In ***Proc. 15th IEEE Symp. Logic in Comp. Sci., 2000.***

Concurrent Omega-Regular Games*

Luca de Alfaro Thomas A. Henzinger

Department of EECS, University of California at Berkeley, CA 94720-1770, USA
Email:{dealfaro,tah}@eecs.berkeley.edu

Abstract

We consider two-player games which are played on a finite state space for an infinite number of rounds. The games are *concurrent*, that is, in each round, the two players choose their moves independently and simultaneously; the current state and the two moves determine a successor state. We consider *omega-regular* winning conditions on the resulting infinite state sequence. To model the independent choice of moves, both players are allowed to use randomization for selecting their moves. This gives rise to the following *qualitative* modes of winning, which can be studied without numerical considerations concerning probabilities: *sure-win* (player 1 can ensure winning with certainty), *almost-sure-win* (player 1 can ensure winning with probability 1), *limit-win* (player 1 can ensure winning with probability arbitrarily close to 1), *bounded-win* (player 1 can ensure winning with probability bounded away from 0), *positive-win* (player 1 can ensure winning with positive probability), and *exist-win* (player 1 can ensure that at least one possible outcome of the game satisfies the winning condition).

We provide algorithms for computing the sets of winning states for each of these winning modes. In particular, we solve concurrent Rabin-chain games in $n^{\mathcal{O}(m)}$ time, where n is the size of the game structure and m is the number of pairs in the Rabin-chain condition. While this complexity is in line with traditional *turn-based* games, where in each state only one of the two players has a choice of moves, our algorithms are considerably more involved than those for turn-based games. This is because concurrent games violate two of the most fundamental properties of turn-based games. First, concurrent games are not determined, but rather exhibit a more general duality property which involves multiple modes of winning. Second, winning strategies for concurrent games may require infinite memory.

*This research was supported in part by the DARPA grants NAG2-1214 and F33615-C-98-3614, the ARO MURI grant DAAH-04-96-1-0341, the MARCO grant 98-DT-660, and the NSF CAREER award CCR-9501708.

1 Introduction

Games provide a model for systems composed of interacting components. In particular, the interaction of a component with its environment is naturally modeled as a two-player game, component vs. environment [ALW89, Dil89, PR89, KV96, AHK97]. The game is played on a finite state space and produces an infinite path: in each round, depending on the current state of the game, the moves of one or both players determine the next state [Sha53]. Questions that may interest us about such games include the following: Does player 1 (say, a process) have a strategy to meet a specification (say, acquire a resource) no matter how player 2 (the other processes) behave? Does player 2 (say, a controller) have a strategy to keep player 1 (the process) from violating a specification? Following a successful tradition, we focus on ω-regular specifications. Depending on the complexity of the specification, we can classify the resulting games as safety ($\Box$), reachability ($\Diamond$), Büchi ($\Box\Diamond$), co-Büchi ($\Diamond\Box$), or Rabin-chain (certain restricted boolean combinations of $\Box\Diamond$ and $\Diamond\Box$). The significance of Rabin-chain specifications (winning conditions) is that every finite-state game with an ω-regular winning condition can be reduced to another finite-state game with a Rabin-chain condition (this is essentially because every ω-regular property can be specified by a deterministic Rabin-chain automaton) [Mos84, Tho90]. For example, the "receptiveness" condition [Dil89], that no environment can invalidate the fairness assumptions about a component, yields a Büchi game in the case of weak fairness, and a Rabin-chain game in the case of strong fairness.

Systems in which the interaction between the components is asynchronous give rise to *turn-based* games, where in each round only one of the two players can choose among several moves. On the other hand, synchronous interaction leads to *concurrent games*, where in each round both players can choose simultaneously and independently among several moves [AHK97]. Both types of games can be played with *deterministic* or *randomized* strategies. A player that uses a deterministic strategy must select a move based on the current state and on the history of the game. A player that uses a randomized strategy selects not a move, but a probability distribution over moves; the move to be played is then chosen at random, according to the distribution. Randomized strategies are not helpful for winning turn-based games, but they can be helpful for winning concurrent games. To see this, consider the concurrent reachability ($\Diamond$) game called MATCHBIT: in each round, both players simultaneously and independently choose a bit (0 or 1); the winning condition is satisfied if the two bits match in any round. For each deterministic strategy of player 1, there is a corresponding strategy for player 2 that prevents player 1 from winning (the strategy for player 2 always chooses a different bit than the one chosen by the strategy for player 1). However, if both players choose their bits truly simultaneously and independently, then it is extremely "likely" that the chosen bits will match in some round. This intuition can be captured mathematically by randomization: if player 1 chooses her bits at random, with uniform probability, then player 1 wins with probability 1/2 at each round, and she can win the game with probability 1.

We study concurrent games, played with randomized strategies: concurrent, as simultaneous, independent choice of moves is needed to adequately model synchronous systems; played with randomized strategies, as randomized choice of moves is needed to adequately analyze the simultaneous, independent choice of moves [Sha53]. Our main result is to solve concurrent Rabin-chain games, played with randomized strategies. The solution we present applies both to *deterministic* concurrent games, in which the current state and the moves determine a unique successor state, and to *probabilistic* concurrent games, in which the current state and the moves determine a probability distribution for the successor state. Probabilistic concurrent games generalize several models, including Markov chains, Markov decision processes, deterministic as well as probabilistic turn-based games, and deterministic concurrent games. Previously, solutions to games have been known only for (1) all varieties of deterministic turn-based games [BL69, GH82, EJ91, Tho95], (2) concurrent Rabin-chain games, played with deterministic strategies (these games can be solved like turn-based games) [AHK97], and (3) concurrent reachability games, played with randomized strategies [dAHK98]. We note that already the case for probabilistic turn-based games was open, even though these games can be analyzed with deterministic strategies.

For deterministic games, played with deterministic strategies, there is only a single "mode" of winning: the unique outcome of the game either does or does not satisfy the winning condition. For fixed randomized strategies, there are many possible outcomes and therefore many modes of winning: player 1 can ensure a win with probability 1 (as in the game MATCHBIT), or with probability greater than 7/8, etc. We are interested in the following "qualitative" modes of winning [dAHK98]: *sure-win* (player 1 has a strategy which guarantees that all possible outcomes satisfy the winning condition), *almost-sure-win* (player 1 has a strategy which guarantees a win with probability 1), and *limit-win* (for each real $\varepsilon > 0$, player 1 has a strategy which guarantees a win with probability $1 - \varepsilon$). The *limit* mode is illustrated by the reachability game SKIRMISH, which is derived from a game of [KS81]: player 1 is hiding; her goal is to run and reach home without being hit by a snowball; player 2 is armed with a single snowball. There are three states, s_{wet}, s_{hide}, and s_{home}, and the winning condition is $\Diamond s_{home}$. The only state with more than one move for either player is the state s_{hide}, where player 1 must choose between *hide* and *run*, and player 2 must choose between *wait* and *throw*. The effects of the moves are shown in Figure 1. To see that from the state s_{hide} player 1 can limit-win, given any $\varepsilon > 0$, suppose that player 1 chooses *run* with probability ε and *hide* with probability $1 - \varepsilon$. On the other hand, player 1 cannot almost-win from s_{hide}: if she never chooses *run*, she risks that player 2 always chooses *wait*, confining her in s_{hide}; if in any round she chooses *run* with positive probability, then the strategy of player 2 that chooses *throw* in that round causes her to lose with positive probability.

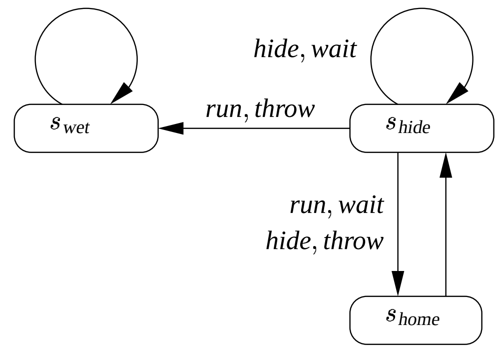

Figure 1: Game SKIRMISH

We provide algorithms that compute, given a concurrent probabilistic Rabin-chain game, the sets of sure-win, almost-sure-win, and limit-win states for each player. All three sets can be computed in time $n^{\mathcal{O}(m)}$, where n is the size of the game structure and m is the number of pairs in the Rabin-chain condition. While this complexity is in line with the solution for turn-based Rabin-chain games [EJ91], our algorithms are considerably more involved than those for turn-based games. This has several reasons.

First, while turn-based games are *determined* [BL69] (in every state, either player 1 has a "winning" strategy, which guarantees a win no matter what player 2 does, or player 2 has a "spoiling" strategy, which guarantees that the winning condition is violated no matter what player 1 does), concurrent games are generally not determined. For example, in the game MATCHBIT, neither does player 1 have a winning strategy nor does player 2 have a spoiling strategy. Instead of determinacy, concurrent games exhibit a more complex form of duality, which is based on the following modes of losing: *exist-loss* (player 2 has a strategy that guarantees that some possible outcome of the game is a loss), *positive-loss* (player 2 has a strategy that guarantees a loss with positive probability), and *bounded-loss* (player 2 has a strategy that guarantees a loss with probability ε for some $\varepsilon > 0$). We show that for all concurrent Rabin-chain games, the *sure-win* states are the complement of the *exist-loss* states, the *almost-sure-win* states are the complement of the *positive-loss* states, and the *limit-win* states are the complement of the *bounded-loss* states.

Second, in sharp contrast with turn-based games [BL69], the limit-winning strategies for a concurrent game may require an infinite amount of memory about the history of the game. This phenomenon occurs not with reachability but with Büchi games. Consider again the game SKIRMISH, together with the Büchi winning condition $\Box\Diamond s_{home}$. The limit-win states are s_{home} and s_{hide}. As explained above, for every ε, from s_{hide} player 1 can reach s_{home} with probability at least $1-\varepsilon$. However, if player 1 uses the same probability ε to choose *run* in every visit to s_{hide}, by always choosing *throw* player 2 can ensure that the probability of infinitely many visits to s_{home} is 0. The proof that there are no finite-memory (rather than memoryless) winning strategies follows a similar argument. On the other hand, for $\varepsilon > 0$, an infinite-memory strategy that ensures winning with probability at least $1-\varepsilon$ can be constructed as follows: for $k \geq 0$, let $\varepsilon_k = 1-(1-\varepsilon)^{-1/2^{k+1}}$, so that $\prod_{k=0}^{\infty}(1-\varepsilon_k) = 1-\varepsilon$; then, at s_{hide}, choose *run* with probability ε_k, where k is the number of prior visits to s_{home}. Thus, the construction of winning strategies for concurrent games often hinges on the analysis of the limit behavior of infinite-memory randomized strategies. In the paper, we provide a complete characterization of the types of winning and spoiling strategies needed for the various subclasses of concurrent games.

