'), we obtain that $\epsilon := \min_{s'\in A'}(\epsilon'_s) > 0$. Almost every run in $(s \models \Box\mathrm{Force}^\sigma(I,Q) \wedge \neg\Box\Diamond Q)$ must visit A' infinitely many times, but Q only finitely many times (and thus have an infinite suffix which never visits Q). Thus,

$$\mathcal{P}_{f^\sigma, f^{1-\sigma}}(s \models \Box\mathrm{Force}^\sigma(I,Q) \wedge \neg\Box\Diamond Q) \leq (1-\epsilon)^\infty = 0. \tag{1}$$

Therefore $\mathcal{P}_{\underline{\mathrm{force}}^\sigma(I,Q)}(s \models \Box\mathrm{Force}^\sigma(I,Q) \wedge \neg\Box\Diamond Q) = 0$. □

Remark 1 *Observe that the inequality (1) holds for any strategy $f^{1-\sigma}$ of the opponent and any extension f^σ of the $\underline{\mathit{force}}^\sigma(I,Q)$ strategy. In particular, we do* **not** *require that $f^{1-\sigma}$ is finite-memory. It is possible that $f^{1-\sigma}$ acts quite differently after each of the (possibly infinitely many) visits to the same state in the attractor. The crucial fact is that the quantity $\epsilon_s > 0$ in Lemma 4.3 is independent of $f^{1-\sigma}$.*

Now we are ready to describe the winning strategy w^0 for player 0. The idea of the strategy w^0 is to keep the run within a force set of F with probability 1. This implies that F will be visited infinitely often with probability 1, by Lemma 5.6. In order to do that, player 0 exploits certain properties of W^0 as follows. By Lemmas 5.2 and 5.1 it follows that there is an i such that $X_i = \overline{M_{i+1}} = X_{i+1}$. From this and the definition of W^0 it follows that $W^0 = \overline{X_i} = M_{i+1}$. From $W^0 = \overline{X_i}$ and Lemma 4.2(5) it follows that W^0 is a 1-trap. From $M_{i+1} = \text{Force}^0(\overline{X_i}, F)$, it follows that $W^0 = \text{Force}^0(W^0, F)$. We define w^0 on any state $s \in \left[W^0\right]^0$ as follows:

- If $s \in W^0 - F$ then $w^0(s) := \underline{\text{force}}^0(W^0, F)(s)$. This definition is possible since $W^0 = \text{Force}^0(W^0, F)$.
- If $s \in W^0 \cap F$ then $w^0(s) := \textit{select}(\text{Post}(s) \cap W^0)$. This is possible since W^0 is a 1-trap, and therefore s has at least one successor in W^0.

Consider any run ρ starting from a state inside W^0, where player 0 follows w^0. Since W^0 is a 1-trap, ρ will always remain inside W^0 regardless of the strategy of player 1. This implies that $\mathcal{P}_{w^0}(s \models \Box W^0) = 1$. Furthermore, by Lemma 5.6 and the definitions of w^0 and W^0, it follows that $\mathcal{P}_{w^0}(s \models \Box W^0 \wedge \neg\Box\Diamond F) = 0$, which gives the following lemma:

Lemma 5.7 *For any* $s \in W^0$, $\mathcal{P}_{w^0}(s \models \Box\Diamond F) = 1$.