Third, the fact that both players can choose among several moves at a state breaks the standard recursive divide-and-conquer approach to the solution of turn-based Rabin-chain games [McN93, Tho95]. For example, the set of states from which player 1 cannot reach a goal no longer forms a proper subgame. Our algorithms are instead presented in symbolic form, using μ-calculus notation, which, as first remarked in [EJ91], offers a powerful tool for writing and analyzing algorithms that traverse state spaces. It also suggests a way for implementing the algorithms symbolically, potentially enabling the analysis of systems with large state spaces [BCM+90].

2 Games

For a finite set A, a *probability distribution* on A is a function $p\colon A \mapsto [0,1]$ such that $\sum_{a\in A} p(a) = 1$. We denote the set of probability distributions on A by $\mathcal{D}(A)$. Given a distribution $p \in \mathcal{D}(A)$, we denote by $\text{Supp}(p) = \{x \in A \mid p(x) > 0\}$ the *support* of p. A (two-player) *game structure* $\mathcal{G} = \langle S, \mathit{Moves}_1, \mathit{Moves}_2, \Gamma_1, \Gamma_2, p\rangle$ consists of the following components:

- A finite state space S.
- Two finite sets Moves_1, Moves_2 of moves. For convenience, we assume $\mathit{Moves}_1 \cap \mathit{Moves}_2 = \emptyset$.
- Two move assignments $\Gamma_i : S \mapsto 2^{\mathit{Moves}_i} \setminus \emptyset$, for $i \in \{1,2\}$. The assignment Γ_i associates with each state $s \in S$ the nonempty set $\Gamma_i(s) \subseteq \mathit{Moves}_i$ of moves available to player i at state s.
- A probabilistic transition function $p\colon S \times \mathit{Moves}_1 \times \mathit{Moves}_2 \mapsto \mathcal{D}(S)$, which associates with every state $s \in S$ and moves $a_1 \in \Gamma_1(s)$ and $a_2 \in \Gamma_2(s)$ a probability distribution $p(s,a_1,a_2) \in \mathcal{D}(S)$ for the successor state.

At every state $s \in S$, player 1 chooses a move $a_1 \in \Gamma_1(s)$, and simultaneously and independently player 2 chooses a move $a_2 \in \Gamma_2(s)$. The game then proceeds to the successor state t with probability $p(s,a_1,a_2)(t)$, for all $t \in S$. For all states $s \in S$ and moves $a_1 \in \Gamma_1(s)$ and $a_2 \in \Gamma_2(s)$, we indicate by $\delta(s,a_1,a_2) = \text{Supp}(p(s,a_1,a_2))$ the set of possible successors of s when moves a_1, a_2 are selected. A *path* of $\mathcal{G}$ is an infinite sequence $\overline{s} = s_0, s_1, s_2, \ldots$ of states in S such that for all $k \geq 0$, there are moves $a_1^k \in \Gamma_1(s_k)$ and $a_2^k \in \Gamma_2(s_k)$ such that $s_{k+1} \in \delta(s_k, a_1^k, a_2^k)$. We denote by Ω the set of all paths.

We define the *size* of the game structure $\mathcal{G}$ to be equal to the number of entries of the transition function δ; specifically, $|\mathcal{G}| = \sum_{s\in S} \sum_{a\in\Gamma_1(s)} \sum_{b\in\Gamma_2(s)} |\delta(s,a,b)|$. Note that this definition assumes that the probability numbers can be represented in constant space; this assumption is conservative with respect to the upper-bound complexity results that we present in the paper. We say that a game structure $\mathcal{G}$ is *turn-based* if at every state at most one player can choose among multiple moves; that is, for every state $s \in S$ there exists at most one $i \in \{1,2\}$ with $|\Gamma_i(s)| > 1$. We say that a game structure $\mathcal{G}$ is *deterministic* if $|\delta(s,a_1,a_2)| = 1$ for all $s \in S$ and all $a_1 \in \Gamma_1(s)$, $a_2 \in \Gamma_2(s)$.

2.1 Strategies

A *strategy* for player $i \in \{1,2\}$ is a mapping $\pi_i : S^+ \mapsto \mathcal{D}(\mathit{Moves}_i)$ that associates with every nonempty finite sequence $\sigma \in S^+$ of states, representing the past history of the game, a probability distribution $\pi_i(\sigma)$ used to select the next move. Thus, the choice of the next move can be history-dependent and randomized. For all sequences $\sigma \in S^*$ and states $s \in S$, we require that $\mathrm{Supp}(\pi_i(\sigma s)) \subseteq \Gamma_i(s)$. We denote by Π_i the set of all strategies for player $i \in \{1, 2\}$.

Given a state $s \in S$ and two strategies $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$, we define *Outcomes*$(s, \pi_1, \pi_2) \subseteq \Omega$ to be the set of paths that can be followed by the game, when the game starts from s and the players use the strategies π_1 and π_2. Formally, $s_0, s_1, s_2, \ldots \in$ *Outcomes*(s, π_1, π_2) if $s_0 = s$ and if for all $k \geq 0$ there exist moves $a_1^k \in \Gamma_1(s_k)$ and $a_2^k \in \Gamma_2(s_k)$ such that $\pi_1(s_0, \ldots, s_k)(a_1^k) > 0$, $\pi_2(s_0, \ldots, s_k)(a_2^k) > 0$, and $p(s_k, a_1^k, a_2^k)(s_{k+1}) > 0$. Once the starting state s and the strategies π_1 and π_2 for the two players have been chosen, they give rise to a probability space over the paths. Hence, the probabilities of events are uniquely defined, where an *event* $\mathcal{E} \subseteq \Omega$ is a measurable set of paths. For an event $\mathcal{E} \subseteq \Omega$, we denote by $\Pr_s^{\pi_1,\pi_2}(\mathcal{E})$ the probability that a path belongs to $\mathcal{E}$ when the game starts from s and the players use the strategies π_1 and π_2. We distinguish the following types of strategies:

- A strategy π for player $i \in \{1, 2\}$ is *deterministic* if for all $\sigma \in S^+$ there exists $a \in$ *Moves*$_i$ such that $\pi(\sigma)(a) = 1$.
- A strategy π is *memoryless* if $\pi(\sigma s) = \pi(s)$ for all $s \in S$ and all $\sigma \in S^*$.
- A strategy π is *finite-memory* if the distribution chosen at every state $s \in S$ depends only on s itself, and on a finite number of bits of information about the past history of the game.

We indicate with Π^D, Π^M, Π^F the classes of deterministic, memoryless, and finite-memory strategies; we let $\Pi^{DM} = \Pi^D \cap \Pi^M$, and we let Π^H (for *history-dependent*) be the class of all strategies.

2.2 Winning conditions

We consider *winning conditions* expressed by LTL formulas, whose atomic propositions correspond to subsets of states. We write $\overline{s} \models \varphi$ to denote the fact that a path $\overline{s}$ satisfies a winning condition φ. Given a winning condition φ, we denote by $[\![\varphi]\!] = \{\overline{s} \in \Omega \mid \overline{s} \models \varphi\}$ the set of paths that satisfy φ. For all initial states $s \in S$, the set of paths $\{s_0, s_1, s_2, \ldots \in [\![\varphi]\!] \mid s_0 = s\}$ is measurable [Var85]. Given an initial state $s \in S$ and a winning condition φ, we consider the following *winning modes* for player 1:

Sure. We say that player 1 *wins surely* if the player has a strategy to ensure that φ holds on every path, or $\exists \pi_1 \in \Pi_1 \,.\, \forall \pi_2 \in \Pi_2 \,.\, \textit{Outcomes}(s, \pi_1, \pi_2) \subseteq [\![\varphi]\!]$.

Almost. We say that player 1 *wins almost surely* if the player has a strategy to win with probability 1, or $\exists \pi_1 \in \Pi_1 \,.\, \forall \pi_2 \in \Pi_2 \,.\, \Pr_s^{\pi_1,\pi_2}([\![\varphi]\!]) = 1$.

Limit. We say that player 1 *wins limit surely* if the player has a strategy to win with probability arbitrarily close to 1, or $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \Pr_s^{\pi_1,\pi_2}([\![\varphi]\!]) = 1$.

Bounded. We say that player 1 *wins boundedly* if the player has a strategy to win with probability bounded away from 0, or $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \Pr_s^{\pi_1,\pi_2}([\![\varphi]\!]) > 0$.

Positive. We say that player 1 *wins positively* if the player has a strategy to win with positive probability, or $\exists \pi_1 \in \Pi_1 \,.\, \forall \pi_2 \in \Pi_2 \,.\, \Pr_s^{\pi_1,\pi_2}([\![\varphi]\!]) > 0$.

Existential. We say that player 1 *wins existentially* if the player has a strategy that ensures that at least one path satisfies φ, or $\exists \pi_1 \in \Pi_1 . \forall \pi_2 \in \Pi_2 . \textit{Outcomes}(s, \pi_1, \pi_2) \cap [\![\varphi]\!] \neq \emptyset$.

Analogous definitions apply for player 2. We abbreviate the winning modes by *sure*, *almost*, *limit*, *bounded*, *positive*, and *exist*. We call these winning modes the *qualitative* winning modes because they can be decided without resorting to numerical computation, as will be shown in the course of the paper. Using a notation derived from *alternating temporal logic* [AHK97], given a player $i \in \{1, 2\}$, a winning mode $\lambda \in \{\textit{sure}, \textit{almost}, \textit{limit}, \textit{bounded}, \textit{positive}, \textit{exist}\}$ and a winning condition φ, we denote by $\langle\langle i \rangle\rangle_\lambda\, \varphi$ the set of states from which player i can win in mode λ the game with winning condition φ. For each player $i \in \{1, 2\}$ and winning condition φ, the containments hold:

$$\langle\langle i \rangle\rangle_{\textit{sure}}\, \varphi \subseteq \langle\langle i \rangle\rangle_{\textit{almost}}\, \varphi \subseteq \langle\langle i \rangle\rangle_{\textit{limit}}\, \varphi$$
$$\subseteq \langle\langle i \rangle\rangle_{\textit{bounded}}\, \varphi \subseteq \langle\langle i \rangle\rangle_{\textit{positive}}\, \varphi \subseteq \langle\langle i \rangle\rangle_{\textit{exist}}\, \varphi \ .$$

In general this containment cannot be strengthened to equality, even for deterministic concurrent games [dAHK98]. We present two theorems that summarize the relations between these sets of winning states; the theorems follow from the algorithms and arguments presented in the later sections. The first theorem states that, for turn-based games, some inclusions can be strengthened to equalities.

Theorem 1 *For every LTL formula φ, and for $i \in \{1, 2\}$, the following assertions hold:*

1. *For probabilistic turn-based games, we have $\langle\langle i \rangle\rangle_{\textit{almost}}\, \varphi = \langle\langle i \rangle\rangle_{\textit{limit}}\, \varphi$.*
2. *For deterministic turn-based games, we have $\langle\langle i \rangle\rangle_{\textit{sure}}\, \varphi = \langle\langle i \rangle\rangle_{\textit{almost}}\, \varphi = \langle\langle i \rangle\rangle_{\textit{limit}}\, \varphi$.*

The first part of the theorem is a consequence of the results presented in the following sections; the second part follows from the determinacy results of [BL69] (except for the terminology). The second theorem expresses the duality between the winning conditions for players 1 and 2. For convenience, given subsets $B_1, B_2 \subseteq S$ of states, we write $\neg B_1$ for $S \setminus B_1$, and $B_1 \wedge B_2$, $B_1 \vee B_2$ for $B_1 \cap B_2$, $B_1 \cup B_2$, respectively.

Theorem 2 (Duality Theorem) *For every LTL formula φ, we have:*

$$\langle\!\langle 1 \rangle\!\rangle_{\mathit{sure}}\, \varphi = \neg \langle\!\langle 2 \rangle\!\rangle_{\mathit{exist}}\, \neg\varphi$$
$$\langle\!\langle 1 \rangle\!\rangle_{\mathit{almost}}\, \varphi = \neg \langle\!\langle 2 \rangle\!\rangle_{\mathit{positive}}\, \neg\varphi$$
$$\langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \varphi = \neg \langle\!\langle 2 \rangle\!\rangle_{\mathit{bounded}}\, \neg\varphi\ .$$

For a mode $\lambda \in \{\mathit{sure}, \mathit{almost}, \mathit{limit}, \mathit{bounded}, \mathit{positive}, \mathit{exist}\}$, define the *dual* $\bar{\lambda}$ of λ by $\overline{\mathit{sure}} = \mathit{exist}$, $\overline{\mathit{almost}} = \mathit{positive}$, $\overline{\mathit{limit}} = \mathit{bounded}$, and $\bar{\bar{\lambda}} = \lambda$. In view of the duality theorem, in this paper we will focus on algorithms for computing the winning states with respect to the winning conditions *sure*, *almost*, and *limit*.

In the following sections, we consider winning conditions that consist in safety and reachability properties, and on winning conditions that correspond to the accepting criteria of Büchi, co-Büchi, and Rabin-chain (or parity) automata [Mos84, Tho90]. We call games with such winning conditions *safety, reachability, Büchi, co-Büchi,* and *Rabin-chain games,* respectively. We remark that the ability of solving games with Rabin-chain winning conditions suffices for solving games with respect to arbitrary ω-regular winning conditions. In fact, we can encode a general ω-regular condition as a deterministic Rabin-chain automaton. By taking the synchronous product of the automaton and the original game, we obtain an (enlarged) game with a Rabin-chain winning condition [Tho95, LW95, KPBV95, BLV96]. The set of winning states of the original structure can be computed by computing the set of winning states of this enlarged game.

2.3 Winning and spoiling strategies

For $\lambda \in \{\mathit{sure}, \mathit{almost}, \mathit{positive}, \mathit{exist}\}$, a λ-*winning strategy* is a strategy for player 1 that realizes the existential quantifier in the definitions of sure, almost, positive, and existential winning modes for all $s \in \langle\!\langle 1 \rangle\!\rangle_\lambda\, \varphi$. A *limit-winning family of strategies* for φ is a family $\{\pi_1[\varepsilon] \mid \varepsilon > 0\}$ of strategies for player 1 such that for all reals $\varepsilon > 0$, all states $s \in \langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \varphi$, and all strategies π_2 of player 2, we have $\Pr_s^{\pi_1[\varepsilon],\pi_2}([\![\varphi]\!]) \geq 1-\varepsilon$. A *bounded-winning strategy* for φ is a strategy π_1 such that, for some $\varepsilon > 0$, we have $\Pr_s^{\pi_1,\pi_2}([\![\varphi]\!]) > \varepsilon$ for all strategies π_2 of player 2. A *spoiling strategy* for player 2 for condition φ and mode $\lambda \in \{\mathit{sure}, \mathit{almost}, \mathit{limit}, \mathit{bounded}, \mathit{positive}, \mathit{exist}\}$ is a (family of) winning strategies for player 2, for condition $\neg\varphi$, and for mode $\bar{\lambda}$. We are interested in determining the smallest class of strategies in which winning and spoiling strategies are guaranteed to exist, with respect to the inclusions $\Pi^{DM} \subseteq \Pi^M \subseteq \Pi^F \subseteq \Pi^H$. The following lemma will be used to show that, for several types of games, finite-memory spoiling strategies for *almost* mode may not exist. The lemma is proved by noting that a game under a fixed finite-memory strategy is equivalent to a Markov decision process, and using results from [CY90, BdA95].

Lemma 1 *Consider any concurrent game $\mathcal{G}$ and any LTL formula φ. If player 2 has a finite-memory spoiling strategy for mode almost and for φ, then $\langle\!\langle 1 \rangle\!\rangle_{\mathit{almost}}\, \varphi = \langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \varphi$.*

3 Safety and Reachability Games

In this section we summarize some results on concurrent safety and reachability games from [dAHK98]. The presentation is rephrased significantly, in a framework that allows us to extend the results to Büchi, co-Büchi, and Rabin-chain winning conditions.

3.1 Safety games

The winning condition of a *safety game* is a formula of the form $\Box B$, where $B \subseteq S$ is a subset of states. To solve these games, we use the *controllable predecessor* operator $\mathrm{Pre}_1 : 2^S \mapsto 2^S$, defined for all $s \in S$ and $X \subseteq S$ by:

$$s \in \mathrm{Pre}_1(X) \quad \mathit{iff} \quad \exists a \in \Gamma_1(s). \forall b \in \Gamma_2(s). \delta(s,a,b) \subseteq X. \tag{1}$$

The subscript 1 of Pre_1 indicates that the predecessor operator refers to player 1; we can define Pre_2 by exchanging the subscripts 1 and 2 in (1). The set of winning states can be computed by the μ-calculus expression $\langle\!\langle 1 \rangle\!\rangle_{\mathit{sure}}\, \Box B = \langle\!\langle 1 \rangle\!\rangle_{\mathit{almost}}\, \Box B = \langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \Box B = \nu X\, .\, (\mathrm{Pre}_1(X) \wedge B)$. The following theorem summarizes the results on safety games [dAHK98].

Theorem 3 *The following assertions hold.*

1. *$\langle\!\langle 1 \rangle\!\rangle_{\mathit{sure}}\, \Box B = \langle\!\langle 1 \rangle\!\rangle_{\mathit{almost}}\, \Box B = \langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \Box B$, and for $\lambda \in \{\mathit{sure}, \mathit{almost}, \mathit{limit}\}$, we have $\langle\!\langle 1 \rangle\!\rangle_\lambda\, \Box\varphi = \langle\!\langle 2 \rangle\!\rangle_{\bar{\lambda}}\, \Diamond\neg\varphi$.*
2. *The complexity of computing $\langle\!\langle 1 \rangle\!\rangle_{\mathit{sure}}\, \Box\varphi$, and the most restrictive classes in which winning and spoiling strategies are guaranteed to exist are given in Table 1(a).*

The predecessor operator Pre has been defined on the basis of the moves available to both players. However, according to our definition of strategy, at each state a player chooses a *distribution* over the moves, rather than a single move. Hence, it is natural to rephrase the definition in terms of distributions over moves; such a formulation

	Complexity	Winning	Spoiling
Sure	$\mathcal{O}(n)$	Π^{DM}	Π^{M}
Almost	$\mathcal{O}(n)$	Π^{DM}	Π^{M}
Limit	$\mathcal{O}(n)$	Π^{DM}	Π^{M}

(a) Safety games.

	Complexity	Winning	Spoiling
Sure	$\mathcal{O}(n)$	Π^{DM}	Π^{M}
Almost	$\mathcal{O}(n^2)$	Π^{M}	Π^{H}
Limit	$\mathcal{O}(n^2)$	Π^{M}	Π^{M}

(b) Reachability games.

	Complexity	Winning	Spoiling
Sure	$\mathcal{O}(n^2)$	Π^{DM}	Π^{M}
Almost	$\mathcal{O}(n^2)$	Π^{M}	Π^{H}
Limit	$\mathcal{O}(n^2)$	Π^{H}	Π^{M}

(c) Büchi games.

	Complexity	Winning	Spoiling
Sure	$\mathcal{O}(n^2)$	Π^{DM}	Π^{M}
Almost	$\mathcal{O}(n^3)$	Π^{M}	Π^{H}
Limit	$\mathcal{O}(n^3)$	Π^{M}	Π^{H}

(d) Co-Büchi games.

	Complexity	Winning	Spoiling
Sure	$\mathcal{O}(n^{2m})$	Π^{DM}	Π^{M}
Almost	$\mathcal{O}(n^{2m+1})$	Π^{H}	Π^{H}
Limit	$\mathcal{O}(n^{2m+1})$	Π^{H}	Π^{H}

(e) Rabin-chain games.