Proof. Consider a state $s \in W^0$. We assume that player 1 uses a total strategy f^1, while player 0 uses a total extension f^0 of w^0. Then,

$$\begin{aligned} 1 = \mathcal{P}_{f^0,f^1}(s \models \Box W^0) &= \mathcal{P}_{f^0,f^1}(s \models \Box W^0 \wedge \Box\Diamond F) \\ &\quad + \mathcal{P}_{f^0,f^1}(s \models \Box W^0 \wedge \neg\Box\Diamond F) \\ &= \mathcal{P}_{f^0,f^1}(s \models \Box W^0 \wedge \Box\Diamond F) \\ &\leq \mathcal{P}_{f^0,f^1}(s \models \Box\Diamond F); \end{aligned}$$

where the first equality follows from the definition of f^0 (extension of w^0) and the fact that W^0 is 1-trap; the second equality holds by definition of the probability measure; the third equality follows from Lemma 5.6 and the definitions of W^0 and w^0; and the inequality follows from the fact that $(s \models \Box W^0 \wedge \Box\Diamond F) \subseteq (s \models \Box\Diamond F)$. Now the result follows since $\mathcal{P}_{f^0,f^1}(s \models \Box\Diamond F) = 1$ holds for any state in $s \in W^0$, regardless of the extension of w^0 and the strategy f^1 of the opponent. $\Box$

Determinacy and Solvability. Memoryless determinacy of almost-sure Büchi-GPLCS follows from Lemmas 5.4 and 5.7. By Lemma 5.3, for any state $s \in S$, we can check whether $s \in W^0$ or $s \in W^1$. This gives the main result:

Theorem 5.8 *Büchi-GPLCS are memoryless determined and solvable, for any regular target set* F.

6 Conclusions and Future Work

We have introduced GPLCS and given a terminating algorithm to compute symbolic representations of the winning sets in almost-sure Büchi-GPLCS. The strategies are memoryless, and our construction implies that the games we consider are memoryless determined.

The problem of deciding GPLCS games is not primitive recursive, since it is harder than the control-state reachability problem for LCS, which was shown to be non-primitive recursive by Schnoebelen in [25]. (For a given LCS and control-state q we can construct a GPLCS by defining $\mathtt{S}^0 = \emptyset$, $\mathtt{S}^1 = \mathtt{S}$, making the state q absorbing and defining F as all configurations where the control-state is not q. Then player 1 has a winning strategy in the GPLCS iff control-state q is reachable in the LCS.)

We remark that there are five immediate extensions of our result. (1) Each winning strategy w^0, w^1 wins against any *mixed* strategy of the opponent. (2) Our algorithm is easily adapted to almost-sure *reachability*-GPLCS. This is achieved by replacing all outgoing transitions of states in F by self-loops, or, equivalently, replacing the definition of X_{i+1} by $X_{i+1} := \text{Force}^1(\overline{F}, \overline{M_{i+1}})$. (3) Our algorithm can be modified to construct symbolic representations of the winning strategies. A strategy is represented as a finite set $\{L_i, L'_i\}_{i=0}^n$ of pairs of regular state languages, where all L_i are disjoint. Such a finite set represents the strategy f^0 where $f^0(s) = \textit{select}(L'_i)$ if $s \in L_i$. (4) We can extend the scheme to *concurrent* games, where the two players move simultaneously, by an appropriate extension of the Pre operator, as in [15]. (5) The algorithm also works when there are *probabilistic* control states in the GPLCS (see, e.g., [1] for definitions), as well as control states owned by the players and probabilistic message losses.

We mention as future work the question whether parity games can be solved for GPLCS. This problem is related to solving almost-sure *coBüchi*-GPLCS (equivalently, characterize the set of states where player 0 wins Büchi-GPLCS with a positive probability). For the full class of strategies, almost-sure coBüchi-GPLCS are undecidable even in one-player games [10]. However, it is conceivable that the problem becomes decidable if both players are restricted to *finite-memory strategies*, as in [10]. Our hope is to prove termination and correctness for the schemes in [15], when extended to GPLCS where the players are restricted to finite-memory strategies.

References

[1] P. A. Abdulla, C. Baier, P. Iyer, and B. Jonsson. Reasoning about probabilistic lossy channel systems. In *Proc. CONCUR 2000,* 11^{th} *Int. Conf. on Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 320–333, 2000.