Table 1: Upper bounds for the time complexity of solving ω-regular games, and types of winning and spoiling strategies; n is the size of the game and, in Rabin-chain games, m is the number of accepting pairs.

will also generalize to the predecessor operators required to solve other types of games, such as reachability games. For $s \in S$, $X \subseteq S$, $\xi_1 \in \mathcal{D}(\Gamma_1(s))$, and $\xi_2 \in \mathcal{D}(\Gamma_2(s))$ we denote by

$$P_s^{\xi_1,\xi_2}(X) = \sum_{a\in\Gamma_1(s)} \sum_{b\in\Gamma_2(s)} \sum_{t\in X} \xi_1(a)\,\xi_2(b)\,p(s,a,b)(t)$$

the one-round probability of a transition into X when players 1 and 2 play at s with distributions ξ_1 and ξ_2, respectively. With this notation, for all $X \subseteq S$ the definition of Pre can be rephrased as

$$s \in \text{Pre}_1(X) \quad \textit{iff}$$
$$\exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \,.\, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \,.\, P_s^{\xi_1,\xi_2}(X) = 1 \,.$$

3.2 Reachability games

The winning condition of a *reachability game* is an eventuality formula $\Diamond B$, where $B \subseteq S$ is a subset of states. Reachability games are more complex than safety games, since the sets of sure, almost, and limit-winning states do not coincide [dAHK98]. To compute these sets of winning states, we introduce three predecessor operators $\text{Spre}, \text{Apre}, \text{Lpre} : 2^S \times 2^S \mapsto 2^S$, corresponding to the winning modes *sure*, *almost*, and *limit*. These predecessor operators are defined in Table 2. We can define symmetrical operators for player 2 by exchanging the subscripts 1 and 2 in the definitions. We note that for all $X, Y \subseteq S$ we have $\text{Spre}_i(Y, X) = \text{Pre}_i(X)$ for $i \in \{1, 2\}$; the notation $\text{Spre}_i(Y, X)$ has been introduced only for notational uniformity. For mode $\lambda \in \{\textit{sure}, \textit{almost}, \textit{limit}\}$ and condition $\Diamond B$, where $B \subseteq S$, the set of winning states can be computed by:

$$\langle\!\langle 1 \rangle\!\rangle_\lambda \Diamond B = \nu Y \,.\, \mu X \,.\, (\lambda\text{pre}_1(Y, X) \vee B) \qquad (2)$$

where λpre is Spre if $\lambda = \textit{sure}$, Apre if $\lambda = \textit{almost}$, and is Lpre if $\lambda = \textit{limit}$. Using the equality $\text{Spre}_1(Y, X) = \text{Pre}_1(X)$ for all X, Y, in the case of sure reachability (2) reduces to $\langle\!\langle 1 \rangle\!\rangle_{\textit{sure}} \Diamond B = \mu X \,.\, (\text{Pre}_1(X) \vee B)$, which is the standard formula for turn-based reachability games [TW68]. Except for the notation, these algorithms are equivalent to those given in [dAHK98].

Intuitively, the algorithms can be understood as follows. For the mode *sure*, formula $\mu X \,.\, (\text{Pre}_1(X) \vee B)$ computes at each iteration i the set of states that can reach B surely in at most i rounds. For the mode *almost*, formula (2) states the existence of a set $Y = \langle\!\langle 1 \rangle\!\rangle_{\textit{almost}} \Diamond B$, and of a series of sets $B = X_1 \subset X_2 \subset \cdots \subset X_k = Y$. For $1 < i \leq k$, the set X_i is obtained by $X_i = \text{Apre}_1(Y, X_{i-1})$. The operator $\text{Apre}_1(Y, X_{i-1})$ states that player 1 can play a distribution over moves that ensures that Y is not left, and that with some probability X_{i-1} is entered. Hence, from every state of Y there is a positive probability of reaching B in at most k rounds, and since Y is never left, the probability of eventually reaching B is 1. The algorithm for mode *limit* is similar, except

$$s \in \text{Spre}_1(Y, X) \quad \textit{iff} \quad \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \,.\, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \,.\, P_s^{\xi_1,\xi_2}(X) = 1$$

$$s \in \text{Apre}_1(Y, X) \quad \textit{iff} \quad \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \,.\, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \,.\, [P_s^{\xi_1,\xi_2}(Y) = 1 \wedge P_s^{\xi_1,\xi_2}(X) > 0]$$

$$s \in \text{Lpre}_1(Y, X) \quad \textit{iff} \quad \forall \alpha > 0 \,.\, \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \,.\, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \,.\, P_s^{\xi_1,\xi_2}(X) > \alpha P_s^{\xi_1,\xi_2}(\neg Y)$$

Table 2: Definition of the predecessor operators Spre_1, Apre_1, and Lpre_1, for $s \in S$ and $X, Y \subseteq S$.

that the operator $\text{Lpre}_1(Y, X_{i-1})$ relaxes the condition of $\text{Apre}_1(Y, X_{i-1})$ by allowing a probability of escape from Y, provided the probability of progress to X_{i-1} can be made arbitrarily larger than the probability of escape from Y. This arbitrarily large ratio accounts for being able to reach B with probability arbitrarily close to 1.

The operator Spre can be computed like Pre. The computation of the operators Apre and Lpre is presented in Section 6. The computation also enables the derivation of the winning and spoiling (families of) strategies; we omit the details due to space limitations. The following theorem, from [dAHK98], summarizes the results for reachability games.

Theorem 4 *The following assertions hold.*

1. *For $B \subseteq S$ and $\lambda \in \{sure, almost, limit\}$, we have $\langle\langle 1 \rangle\rangle_\lambda \diamond B = \neg \langle\langle 2 \rangle\rangle_{\bar{\lambda}} \,\square \neg B$.*
2. *The complexity of computing the sets of winning states, and the most restrictive classes in which winning and spoiling strategies are guaranteed to exist are given in Table 1(b).*

4 Büchi and Co-Büchi Games

4.1 Büchi games

The winning condition of a *Büchi game* is a formula $\square\diamond B$, where $B \subseteq S$ is a subset of states. For $\lambda \in \{sure, almost, limit\}$ and $B \subseteq S$, the set of winning states can be computed as:

$$\langle\langle 1 \rangle\rangle_\lambda \,\square\diamond B = \nu Y \,.\, \mu X \,.\, \big[(\neg B \wedge \lambda\text{pre}_1(Y, X)) \vee (B \wedge \text{Pre}_1(Y))\big] \,. \tag{3}$$

For $\lambda = sure$, this expression reduces to $\langle\langle 1 \rangle\rangle_{sure} \,\square\diamond B = \nu Y \,.\, \mu X \,.\, [(\neg B \wedge \text{Pre}_1(X)) \vee (B \wedge \text{Pre}_1(Y))]$, which coincides with the solution of [EJ91]. For the mode *almost*, formula (3) states the existence of a set $Y = \langle\langle 1 \rangle\rangle_{almost} \,\square\diamond B$, and of a series of sets $X_1 \subset X_2 \subset \cdots \subset X_k = Y$. Since $\lambda\text{pre}_1(Y, \emptyset) = \emptyset$, we have $X_1 \subseteq B$; from X_1, the operator $\text{Pre}_1(Y)$ ensures that Y is not left. For $1 < i \leq k$, the operator $\text{Apre}_1(Y, X_{i-1})$ ensures that Y is not left, and that with some probability, from X_i we proceed to X_{i-1}. Hence, from every state of Y there is a positive probability of reaching B in at most k rounds. Since Y is never left, the probability of eventually reaching B infinitely often is 1. The algorithm for mode *limit* is similar, except that for $1 < i \leq k$ the operator $\text{Lpre}_1(Y, X_{i-1})$ relaxes the condition of $\text{Apre}_1(Y, X_{i-1})$ by allowing a probability of escape from Y, provided the probability of progress to X_{i-1} can be made arbitrarily larger than the probability of escape from Y. We do not need to relax the condition for X_1, writing for instance $B \wedge \text{Lpre}_1(Y, Y)$. In fact, we have $\text{Lpre}_1(Y, Y) = \text{Pre}_1(Y)$: if the probability of leaving Y can be made arbitrarily small, then $\text{Pre}_1(Y)$ holds. The following theorem summarizes the results for Büchi games.

Theorem 5 *The following assertions hold.*

1. *For $B \subseteq S$ and $\lambda \in \{sure, almost, limit\}$, we have $\langle\langle 1 \rangle\rangle_\lambda \,\square\diamond B = \neg \langle\langle 2 \rangle\rangle_{\bar{\lambda}} \diamond\square \neg B$.*
2. *The complexity of computing the sets of winning states, and the most restrictive classes in which winning and spoiling strategies are guaranteed to exist are given in Table 1(c).*

The winning and spoiling strategies can be constructed on the basis of the μ-calculus expression (3) and of the predecessor operators appearing in it; we omit the constructions due to lack of space.

4.2 Co-Büchi games

The winning condition of a *co-Büchi game* is a formula of the form $\diamond\square B$, where $B \subseteq S$ is a subset of states. The solution for sure co-Büchi games coincides with the solution for deterministic turn-based games [EJ91]: $\langle\langle 1 \rangle\rangle_{sure} \diamond\square B = \mu X \,.\, \nu Y \,.\, [(B \wedge \text{Pre}_1(Y)) \vee (\neg B \wedge \text{Pre}_1(X))]$.

To gain some intuition about co-Büchi games for modes *almost* and *limit*, consider the games depicted in Figure 2. State s_1 is the only state at which the players can choose among more than one move; in both games we have $\Gamma_1(s_1) = \{a, b, c\}$ and $\Gamma_2(s_1) = \{d, e, f\}$; the winning condition is $\diamond\square\{s_0, s_1, s_2\}$. Intuitively, from state s_1, there are three types of transitions: *success transitions*

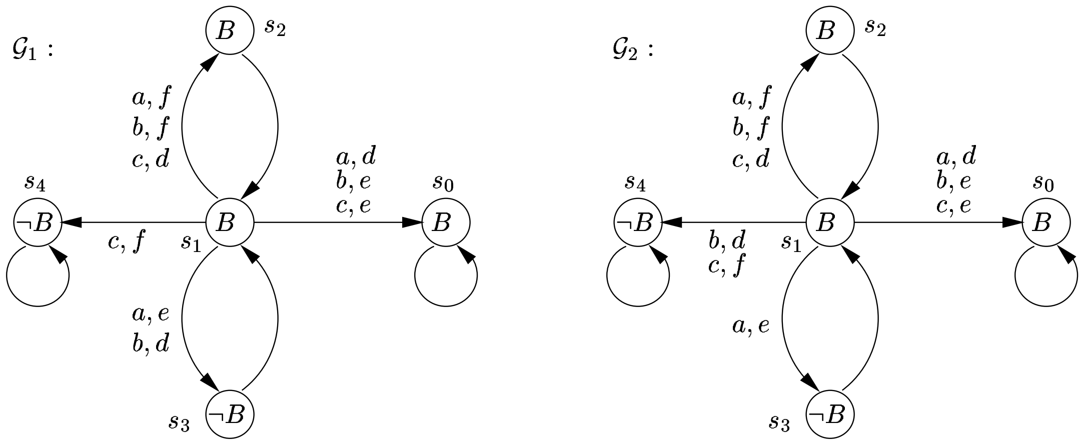

Figure 2: Co-Büchi games $\mathcal{G}_1$ and $\mathcal{G}_2$. State s_1 is the only state at which the players can choose among more than one move; in both games we have $\Gamma_1(s_1) = \{a, b, c\}$ and $\Gamma_2(s_1) = \{d, e, f\}$. The winning condition is $\Diamond\Box B$, where $B = \{s_0, s_1, s_2\}$.

to s_0, *failure transitions* to s_4, and *nuisance transitions* to s_3. Failure transitions cause player 1 to lose; success transition bring player 1 closer to winning; nuisance transitions cause player 1 to lose only if they are repeated infinitely often. A transition to $\{s_1, s_2\}$ is neutral: the outcome of the game depends on whether failure transitions, or infinitely many nuisance transitions, occur.