[2] P. A. Abdulla, N. Ben Henda, R. Mayr, S. Sandberg, and L. de Alfaro. Stochastic games with lossy channels. Technical Report 2007-005, Dept. of Information Technology, Uppsala University, Sweden, Feb. 2007.

[3] P. A. Abdulla, A. Bouajjani, and J. d'Orso. Deciding monotonic games. In *Proc. CSL 2003*, volume 2803 of *Lecture Notes in Computer Science*, pages 1–14, 2003.

[4] P. A. Abdulla, N. B. Henda, and R. Mayr. Verifying infinite Markov chains with a finite attractor or the global coarseness property. In *Proc. LICS '05,* 21^{st} *IEEE Int. Symp. on Logic in Computer Science*, pages 127–136, 2005.

[5] P. A. Abdulla, N. B. Henda, R. Mayr, and S. Sandberg. Eager Markov chains. In *Proc. ATVA '06,* 4^{th} *Int. Symp. on Automated Technology for Verification and Analysis*, volume 4218 of *Lecture Notes in Computer Science*, pages 24–38. Springer Verlag, 2006.

[6] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[7] P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *Proc. FOSSACS03, Conf. on Foundations of Software Science and Computation Structures*, volume 2620 of *Lecture Notes in Computer Science*, pages 39–53, 2003.

[8] E. Asarin and P. Collins. Noisy Turing machines. In *Proc. ICALP '05,* 32^{nd} *International Colloquium on Automata, Languages, and Programming*, pages 1031–1042, 2005.

[9] C. Baier, N. Bertrand, and P. Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. In *Proc. LPAR'2006*, pages 347–361, 2006.

[10] C. Baier, N. Bertrand, and P. Schnoebelen. Verifying nondeterministic probabilistic channel systems against ω-regular linear-time properties. *ACM Transactions on Comp. Logic*, 2006. To appear.

[11] N. Bertrand and P. Schnoebelen. Model checking lossy channels systems is probably decidable. In *Proc. FOSSACS03, Conf. on Foundations of Software Science and Computation Structures*, volume 2620 of *Lecture Notes in Computer Science*, pages 120–135, 2003.

[12] K. Chatterjee, M. Jurdziński, and T. Henzinger. Simple stochastic parity games. In *Proceedings of the International Conference for Computer Science Logic (CSL)*, volume 2803 of *Lecture Notes in Computer Science*, pages 100–113. Springer Verlag, 2003.

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

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

[15] L. de Alfaro and T. Henzinger. Concurrent omega-regular games. In *Proc. LICS '00,* 16^{th} *IEEE Int. Symp. on Logic in Computer Science*, pages 141–156, Washington - Brussels - Tokyo, June 2000. IEEE.

[16] L. de Alfaro, T. Henzinger, and O. Kupferman. Concurrent reachability games. In *Proc.* 39^{th} *Annual Symp. Foundations of Computer Science*, pages 564–575. IEEE Computer Society Press, 1998.

[17] L. de Alfaro, T. Henzinger, and R. Majumdar. Symbolic algorithms for infinite-state games. In K. G. Larsen and M. Nielsen, editors, *Proc. CONCUR 2001,* 12^{th} *Int. Conf. on Concurrency Theory*, volume 2154 of *Lecture Notes in Computer Science*, pages 536–550. Springer Verlag, 2001.

[18] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *Proc. LICS '04,* 20^{th} *IEEE Int. Symp. on Logic in Computer Science*, pages 12–21, 2004.

[19] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proc. ICALP '05, 32^{nd} International Colloquium on Automata, Languages, and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 891–903, 2005.

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

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

[22] P. Iyer and M. Narasimha. Probabilistic lossy channel systems. In *TAPSOFT '97: Theory and Practice of Software Development*, volume 1214 of *Lecture Notes in Computer Science*, pages 667–681, 1997.

[23] J. Kemeny, J. Snell, and A. Knapp. *Denumerable Markov Chains.* D Van Nostad Co., 1966.