In game $\mathcal{G}_1$, player 1 can win with probability 1 from s_1 by playing moves a and b with probability 1/2 each. In fact, if player 1 uses this distribution, then (a) no move of player 2 can cause a failure transition, and (b) the probability of a success transition is proportional to that of a nuisance transition, so that the probability of infinitely many visits to s_3 is 0. On the other hand, in game $\mathcal{G}_2$ player 1 cannot win with probability 1 from s_1. In fact, to avoid failure player 1 must play move a deterministically; player 2 can then force infinitely many visits to s_3 by playing move e.[1] Nevertheless, player 1 can win game $\mathcal{G}_2$ from s_1 with probability arbitrarily close to 1 by playing move a with probability $1 - \varepsilon$ and move b with probability ε, and letting $\varepsilon \to 0$. This distribution ensures that (a) the probability of success can be made arbitrarily larger than the probability of failure by choosing ε, and (b) for every ε, the probability of success is proportional to the probability of nuisance. Part (b) ensures that the probability of infinitely many nuisance transitions is 0; part (a) ensures that the probability of success can be made arbitrarily close to 1.

The solution formulas (2) and (3) for reachability and Büchi games involved two-argument predecessor operators, such as $\mathrm{Lpre}_1(Y, X)$. In fact, the algorithms for solving these games need to consider only two types of transitions: success transitions (to X) and failure ones (to $\neg Y$). To solve co-Büchi games, we need predecessor operators that take three arguments $X, Y, Z \subseteq S$: a successful transition is one that enters X; a failure transition is one that leaves Z, and a nuisance transition is one that leaves Y. Corresponding to *almost* and *limit* modes, we introduce the operators $\mathrm{AFpre}_1, \mathrm{LFpre}_1 : 2^S \times 2^S \times 2^S \mapsto 2^S$. The definitions reflect the previous analysis. For all $s \in S$ and $X, Y, Z \subseteq S$, we let:

$$
\begin{aligned}
& s \in \mathrm{AFpre}_1(Z, Y, X) \textit{ iff} \\
& \qquad \exists \beta \in \mathbb{R}_{>0} \, . \\
& \qquad \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \, . \, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \, . \\
& \qquad \left(\begin{array}{c} P_s^{\xi_1, \xi_2}(Z) = 1 \\ \wedge \\ P_s^{\xi_1, \xi_2}(X) \geq \beta P_s^{\xi_1, \xi_2}(\neg Y) \end{array} \right)
\end{aligned}
$$

$$
\begin{aligned}
& s \in \mathrm{LFpre}_1(Z, Y, X) \textit{ iff} \\
& \qquad \forall \alpha \in \mathbb{R}_{>0} \, . \, \exists \beta \in \mathbb{R}_{>0} \, . \\
& \qquad \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \, . \, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \, . \\
& \qquad \left(\begin{array}{c} P_s^{\xi_1, \xi_2}(X) \geq \alpha P_s^{\xi_1, \xi_2}(\neg Z) \\ \wedge \\ P_s^{\xi_1, \xi_2}(X) \geq \beta P_s^{\xi_1, \xi_2}(\neg Y) \end{array} \right)
\end{aligned}
$$

For $B \subseteq S$ and mode $\lambda \in \{\textit{sure}, \textit{almost}, \textit{limit}\}$, the set of winning states of co-Büchi games is given by:

$$
\langle\!\langle 1 \rangle\!\rangle_\lambda \, \Diamond\Box B = \nu Z . \mu X . \nu Y . \left(\begin{array}{c} (B \wedge \lambda\mathrm{Fpre}_1(Z, Y, X)) \\ \vee \\ (\neg B \wedge \lambda\mathrm{pre}_1(Z, X)) \end{array} \right) \tag{4}
$$

[1] An infinite-memory spoiling policy for player 2 consists in playing at s_1 moves d and f with probability $(1/2)^{(1+1/2^{k+1})}$ each, and move e with probability $1 - (1/2)^{1/2^{k+1}}$, where k is the number of previous visits to s_1.

where $\lambda\text{Fpre}_1(Z,Y,X) = \text{Pre}_1(Y)$ if $\lambda = \textit{sure}$, $\lambda\text{Fpre}_1(Z,Y,X) = \text{AFpre}_1(Z,Y,X)$ if $\lambda = \textit{almost}$, and $\lambda\text{Fpre}_1(Z,Y,X) = \text{LFpre}_1(Z,Y,X)$ if $\lambda = \textit{limit}$. For $\lambda = \textit{sure}$, the above formula reduces to the solution for deterministic turn-based games of [EJ91].

Informally, algorithm (4) can be understood as follows. For the mode *almost*, we can write the fixpoint Z^* of (4) as an increasing sequence of sets $X_1 \subset \cdots \subset X_k = Z^*$ obtained as follows. Since $\text{AFpre}_1(Z^*, Y, \emptyset) = \text{Pre}_1(Y)$ and $\text{Apre}_1(Z^*, \emptyset) = \emptyset$ for all $Y \subseteq S$, the first set is $X_1 = \nu Y \,.\, [B \wedge \text{Pre}_1(Y)] = \langle\!\langle 1 \rangle\!\rangle_{\textit{sure}} \Box B$. For $1 \leq i < k$, the set X_{i+1} is obtained from X_i in one of two ways:

- either $X_{i+1} = X_i \cup \{s_i\}$,
  where $s_i \notin B$ and $s_i \in \text{Apre}_1(Z^*, X_i)$;
- or $X_{i+1} = X_i \cup Y_i$,
  where $Y_i \subseteq B$ and $Y_i \subseteq \text{AFpre}_1(Z^*, Y_i, X_i)$.

If X_{i+1} is obtained in the first way, then at s_i player 1 can avoid leaving Z^* while proceeding with positive probability to X_i. If X_{i+1} is obtained in the second way, then from $Y_i \subseteq B$ player 1 can avoid leaving Z^*, while going to X_i with probability proportional to that of leaving Y_i. Putting these two observations together, we can show by induction on i, from $i = k-1$ downto $i = 0$, that

$$Z^* \setminus X_i \subseteq \langle\!\langle 1 \rangle\!\rangle_{\textit{almost}} (\Diamond\Box(B \wedge (Z^* \setminus X_i)) \vee \Diamond X_i).$$

This shows that $Z^* \subseteq \langle\!\langle 1 \rangle\!\rangle_{\textit{almost}} \Diamond\Box B$, which is one direction of (4). The proof of the reverse containment is based on the complementation of the μ-calculus formula (4), and is omitted due to space constraints.

For the mode *limit*, we again write the fixpoint Z^* of (4) as an increasing sequence of sets $X_1 \subset X_2 \subset \cdots \subset X_k = Z^*$ with $X_1 = \langle\!\langle 1 \rangle\!\rangle_{\textit{sure}} \Box B$. For $1 \leq i < k$, set X_{i+1} is obtained from X_i in one of the two ways described for mode *almost*, except that Apre is replaced by Lpre and AFpre is replaced by LFpre. Proceeding similarly to mode *almost*, we can show by induction on i that $Z^* \setminus X_i \subseteq \langle\!\langle 1 \rangle\!\rangle_{\textit{limit}} (\Diamond\Box(B \wedge (Z^* \setminus X_i)) \vee \Diamond X_i)$. This leads to $Z^* \subseteq \langle\!\langle 1 \rangle\!\rangle_{\textit{limit}} \Diamond\Box B$; the proof of the reverse inclusion is again based on the complementation of the μ-calculus expression (4). The computation of operators AFpre and LFpre is given in Section 6. The following theorem summarizes the results for co-Büchi games.

Theorem 6 *The following assertions hold.*

1. *For $B \subseteq S$ and $\lambda \in \{\textit{sure}, \textit{almost}, \textit{limit}\}$, we have $\langle\!\langle 1 \rangle\!\rangle_{\lambda} \Diamond\Box B = \neg\langle\!\langle 2 \rangle\!\rangle_{\bar{\lambda}} \Box\Diamond\neg B$.*
2. *The complexity of computing the sets of winning states, and the most restrictive classes in which winning and spoiling strategies are guaranteed to exist are given in Table 1(d).*

5 Rabin-Chain Games

The winning condition of a *Rabin-chain game* is a formula

$$\rho = \bigvee_{i=0}^{m-1} (\Box\Diamond U_{2i} \wedge \neg\Box\Diamond U_{2i+1}) \,, \qquad (5)$$

where $m \geq 0$ is the number of *accepting pairs*, and $\emptyset = U_{2m} \subseteq U_{2m-1} \subseteq U_{2m-2} \subseteq \cdots \subseteq U_0 = S$ [Mos84]. Perhaps a more intuitive characterization of ρ is that of a *parity game* [EJ91]. For $0 \leq i \leq 2m-1$, let $B_i = U_i \setminus U_{i+1}$ be the set of states of *color* i. The total number of colors is $2m$. Given a path $\bar{s}$, let $\textit{Infi}(\bar{s}) \subseteq S$ be the set of states that occur infinitely often along $\bar{s}$, and let $\textit{MaxCol}(\bar{s}) = \max\{i \in \{0, \ldots, 2m-1\} \mid B_i \cap \textit{Infi}(\bar{s}) \neq \emptyset\}$ be the largest color appearing infinitely often along the path. Then, we have $[\![\rho]\!] = \{\bar{s} \in \Omega \mid \textit{MaxCol}(\bar{s}) \text{ is even}\}$. For the winning mode $\lambda = \textit{sure}$, the set of winning states can be computed using the formula of [EJ91]:

$$\langle\!\langle 1 \rangle\!\rangle_{\textit{sure}}\, \rho = \mu X_{2m-1} \,.\, \nu Y_{2m-2} \,.\, \cdots \mu X_1 \,.\, \nu Y_0 \,.\, \big((B_0 \wedge \text{Pre}_1(Y_0)) \vee \cdots \vee (B_{2m-1} \wedge \text{Pre}_1(X_{2m-1}))\big).$$

For the winning modes *almost* and *limit*, the solution of a Rabin-chain game is obtained by recursively nesting the solution of the algorithm for co-Büchi games. In fact, the co-Büchi accepting condition corresponds to a Rabin-chain game with only two colors, B_0 and B_1. In the nesting, there is one instance of the co-Büchi algorithm for each accepting pair $0 \leq i < m$, corresponding to the colors B_{2i} and B_{2i+1}. The nesting involves not only the μ-calculus expression (4), but also the definition of the predecessor operators. To understand this latter nesting, which is a key point of the algorithm, let us revisit the definition of operator LFpre_1. For $X, Y, Z \subseteq S$, the operator $\text{LFpre}_1(Z, Y, X)$ consists of two conditions, hierarchically combined: at the top level, the condition $\langle \alpha, Z, X \rangle$; at the next (and bottom) level, the condition $\langle \beta, Y, X \rangle$. These conditions have the following meaning:

- The condition $\langle \alpha, Z, X \rangle$ requires that the probability of entering X can be made arbitrarily larger than the probability of leaving Z, and that the next lower-level condition is also satisfied. Furthermore, if there are no lower-level conditions, then the probability of going to X must be positive.
- The condition $\langle \beta, Y, X \rangle$ requires that either the probability of entering X is some non-zero fraction of the probability of leaving Y or, if there are lower-level conditions, that the next lower-level condition is satisfied.