[24] A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *Proc. ICALP '03, 30^{th} International Colloquium on Automata, Languages, and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 1008–1021, 2003.

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

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

[27] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS '85, 26^{th} Annual Symp. Foundations of Computer Science*, pages 327–338, 1985.

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

A Appendix

A.1 Section 4 (Reachability Games on GPLCS)

Lemma 4.2. *Let $Q = \mathit{Force}^\sigma(I, F)$. Then:*

(1) If F and I are regular then Q is regular.

(2) If F and I are upward closed then $[Q]^R$ is upward closed.

(3) Let $s \in I - Q$. If $s \in S^\sigma \cup S^R$, then $\mathit{Post}(s) \subseteq \overline{Q}$. If $s \in S^{1-\sigma}$, then $\mathit{Post}(s) \cap \overline{Q} \neq \emptyset$.

(4) $\mathit{Force}^\sigma(Q, F) = Q$.

(5) $\overline{\mathit{Force}^\sigma(F)}$ is a σ-trap.

Proof. (1) and (2) were proved in the main text.

(3) follows since, by definition, Q is closed under the $\mathrm{Pre}^\sigma(\cdot) \cap I$, $\widetilde{\mathrm{Pre}}^{1-\sigma}(\cdot) \cap I$ and $\mathrm{Pre}^R(\cdot) \cap I$ operators.

(4) Since any force-set is a subset of the invariant, $\mathrm{Force}^\sigma(Q, F) \subseteq Q$. It remains to prove that $Q \subseteq \mathrm{Force}^\sigma(Q, F)$. In order to do that, we recall the construction of the force sets. We use $\{E_i\}_{i\in\mathbb{N}}$, $\{D_i\}_{i\in\mathbb{N}}$ and $\{\tilde{E}_i\}_{i\in\mathbb{N}}$, $\{\tilde{D}_i\}_{i\in\mathbb{N}}$ to denote the sequences used in the construction of $Q = \mathrm{Force}^\sigma(I, F)$ and $\mathrm{Force}^\sigma(Q, F)$, respectively. We prove by induction on i that $E_i \subseteq \tilde{E}_i$ and $D_i \subseteq \tilde{D}_i$ for all $i \in \mathbb{N}$. The base case is trivial. Assume the claim holds for $i \geq 0$. By definition, $D_{i+1} = (D_i \cup \mathrm{Pre}^R(E_i)) \cap I$ and $\tilde{D}_{i+1} = (\tilde{D}_i \cup \mathrm{Pre}^R(\tilde{E}_i)) \cap Q$. By induction hypothesis, $E_i \subseteq \tilde{E}_i$ and thus $\mathrm{Pre}^R(E_i) \subseteq \mathrm{Pre}^R(\tilde{E}_i)$. Moreover, $D_{i+1} \subseteq Q$ by definition of Q. This combined with the induction hypothesis $D_i \subseteq \tilde{D}_i$ gives $D_{i+1} \subseteq \tilde{D}_{i+1}$. By a similar argument, we obtain that $E_{i+1} \subseteq \tilde{E}_{i+1}$.

(5) follows from (3) and the definition of a σ-trap. □

A.2 Section 5 (Buchi-Games on GPLCS)

Proof of Lemma 5.2. The following results used in the proof were explained in the main text (proof sketch of Lemma 5.2) as (i)–(iv). Below, we give a detailed proof of each of them.