We write the hierarchical combination of conditions as a sequence, starting with the top-level condition, and proceeding in order of decreasing level. We will have:

$$\text{LFpre}_1(Z, Y, X) = \langle \alpha, Z, X \rangle \odot \langle \beta, Y, X \rangle \quad (6)$$

$$\text{Lpre}_1(Y, X) = \langle \alpha, Y, X \rangle \quad (7)$$

for all $X, Y \subseteq S$. In order to define the operator AFpre_1, we define also a γ-triple, for all $Z, X \subseteq S$:

- The condition $\langle \gamma, Z, X \rangle$ requires that the probability of leaving Z is 0, and that the next lower-level condition is satisfied. Furthermore, if there are no lower-level conditions, then the probability of going to X must be positive.

With this definition, for all $Z, Y, X \subseteq S$ we will have:

$$\text{Apre}_1(Y, X) = \langle \gamma, Y, X \rangle \quad (8)$$

$$\text{AFpre}_1(Z, Y, X) = \langle \gamma, Z, X \rangle \odot \langle \beta, Y, X \rangle. \quad (9)$$

Formally, to define the meaning of a sequence θ of α, β, and γ-triples, we define separately the *quantifier prefix* $\mathcal{Q}(\theta)$ and the *quantifier-free part* $\mathcal{P}(\theta)$ of θ. The meaning of the sequence θ is then defined by:

$$s \in \theta \textit{ iff } \mathcal{Q}(\theta) \,.\, \exists \xi_1 \in \mathcal{D}(\Gamma_1(s)) \,.\, \forall \xi_2 \in \mathcal{D}(\Gamma_2(s)) \,.\, \mathcal{P}(\theta).$$

The quantifier prefix $\mathcal{Q}(\theta)$ is defined as follows:

$$\mathcal{Q}(\emptyset) = \emptyset$$
$$\mathcal{Q}(\langle \alpha, Y, X \rangle \odot \theta) = (\forall \alpha_X > 0 \,.\, \mathcal{Q}(\theta))$$
$$\mathcal{Q}(\langle \beta, Y, X \rangle \odot \theta) = (\exists \beta_X > 0 \,.\, \mathcal{Q}(\theta))$$
$$\mathcal{Q}(\langle \gamma, Y, X \rangle \odot \theta) = \mathcal{Q}(\theta)$$

The quantifier-free part $\mathcal{P}(\theta)$ is defined as follows:

$$\mathcal{P}(\langle \alpha, Y, X \rangle) = (P_s^{\xi_1,\xi_2}(X) > \alpha_X P_s^{\xi_1,\xi_2}(\neg Y))$$
$$\mathcal{P}(\langle \beta, Y, X \rangle) = (P_s^{\xi_1,\xi_2}(X) \geq \beta_X P_s^{\xi_1,\xi_2}(\neg Y))$$
$$\mathcal{P}(\langle \gamma, Y, X \rangle) = (P_s^{\xi_1,\xi_2}(Y) = 1 \wedge P_s^{\xi_1,\xi_2}(X) > 0)$$

$$\mathcal{P}(\langle \alpha, Y, X \rangle \odot \theta) = \left(\begin{array}{c} P_s^{\xi_1,\xi_2}(X) \geq \alpha_X P_s^{\xi_1,\xi_2}(\neg Y) \\ \wedge\, \mathcal{P}(\theta) \end{array} \right)$$

$$\mathcal{P}(\langle \beta, Y, X \rangle \odot \theta) = \left(\begin{array}{c} P_s^{\xi_1,\xi_2}(X) > \beta_X P_s^{\xi_1,\xi_2}(\neg Y) \\ \vee\, \mathcal{P}(\theta) \end{array} \right)$$

$$\mathcal{P}(\langle \gamma, Y, X \rangle \odot \theta) = \left(\begin{array}{c} P_s^{\xi_1,\xi_2}(Y) = 1 \\ \wedge\, \mathcal{P}(\theta) \end{array} \right).$$

In our algorithms, when the triple $\langle \beta, Y, X \rangle$ occurs before the end of the sequence, it will always be followed by the triple $\langle \alpha, Y, X' \rangle$, for some $X' \subseteq S$. Hence, either the probability of X is a non-zero fraction of the probability of leaving Y, or else the following triple $\langle \alpha, Y, X' \rangle$ ensures that the probability of leaving Y is either 0, or can be made arbitrarily small.

5.1 Two-pair Rabin-chain games

Before presenting the solution for general Rabin-chain games, we present the solution for two-pair Rabin-chain games; the general solution will be a straightforward generalization of the two-pair solution. In these games, we have $m = 2$ and the colors are B_0, B_1, B_2, and B_3. For mode *limit*, the set of winning states is given by:

$$\langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \rho = \nu Z . \mu X_3 . \nu Y_2 . \mu X_1 . \nu Y_0 . \quad (10)$$
$$\left(\begin{array}{c} B_3 \wedge \langle \alpha, Z, X_3 \rangle \\ \vee \\ B_2 \wedge \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \\ \vee \\ B_1 \wedge \left(\begin{array}{l} \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \\ \odot \langle \alpha, Y_2, X_1 \rangle \end{array} \right) \\ \vee \\ B_0 \wedge \left(\begin{array}{l} \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \\ \odot \langle \alpha, Y_2, X_1 \rangle \odot \langle \beta, Y_0, X_1 \rangle \end{array} \right) \end{array} \right).$$

The solution for mode *almost* is similar, except that the triple $\langle \alpha, Z, X_3 \rangle$ is replaced with $\langle \gamma, Z, X_3 \rangle$ (the other α-triple is unchanged). In order to analyze this solution, it is convenient to write it in the following form:

$$\langle\!\langle 1 \rangle\!\rangle_{\mathit{limit}}\, \rho = \nu Z . \mu X_3 . \nu Y_2 . \quad (11)$$
$$LJ[Z, X_3, Y_2] \left(\begin{array}{c} B_3 \wedge \langle \alpha, Z, X_3 \rangle \\ \vee \\ B_2 \wedge \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \end{array} \right),$$

where for all $T \subseteq B_3 \cup B_2$, we have:

$$LJ[Z, X_3, Y_2](T) = \mu X_1 . \nu Y_0 .$$
$$\left(\begin{array}{c} T \\ \vee \\ B_1 \wedge \left(\begin{array}{l} \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \\ \odot \langle \alpha, Y_2, X_1 \rangle \end{array} \right) \\ \vee \\ B_0 \wedge \left(\begin{array}{l} \langle \alpha, Z, X_3 \rangle \odot \langle \beta, Y_2, X_3 \rangle \\ \odot \langle \alpha, Y_2, X_1 \rangle \odot \langle \beta, Y_0, X_1 \rangle \end{array} \right) \end{array} \right).$$

Note that, for all $Z, X_3, Y_2 \subseteq S$ and $T \subseteq B_3 \cup B_2$, we have $LJ[Z, X_3, Y_2](T) = T \cup Q$ for some $Q \subseteq B_0 \cup B_1$: i.e., function $LJ[\cdot](T)$ adds to T a subset of $B_0 \cup B_1$.

We give an informal explanation of the algorithm, with the aid of the example game depicted in Figure 3. Similarly to co-Büchi games, the fixpoint Z^* of (11) can be written as an increasing sequence of sets $X_3^{(1)} \subset X_3^{(2)} \subset \cdots \subset X_3^{(k)}$, where for $1 \leq i < k$, the set $X_3^{(i+1)}$ is obtained from $X_3^{(i)}$ in one of the two following ways:

A1. either $X_3^{(i+1)} = X_3^{(i)} \cup \{s_i\}$, where $s_i \in B_3$ and $s_i \in \langle \alpha, Z^*, X_3^{(i)} \rangle$;

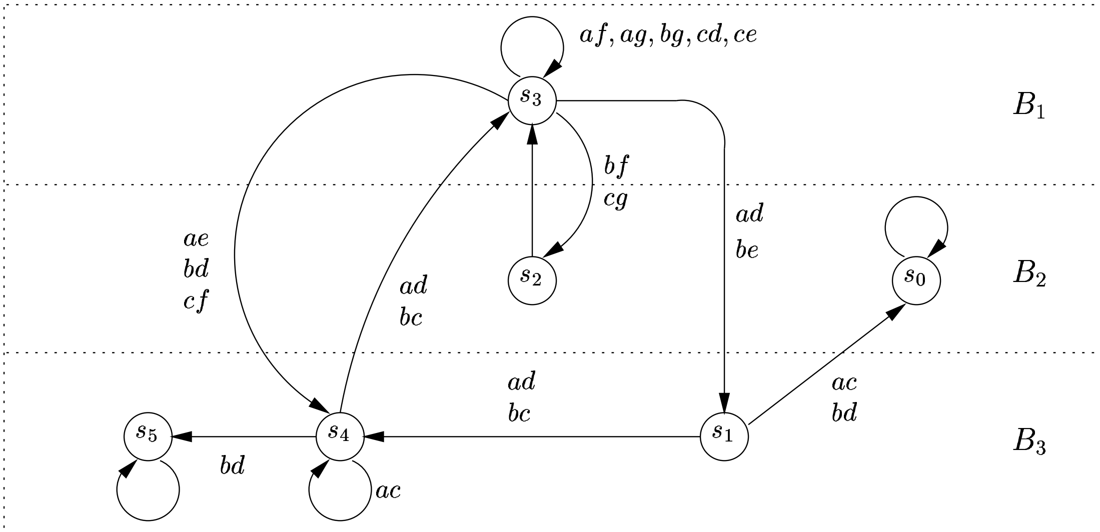

Figure 3: A Rabin-chain game with two-pair accepting condition. The colors are B_0, B_1, B_2, B_3, with $B_0 = \emptyset$, $B_1 = \{s_3\}$, $B_2 = \{s_0, s_2\}$, and $B_3 = \{s_1, s_4, s_5\}$. The moves are $\Gamma_1(s_1) = \{a, b\}$, $\Gamma_2(s_1) = \{c, d\}$, $\Gamma_1(s_3) = \{a, b, c\}$, $\Gamma_2(s_3) = \{d, e, f, g\}$, $\Gamma_1(s_4) = \{a, b\}$, $\Gamma_2(s_4) = \{c, d\}$; at states s_0, s_2, and s_5 the two players have only one move.

A2. or $X_3^{(i+1)} = X_3^{(i)} \cup Y_2^{(i)}$, where $Y_2^{(i)}$ is a fixpoint of (11) that satisfies the following conditions:

$$Y_2^{(i)} \cap B_2 = \langle \alpha, Z^*, X_3^{(i)} \rangle \odot \langle \beta, Y_2^{(i)}, X_3^{(i)} \rangle \qquad (12)$$

$$Y_2^{(i)} = LJ[Z^*, X_3^{(i)}, Y_2^{(i)}](X_3^{(i)} \cup (Y_2^{(i)} \cap B_2)). (13)$$

Note that $Y_2^{(i)} \subseteq B_0 \cup B_1 \cup B_2$.

In the game of Figure 3, we have $\langle\langle 1 \rangle\rangle_{limit}\, \rho = Z^* = \{s_0, s_1, s_2, s_3, s_4\}$, and the set Z^* is computed in $k = 4$ steps, with $X_3^{(1)} = \{s_0\}$, $X_3^{(2)} = \{s_0, s_1\}$, $X_3^{(3)} = \{s_0, s_1, s_2, s_3\}$, $X_3^{(4)} = \{s_0, s_1, s_2, s_3, s_4\}$. The states s_1 and s_4 are added according to condition A1 above:

- at state s_1, if player 1 plays moves a and b with equal probability, he can reach state s_0 with probability 1/2, and leave Z^* with probability 0;
- at state s_4, if player 1 plays move a with probability $1 - \varepsilon$ and move b with probability ε, for $\varepsilon > 0$, he can reach s_3 with probability at least $1 - \varepsilon$, and leave Z^* with probability at most ε.

State s_0 is added trivially according to condition A2. The nontrivial case is for $Y_2^{(2)} = \{s_2, s_3\}$, added to $X_3^{(2)} = \{s_0, s_1\}$ according to condition A2. This case is explained below.

For $i \in \{1, \ldots, k-1\}$, assume that $Y_2^{(i)}$ has been added according to condition A2. At $Y_2^{(i)} \cap B_2$, as in the case of co-Büchi games, the game proceeds to $X_3^{(i)}$ with probability arbitrarily larger than that of leaving Z^*, and equal to at least a fraction of the probability of leaving $Y_2^{(i)} \cup X^{(i)}$. At the states $s \in Y_2^{(i)} \cap (B_0 \cup B_1)$, the predecessor operators correspond to the nesting of two games: the top co-Büchi game with colors B_3 and B_2 and objective $\Diamond\Box B_2$, and a bottom pseudo-co-Büchi game with colors B_0 and B_1, and objective $\Diamond(X_3^{(i)} \vee (Y_2^{(i)} \wedge B_2) \vee \Box B_0)$. The second game corresponds to expression (13). We can show that, unless player 2 plays so that player 1 can win the top game, player 1 can win the bottom game with probability arbitrarily close to 1.

To make this notion more precise, given $s \in B_1$ (resp. $s \in B_0$), let θ_s be the sequence of 3 (resp. 4) triples corresponding to s in (10). If $s \in \theta_s$ then, given the values for α_{X_3} and α_{X_1}, we can construct a suitable distribution $\xi_1 \in \mathcal{D}(\Gamma_1(s))$ and select suitable values for β_{X_1} and β_{X_3} such that $\mathcal{P}(\theta_s)$ is satisfied for all $\xi_2 \in \mathcal{D}(\Gamma_2(s))$. Given $\xi_2 \in \mathcal{D}(\Gamma_2(s))$, we say that ξ_2 *satisfies the top game* if $P_s^{\xi_1, \xi_2}(X_3) > 0$, and we say that ξ_2 *satisfies the bottom game* otherwise.

Corresponding to this hierarchy of games, if the game enters $Y_2^{(i)} \cap (B_0 \cup B_1)$ there are two cases:

B1. If player 2 chooses a distribution that satisfies the top game, then the game proceeds to $X_3^{(i)}$ with a *positive* probability that is arbitrarily larger than that of leaving Z^*, and equal to at least a fraction of the probability of leaving $Y_2^{(i)} \cup X^{(i)}$.

B2. If player 2 chooses only distributions that satisfy the bottom game, then we can rewrite the expression for $Y_2^{(i)}$ as follows:

$$Y_2^{(i)} = \mu X_1.\nu Y_0.$$

$$\left(\begin{array}{ll} & X_3^{(i)} \vee (B_2 \wedge Y_2^{(i)}) \\ \vee & (B_1 \wedge \langle \alpha, Y_2^{(i)}, X_1 \rangle) \\ \vee & (B_0 \wedge \langle \alpha, Y_2^{(i)}, X_1 \rangle \odot \langle \beta, Y_0, X_1 \rangle) \end{array} \right).$$

By analogy with expression (4) for limit co-Büchi games, in this case we conclude that with probability arbitrarily close to 1, we have that either player 1 wins the game while staying in $Y_2^{(i)} \cap (B_0 \cup B_1)$, or the game proceeds to $X_3^{(i)} \cup (B_2 \cap Y_2^{(i)})$.

The precise analysis is more complex, since player 2 can play a strategy that causes a mix of the above cases. Nevertheless, we can prove that with probability arbitrarily close to 1 the paths that enter $Y_2^{(i)}$ behave as follows:

- either they win in $Y_2^{(i)}$,
- or they leave $Y_2^{(i)}$, proceeding to $X^{(i)}$ with probability arbitrarily larger than that of leaving Z^*, and equal to at least a fraction of the probability of leaving $Y_2^{(i)} \cup X^{(i)}$.

Reasoning as in the case of co-Büchi games, this in turns enables us to prove by induction on i, from $i = k - 1$ down to $i = 0$, that

$$Z^* \setminus X_3^{(i)} \subseteq \langle\langle 1 \rangle\rangle_{limit} ((\rho \wedge \Box(Z^* \setminus X_3^{(i)})) \vee \Diamond X_3^{(i)}),$$

which shows $Z^* \subseteq \langle\langle 1 \rangle\rangle_{limit}\, \rho$. The argument for the reverse inclusion verges on the duality of the α and β conditions, which in turn is proved on the basis of the algorithms for the computation of the predecessor operators.

In the game of Figure 3, the set $Y_2^{(2)} = \{s_2, s_3\}$ is added to the $X_3^{(2)} = \{s_0, s_1\}$. If we evaluate (13) in the same iterative fashion as (11), then when s_3 is added we have $X_1^{(1)} = \{s_2, s_3, s_0\}$. Given α_{X_3}, player 1 chooses the distribution that plays moves a and b with probability $\alpha_{X_3}/(1 + 2\alpha_{X_3})$, and move c with probability $1/(1 + 2\alpha_{X_3})$ each. There are two cases:

- If player 2 plays moves d or e, then the game from $Y_2^{(2)}$ goes to s_1 and s_3 with probability 1/2 each. This corresponds to winning the top game. Indeed, by choosing the distribution for player 1 at state s_4 according to a sequence $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$ such that $\prod_{j=0}^{\infty}(1 - \varepsilon_j)$ is arbitrarily close to 1, player 1 can proceed to s_0 (and win the two-pair Rabin-chain game) with probability arbitrarily close to 1.
- If player 2 plays moves f or g, then from s_3 the game goes to s_2 with probability α_{X_3} times greater than that of going to s_4. This corresponds to winning the bottom game. Indeed, by choosing α_{X_3} after each visit to s_2 according to a sequence $\alpha_{X_3} = \alpha_0, \alpha_1, \alpha_2, \ldots$ such that $\prod_{j=0}^{\infty}(1 - \alpha_j)$ is arbitrarily close to 1, player 1 can stay in $\{s_2, s_3\}$ and visit s_2 infinitely often with probability arbitrarily close to 1.

The analysis of strategies of player 2 that mix the above behaviors is technically more difficult, but confirms that player 1 can win the Rabin-chain game with probability arbitrarily close to 1.

5.2 General Rabin-chain games

Consider a Rabin chain games with $m > 0$ pairs and accepting condition defined as in (5). Define the predecessor operators:

$$\text{ARpre}^m_{2m-1,1} = \langle \gamma, Y_{2m}, X_{2m-1} \rangle$$
$$\text{ARpre}^m_{2m-2,1} = \langle \gamma, Y_{2m}, X_{2m-1} \rangle \odot \langle \beta, Y_{2m-2}, X_{2m-1} \rangle$$

$$\text{LRpre}^m_{2m-1,1} = \langle \alpha, Y_{2m}, X_{2m-1} \rangle$$
$$\text{LRpre}^m_{2m-2,1} = \langle \alpha, Y_{2m}, X_{2m-1} \rangle \odot \langle \beta, Y_{2m-2}, X_{2m-1} \rangle$$

and, for $i \in \{0, \ldots, m-2\}$, by:

$$\lambda\text{Rpre}^m_{2i+1,1} = \lambda\text{Rpre}^m_{2i+2,1} \odot \langle \alpha, Y_{2i+2}, X_{2i+1} \rangle$$
$$\lambda\text{Rpre}^m_{2i,1} = \lambda\text{Rpre}^m_{2i+1,1} \odot \langle \beta, Y_{2i}, X_{2i+1} \rangle.$$

For a winning mode $\lambda \in \{\textit{almost}, \textit{limit}\}$, let λRpre be ARpre if $\lambda = \textit{almost}$, and LRpre if $\lambda = \textit{limit}$.

The set of λ-winning states can be computed by defining the operator λJ by $\lambda J^m_{-1}(T) = T$ and, for $i \in \{0, \ldots, m-1\}$, by

$$\lambda J^m_i(T) = \mu X_{2i+1} \,.\, \nu Y_{2i} \,.$$
$$\lambda J^m_{i-1} \left(\begin{array}{lll} T & \vee & (B_{2i+1} \wedge \lambda\text{Rpre}^m_{2i+1,1}) \\ & \vee & (B_{2i} \wedge \lambda\text{Rpre}^m_{2i,1}) \end{array} \right).$$

The set of winning states is then given by

$$\langle\langle 1 \rangle\rangle_\lambda\, \rho = \nu Y_{2m} \,.\, \lambda J^m_{2m-1}(\emptyset). \tag{14}$$

The following theorem summarizes the results for Rabin-chain games.