Lemma A1. *The set $[X_i]^R$ is upward closed for each $i \geq 0$.*

Proof. We use induction on i. The base case is trivial since $X_0 = \emptyset$. Assume inductively that $[X_i]^R$ is upward closed and define $Y := [\overline{M_{i+1}}]^{01} \cup [X_i]^R$ as in the main text. See Figure 1. We now prove that $[\overline{M_{i+1}}]^R \subseteq Y \cup \mathrm{Pre}^R(Y)$. We apply Lemma 4.2(3) to the construction of M_{i+1} and obtain $\mathrm{Post}([\overline{X_i} - M_{i+1}]^R) \subseteq$

$\overline{M_{i+1}}$. By basic set operations, this implies $[\overline{M_{i+1}}]^R - [X_i]^R \subseteq \mathrm{Pre}^R(\overline{M_{i+1}})$. Taking the union with $[X_i]^R$ on both sides gives $[\overline{M_{i+1}}]^R \subseteq \mathrm{Pre}^R(\overline{M_{i+1}}) \cup [X_i]^R$. Since the graph is bipartite, we even have $[\overline{M_{i+1}}]^R \subseteq \mathrm{Pre}^R([\overline{M_{i+1}}]^{01}) \cup [X_i]^R$. By the definition of Y, this implies that $\overline{M_{i+1}} \subseteq Y \cup \mathrm{Pre}^R(Y)$.

It follows that $\mathrm{Force}^1(\overline{M_{i+1}}) \subseteq \mathrm{Force}^1(Y \cup \mathrm{Pre}^R(Y)) = \mathrm{Force}^1(Y)$. By Lemma 5.1, we have $Y \subseteq \overline{M_{i+1}}$, and hence also $\mathrm{Force}^1(Y) \subseteq \mathrm{Force}^1(\overline{M_{i+1}})$. Thus, $X_{i+1} = \mathrm{Force}^1(Y)$. Since $[Y]^R = [X_i]^R$, the induction hypothesis implies that $[Y]^R$ is upward closed. Since $X_{i+1} = \mathrm{Force}^1(\overline{M_{i+1}}) = \mathrm{Force}^1(Y)$, Lemma 4.2(2) implies that $[X_{i+1}]^R$ is upward closed. □

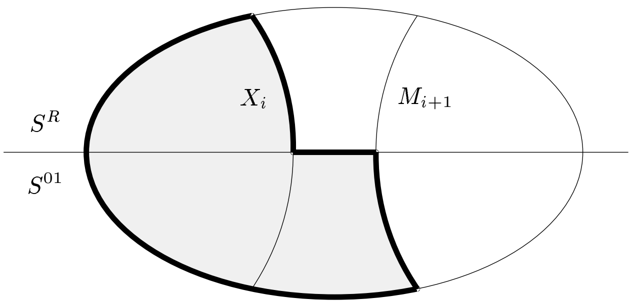

Figure 1. The big ellipse represents S and is partitioned into to S^{01} and S^R. The left and right parts represent X_i and M_{i+1}. The highlighted zone surrounded by a bold line represents Y.

Lemma A2. *There exists a $j \geq 1$ such that $X_j = \overline{M_j}$.*

Proof. By Lemma A1 and well quasi-ordering of $\preceq$ it follows that there is a k such that $[X_k]^R = [X_{k+1}]^R$. We choose $j := k+1$. By Lemma 5.1, we only need to show that $X_{k+1} \subseteq \overline{M_{k+1}}$. Also, from Lemma 5.1 and $[X_k]^R = [X_{k+1}]^R$, we know that $[X_k]^R = [\overline{M_{k+1}}]^R = [X_{k+1}]^R$. We now prove $X_{k+1} \subseteq \overline{M_{k+1}}$. Since $X_{k+1} = \mathrm{Force}^1(\overline{M_{k+1}})$, it suffices to prove that $\mathrm{Pre}^R(\overline{M_{k+1}}), \mathrm{Pre}^1(\overline{M_{k+1}}), \widetilde{\mathrm{Pre}}^0(\overline{M_{k+1}}) \subseteq \overline{M_{k+1}}$; which we do as follows:

- Since $X_{k+1} = \mathrm{Force}^1(\overline{M_{k+1}})$, it follows that $\mathrm{Pre}^R(\overline{M_{k+1}}) \subseteq [X_{k+1}]^R$ and hence $\mathrm{Pre}^R(\overline{M_{k+1}}) \subseteq [\overline{M_{k+1}}]^R \subseteq \overline{M_{k+1}}$.
- By the definition of the game induced by the GPLCS, the graph is bipartite such that $\mathrm{Pre}^1(\overline{M_{k+1}}) = \mathrm{Pre}^1([\overline{M_{k+1}}]^R)$. From $[X_k]^R = [\overline{M_{k+1}}]^R$ we get $\mathrm{Pre}^1(\overline{M_{k+1}}) = \mathrm{Pre}^1([X_k]^R)$. From the definition of X_k it follows that $\mathrm{Pre}^1([X_k]^R) \subseteq X_k$, so $\mathrm{Pre}^1(\overline{M_{k+1}}) \subseteq X_k$From Lemma 5.1, we know that $X_k \subseteq \overline{M_{k+1}}$. Consequently, $\mathrm{Pre}^1(\overline{M_{k+1}}) \subseteq \overline{M_{k+1}}$.
- The proof that $\widetilde{\mathrm{Pre}}^0(\overline{M_{k+1}}) \subseteq \overline{M_{k+1}}$ is similar.

□

We now perform steps (iii) and (iv) from the main text in one lemma.

Lemma 5.2. *The sequence $\{X_i\}_{i\in\mathbb{N}}$ converges for any set $F \subseteq S$ of states.*

Proof. By Lemma A2, there is a $j \geq 1$ such that $X_j = \overline{M_j}$. We show that both $M_{j+1} = M_j$ and $X_{j+1} = X_j$. Since $M_{j+1} = \mathrm{Force}^0(\overline{X_j}, F)$, we have that $M_{j+1} = \mathrm{Force}^0(M_j, F)$. From Lemma 4.2(4) and the fact that $M_j = \mathrm{Force}^0(\overline{X_{j-1}}, F)$, it follows that $M_{j+1} = M_j$. Now, $X_{j+1} = \mathrm{Force}^1(\overline{M_{j+1}}) = \mathrm{Force}^1(\overline{M_j}) = X_j$. □

Proof of Lemma 5.4. The following auxiliary lemma will be used in the sequel.

Lemma A3. *Let $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ be a game, $Q \subseteq S$ a set of states, $s \in S$ a state, and f^0, f^1 two total strategies of player 0 and 1, respectively. The following two properties are equivalent.*

(1) There is a path $s_0 \cdots s_j$ from s to Q, such that for each $\sigma \in \{0,1\}$ and each $i : 0 \leq i < j$, if $s_i \in S^\sigma$ then $f^\sigma(s_0 \cdots s_i) = s_{i+1}$.

(2) $\mathcal{P}_{f^0,f^1}(s \models \Diamond Q) > 0$.

Proof. Follows by the definition of the probability measure. □

The following can be seen as a version of Lemma A3 and is needed in the proof of Lemma 5.4.

Lemma A4. *Let $\mathcal{G} = (S, S^0, S^1, S^R, \longrightarrow, P)$ be a game, φ a CTL* path formula, $s, s' \in S$ states, and $(\tilde{f}^0, \tilde{f}^1, f^0, f^1)$ a quadruple of strategies. Assume that $\tilde{f}^0, \tilde{f}^1$ are memoryless strategies of player 0 and 1, respectively, and $f^0 \supseteq \tilde{f}^0, f^1 \supseteq \tilde{f}^1$ are extensions of them. If $\mathcal{P}_{f^0,f^1}(s \models \Diamond\{s'\}) > 0$ and $\mathcal{P}_{\tilde{f}^0,\tilde{f}^1}(s' \models \Diamond\varphi) > 0$, then $\mathcal{P}_{f^0,f^1}(s \models \Diamond\varphi) > 0$.*

Proof. We will first use Lemma A3 twice to obtain paths from s to s' and from s' to $Q := \{s : \mathcal{P}_{f^0,f^1}(s \models \varphi) > 0)\}$. Then we will concatenate these paths and use the opposite direction of the equivalence in Lemma A3 to prove the result.