Theorem 7 *The following assertions hold.*

1. *For every Rabin-chain condition ρ and $\lambda \in \{\textit{sure}, \textit{almost}, \textit{limit}\}$, we have $\langle\langle 1 \rangle\rangle_\lambda\, \rho = \neg \langle\langle 2 \rangle\rangle_{\bar{\lambda}}\, \neg\rho$.*
2. *The complexity of computing the sets of winning states, and the most restrictive classes in which winning and spoiling strategies are guaranteed to exist are given in Table 1(e).*

Unlike in the previous classes of games, winning a Rabin-chain game with mode *almost* may require the use of infinite-memory strategies. To see this, consider a modification of the game SKIRMISH of Figure 1, in which from state s_{wet} we deterministically proceed to state s_{hide}. We consider 4 colors (hence $m = 2$), with $B_0 = \emptyset$, $B_1 = \{s_{hide}\}$, $B_2 = \{s_{home}\}$, and $B_3 = \{s_{wet}\}$. It is easy to see that if player 1 plays with a memoryless strategy, he wins with probability 0: in fact, if he plays only move *hide*, player 2 can reply with move *wait*; if he plays move *run* with positive probability, player 2 can counter by playing move *throw* deterministically. The argument for finite-memory strategies is similar. On the other hand, player 1 can win with probability 1 by playing move *run* with probability $1/2^{k+1}$ and move *hide* with probability $1-1/2^{k+1}$, where k is the number of prior visits to $\{s_{wet}, s_{home}\}$.

Rabin-chain games also exhibit the following global duality between the limit and almost winning modes of the two players.

Theorem 8 *Let ρ be a Rabin-chain condition. If $\langle\!\langle 2\rangle\!\rangle_{limit}\, \neg\rho = \emptyset$, then $\langle\!\langle 1\rangle\!\rangle_{almost}\, \rho = S$, where as usual S is the set of all states.*

Combining this result with the Duality Theorem (Theorem 2), we obtain the following corollary.

Corollary 1 *For a Rabin-chain condition ρ, we have that $\langle\!\langle 1\rangle\!\rangle_{positive}\, \rho \neq \emptyset$ iff $\langle\!\langle 1\rangle\!\rangle_{limit}\, \rho \neq \emptyset$.*

Intuitively, this corollary states that there is some state from which a player can win a Rabin-chain game with positive probability only if there is some state from which it can win with probability arbitrarily close to 1. This generalizes to two-player concurrent Rabin-chain games a property that holds for Markov chains, where it is an immediate consequence of the decomposition in closed recurrent classes [KSK66].

6 Appendix: Computation of Predecessor Operators

Finally, we present algorithms to decide whether a state s belongs to the sets computed by the predecessor operators ARpre and LRpre. These algorithms rely on μ-calculus expressions that are evaluated over the set $\Gamma_1(s)$ of moves of player 1 at s (rather than on μ-calculus expression evaluated over the set of states of the game). Given a state s, for all subsets of states $X, Y \subseteq S$ we define two functions $A^s_Y : 2^{\Gamma_2(s)} \mapsto 2^{\Gamma_1(s)}$ and $B^s_X : 2^{\Gamma_1(s)} \mapsto 2^{\Gamma_2(s)}$ by

$$A^s_Y(V_2) = \{a \in \Gamma_1(s) \mid \forall b \in \Gamma_2(s) \,.\, \delta(s,a,b) \not\subseteq Y \rightarrow b \in V_2\}$$

$$B^s_X(V_1) = \{b \in \Gamma_2(s) \mid \exists a \in V_1 \,.\, \delta(s,a,b) \cap X \neq \emptyset\},$$

where $V_1 \subseteq \Gamma_1(s)$ and $V_2 \subseteq \Gamma_2(s)$. Given a sequence of triples θ, we inductively define the *μ-prefix* $\mathcal{M}(\theta)$ and the *condition* $\mathcal{C}(\theta)$ of θ as follows, for $X, Y \subseteq S$:

$$\begin{aligned}
\mathcal{M}(\emptyset) &= \emptyset \\
\mathcal{M}(\theta \odot \langle \alpha, Y, X\rangle) &= \mu W_X \,.\, \mathcal{M}(\theta) \\
\mathcal{M}(\theta \odot \langle \beta, Y, X\rangle) &= \nu V_X \,.\, \mathcal{M}(\theta) \\
\mathcal{M}(\theta \odot \langle \gamma, Y, X\rangle) &= \mathcal{M}(\theta)
\end{aligned}$$

$$\begin{aligned}
\mathcal{C}(\langle \alpha, Y, X\rangle) &= A^s_Y(B^s_X(W_X)) \\
\mathcal{C}(\langle \beta, Y, X\rangle) &= A^s_Y(B^s_X(V_X)) \\
\mathcal{C}(\langle \gamma, Y, X\rangle) &= A^s_Y(\emptyset) \\
\mathcal{C}(\theta \odot \langle \alpha, Y, X\rangle) &= \mathcal{C}(\theta) \vee A^s_Y(B^s_X(W_X)) \\
\mathcal{C}(\theta \odot \langle \beta, Y, X\rangle) &= \mathcal{C}(\theta) \wedge A^s_Y(B^s_X(V_X)).
\end{aligned}$$

We can decide whether a state s satisfies a predecessor predicate as follows:

$$s \in (\theta \odot \langle \alpha, Y, X\rangle) \text{ iff } B^s_X\big(\mathcal{M}(\theta \odot \langle \alpha, Y, X\rangle) \,.\, \mathcal{C}(\theta \odot \langle \alpha, Y, X\rangle)\big) = \Gamma_2(s)$$

$$s \in (\theta \odot \langle \beta, Y, X\rangle) \text{ iff } \mathcal{M}(\theta \odot \langle \beta, Y, X\rangle) \,.\, \mathcal{C}(\theta \odot \langle \beta, Y, X\rangle) \neq \emptyset$$

$$s \in \langle \gamma, Y, X\rangle \text{ iff } B^s_X\big(\mathcal{C}(\langle \gamma, Y, X\rangle)\big) = \Gamma_2(s).$$

In an m-pair Rabin-chain game, these algorithms rely on the assumption that $Y_{2i} \subseteq Y_{2k}$ for all $0 \leq i < k \leq m$, and that $X_{2k+1} \subseteq X_{2i+1}$ for all $0 \leq i < k < m$. This assumption, introduced to simplify the required notation, holds when the μ-calculus formula (14) is evaluated in the usual iterative fashion. Using (6), (7), (8), and (9), the above algorithms also enable the computation of the predecessor operators Apre_1, Lpre_1, AFpre_1, and LFpre_1: in particular, for all $s \in S$ and $X, Y, Z \subseteq S$ we have:

$$s \in \text{Apre}_1(Y, X) \text{ iff } B^s_X(A^s_Y(\emptyset)) = \Gamma_2(s)$$

$$s \in \text{Lpre}_1(Y, X) \text{ iff } B^s_X(\mu W \,.\, A^s_Y(B^s_X(W))) = \Gamma_2(s)$$

$$s \in \text{AFpre}_1(Z, Y, X) \text{ iff } \nu V \,.\, \big(A^s_Z(\emptyset) \wedge A^s_Y(B^s_X(V))\big) \neq \emptyset$$

$$s \in \text{LFpre}_1(Z, Y, X) \text{ iff } \nu V \,.\, \mu W \,.\, \big(A^s_Z(B^s_X(W)) \wedge A^s_Y(B^s_X(V))\big) \neq \emptyset.$$

Due to space limitations, we omit the correctness proofs of the algorithms.

References

[AHK97] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proc. 38th IEEE Symp. Found. of Comp. Sci.*, pages 100–109. IEEE Computer Society Press, 1997.

[ALW89] M. Abadi, L. Lamport, and P. Wolper. Realizable and unrealizable concurrent program specifications. In *Proc. 16th Int. Colloq. Aut. Lang. Prog.*, volume 372 of *Lect. Notes in Comp. Sci.*, pages 1–17. Springer-Verlag, 1989.

[BCM+90] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proc. 5th IEEE Symp. Logic in Comp. Sci.*, pages 428–439. IEEE Computer Society Press, 1990.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Found. of Software Tech. and Theor. Comp. Sci.*, volume 1026 of *Lect. Notes in Comp. Sci.*, pages 499–513. Springer-Verlag, 1995.

[BL69] J.R. Büchi and L.H. Landweber. Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.*, 138:295–311, 1969.

[BLV96] N. Buhrke, H. Lescow, and J. Vöge. Strategy construction in infinite games with strett and rabin chain winning conditions. In *TACAS 96*, volume 1055 of *Lect. Notes in Comp. Sci.*, pages 207–225. Springer, 1996.

[CY90] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *Proc. 17th Int. Colloq. Aut. Lang. Prog.*, volume 443 of *Lect. Notes in Comp. Sci.*, pages 336–349. Springer-Verlag, 1990.

[dAHK98] L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. In *Proc. 39th IEEE Symp. Found. of Comp. Sci.*, pages 564–575. IEEE Computer Society Press, 1998.

[Dil89] D.L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-independent Circuits*. The MIT Press, 1989.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proc. 32nd IEEE Symp. Found. of Comp. Sci.*, pages 368–377. IEEE Computer Society Press, 1991.

[GH82] Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proc. 14th ACM Symp. Theory of Comp.*, pages 60–65. ACM Press, 1982.

[KPBV95] S. Krishnan, A. Puri, R. Brayton, and P. Varaiya. Rabin index, chain automata and applications to automata and games. In *CAV 95: Computer Aided Verification*, volume 939 of *Lect. Notes in Comp. Sci.*, pages 253–266. Springer-Verlag, 1995.

[KS81] P.R. Kumar and T.H. Shiau. Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games. *SIAM J. Control and Optimization*, 19(5):617–634, 1981.

[KSK66] J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains*. D. Van Nostrand Company, 1966.

[KV96] O. Kupferman and M.Y. Vardi. Module checking. In *Proc. 8th Conference on Computer Aided Verification*, volume 1102 of *Lect. Notes in Comp. Sci.*, pages 75–86. Springer-Verlag, 1996.

[LW95] H. Lescow and T. Wilke. On polynomial-size programs winning finite-state games. In *CAV 95: Computer Aided Verification*, volume 939 of *Lect. Notes in Comp. Sci.*, pages 239–252. Springer-Verlag, 1995.

[McN93] R. McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl. Logic*, 65:149–184, 1993.

[Mos84] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *Computation Theory*, volume 208 of *Lect. Notes in Comp. Sci.*, pages 157–168. Springer-Verlag, 1984.

[PR89] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th Annual Symposium on Principles of Programming Languages*, pages 179–190. ACM Press, 1989.

[Sha53] L.S. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. USA*, 39:1095–1100, 1953.

[Tho90] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 4, pages 135–191. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of 12th Annual Symp. on Theor. Asp. of Comp. Sci.*, volume 900 of *Lect. Notes in Comp. Sci.*, pages 1–13. Springer-Verlag, 1995.

[TW68] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical System Theory*, 2:57–81, 1968.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state systems. In *Proc. 26th IEEE Symp. Found. of Comp. Sci.*, pages 327–338. IEEE Computer Society Press, 1985.