Let $g^0 \supseteq f^0$ and $g^1 \supseteq f^1$ be total strategies extending f^0, f^1. We will prove that $\mathcal{P}_{g^0,g^1}(s \models \Diamond Q) > 0$. By

Lemma A3, there is a path $s_0 \cdots s_j$ from s to s', such that

$$\text{for each } \sigma \in \{0,1\} \text{ and each } i : 0 \le i < j, \\ \text{if } s_i \in S^\sigma \text{ then } g^\sigma(s_0 \cdots s_i) = s_{i+1}. \tag{2}$$

Define the strategies h^0, h^1 of both players by $h^\sigma(\pi) := g^\sigma(s_0 \cdots s_{j-1}\pi)$ for any $\sigma \in \{0,1\}$ and path $\pi \in \Pi_{S^\sigma}$. Since $\tilde{f}^0, \tilde{f}^1$ are memoryless, we have $h^0 \supseteq \tilde{f}^0$ and $h^1 \supseteq \tilde{f}^1$. Hence, the hypothesis of this lemma implies that $\mathcal{P}_{h^0,h^1}(s' \models \Diamond Q) > 0$. Therefore, we can apply Lemma A3 again and obtain that there is a path $s_j \cdots s_k$ from $s_j = s'$ to some $s_k \in Q$, such that

$$\text{for each } \sigma \in \{0,1\} \text{ and each } i : j \le i < k, \\ \text{if } s_i \in S^\sigma \text{ then } h^\sigma(s_j \cdots s_i) = s_{i+1}. \tag{3}$$

Consider the path $s_0 \cdots s_k$. We will prove that for each $\sigma \in \{0,1\}$ and each $i : 0 \le i < k$, if $s_i \in S^\sigma$ then $g^\sigma(s_0 \cdots s_i) = s_{i+1}$. This follows from (2) if $i < j$. Take any $\sigma \in \{0,1\}$ and any $i : j \le i < k$. If $s_i \in S^\sigma$ then $g^\sigma(s_0 \cdots s_i) = h^\sigma(s_j \cdots s_i)$ by the definition of h^σ. Hence, (3) implies that $g^\sigma(s_0 \cdots s_i) = s_{i+1}$. By Lemma A3, $\mathcal{P}_{g^0,g^1}(s \models \Diamond Q) > 0$. This is equivalent to $\mathcal{P}_{g^0,g^1}(s \models \Diamond \varphi) > 0$. Since g^0, g^1 were arbitrary, the lemma follows. □

Lemma 5.4. *For each $s \in W^1$, $\mathcal{P}_{w^1}(s \models \neg\Box\Diamond F) > 0$.*

Proof. We first observe the following facts for each $i \in \mathbb{N}$:

$$\overline{X_i} \text{ is a 1-trap;} \tag{4}$$

$$F \subseteq X_i \cup M_{i+1}. \tag{5}$$

(4) holds by the definition of X_i and Lemma 4.2(5). By the definition of M_{i+1}, for any $i \in \mathbb{N}$, we have $F - X_i \subseteq M_{i+1}$. This is equivalent to (5).

We now use induction on i to prove that for each $i \in \mathbb{N}$ and each $s \in X_i$,

$$\mathcal{P}_{x_i}(s \models \neg\Box\Diamond F) > 0. \tag{6}$$

The base case $i = 0$ is trivial since $X_0 = \emptyset$.

For the induction step we assume that the claim is true for some $i \ge 0$ and prove the claim for $i+1$. For $s \in X_{i+1}$ there are three cases. Note that by Lemma 5.1, we have $X_i \subseteq \overline{M_{i+1}} \subseteq X_{i+1}$. We prove (6) for all three cases: first on X_i, then on $\overline{M_{i+1}} - X_i$, and finally on $X_{i+1} - \overline{M_{i+1}}$ (see Figure 2).

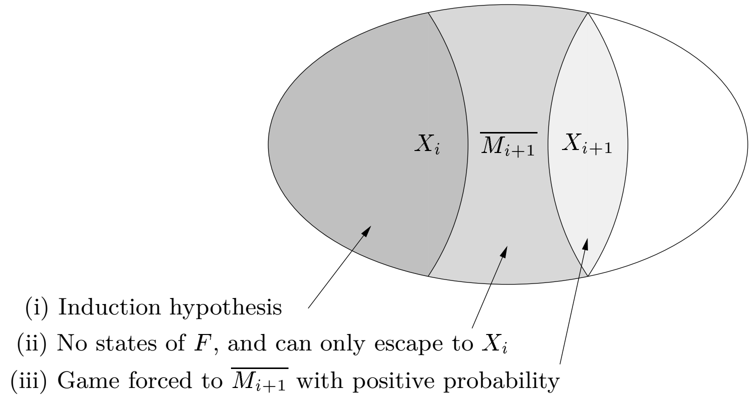

Figure 2. The big ellipse is S, the dark gray part is X_i, the dark and middle gray parts together form $\overline{M_{i+1}}$, and all three gray parts constitute X_{i+1}. In Lemma 5.4, the proof that x_{i+1} wins on X_{i+1} is in three steps, corresponding to the dark gray, middle gray, and light gray parts.

(i) By the induction hypothesis and the definition of x_{i+1} on X_i, player 1 wins with a positive probability on X_i.

(ii) Let $Z := \overline{M_{i+1}} - X_i$, let $f^1 \supseteq x_{i+1}$ be any total strategy extending x_{i+1}, and let f^0 be any total strategy of player 0. By the definition of x_{i+1}, for any $s \in Z$ we have $\mathcal{P}_{f^0,f^1}(s \models \Box\overline{M_{i+1}}) = 1$. Hence, for any $s \in Z$, either $\mathcal{P}_{f^0,f^1}(s \models \Diamond X_i) > 0$ or $\mathcal{P}_{f^0,f^1}(s \models \Box Z) = 1$. In the first case, Lemma A4 (with the quadruple $(\emptyset, x_{i+1}, f^0, f^1)$) and the induction hypothesis implies that $\mathcal{P}_{f^0,f^1}(s \models \Diamond\neg\Box\Diamond F) > 0$, which is equivalent to $\mathcal{P}_{f^0,f^1}(s \models \neg\Box\Diamond F) > 0$. In the second case, $\mathcal{P}_{f^0,f^1}(s \models \neg\Box\Diamond F) = 1$ since, by (5), Z does not contain any states in F.

(iii) By Lemma A4 and the definition of x_{i+1}, (6) holds for any state $s \in X_{i+1} - \overline{M_{i+1}}$ too.

Thus, (6) holds on the entire X_{i+1}.

We are now able to prove the lemma. By the definition of W^1, there is a minimal i such that $s \in X_i$ and $w^1 \supseteq x_i$. The result now follows by (6). □

Lemma 5.5. *Let $\mathcal{L} = (\mathtt{S}, \mathtt{S}^0, \mathtt{S}^1, \mathtt{C}, \mathtt{M}, \mathtt{T}, \lambda)$ be a GPLCS and let $\mathcal{G}$ be the game induced by $\mathcal{L}$. The set $A = (\mathtt{S} \times \varepsilon \times \{0,1\})$ is a finite attractor in $\mathcal{G}$.*

Proof. This was proved for *probabilistic LCS*, where moves in the control graph are taken probabilistically instead of by two competing players, in [11, 7, 4, 5]. The proof relies on the observation that if the number of messages in some channel is big enough, it is more likely that the number of messages decreases than that it increases. The proof straightforwardly generalizes to GPLCS. □