Concurrent Games and Semi-Random Determinacy

Stéphane Le Roux
Darmstadt Technical University, Department of Mathematics, Darmstadt, Germany
leroux@mathematik.tu-darmstadt.de

Abstract

Consider concurrent, infinite duration, two-player win/lose games played on graphs. If the winning condition satisfies some simple requirement, the existence of Player 1 winning (finite-memory) strategies is equivalent to the existence of winning (finite-memory) strategies in finitely many derived one-player games. Several classical winning conditions satisfy this simple requirement.

Under an additional requirement on the winning condition, the non-existence of Player 1 winning strategies from all vertices is equivalent to the existence of Player 2 stochastic strategies almost-sure winning from all vertices. Only few classical winning conditions satisfy this additional requirement, but a fairness variant of omega-regular languages does.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory, Theory of computation → Verification by model checking, Software and its engineering → Software verification, Software and its engineering → Model checking

Keywords and phrases Two-player win/lose, graph, infinite duration, abstract winning condition

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.40

Acknowledgements Referees (of several conferences) and Sasha Rubin made helpful comments. A simplification of the proof of Lemma 11 was triggered by a conversation with Arno Pauly.

1 Introduction

Computer science models systems interacting concurrently with their environment via infinite duration two-player win/lose games played on graphs: a play starts at a *state* of the graph, where the players *concurrently* choose one *action* each and thus induce the next state, and so on for infinitely many rounds. The *winning condition* is a given subset W of the infinite sequences of states, and Player 1 wins the play iff the sequence of visited states belongs to W. A *strategy* of a player prescribes one action depending on what has been played so far, and a *winning strategy* is a strategy ensuring victory regardless of the opponent strategy.

There are games where neither of the players has a winning strategy, but Borel determinacy [25] guarantees the existence of a winning strategy in games where the players play alternately and the winning condition is a Borel set. Under Borel condition again, Blackwell determinacy [26] guarantees a weaker conclusion when the players play concurrently: there exists a value $v \in [0,1]$ such that for all $\epsilon > 0$ the players have stochastic strategies guaranteeing victory with probability $v - \epsilon$ and $1 - v - \epsilon$, respectively.

In the special case of concurrent games played on finite graphs with ω-regular winning conditions, [11] designed algorithms to decide the existence of (stochastic) strategies that are winning, winning with probability one, and winning with probability $1 - \epsilon$ for all $\epsilon > 0$. [11] also mentions a three-state game where only the latter exist, which exemplifies the complexity of the concurrent ω-regular games on finite graphs. Then [6] studied concurrent

q_0		b_1	b_2
	a_1	$0, q_0$	$0, q_1$
	a_2	$1, q_0$	$0, q_0$

q_1		b_1	b_2
	a_1	$2, q_0$	$1, q_0$
	a_2	$2, q_1$	$2, q_1$

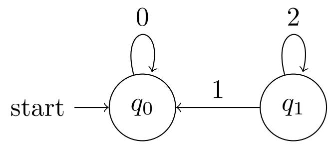

Figure 1 To the left, a concurrent game with states q_0, q_1, colors $0, 1, 2$, and two actions per player. To the right, a one-player game derived by using the delayed response $[(0, q_0)(0, q_0)]; [(1, q_0)(2, q_1)]$.

prefix independent winning conditions, which is strictly more general than the ω-regular conditions, and [13] further improved upon some results. Some of these results were extended recently to multi-player multi-outcome games, see e.g. [3], [15].

The new games. This article studies slightly different games: when the players concurrently choose one action each, it also produces a *color*; the winning condition is now a given subset W of the infinite sequences of colors; and Player 1 wins the play iff the produced sequence of colors belongs to W. There are two differences between the classical games and the new games. First, the winning condition does not involve the visited states but the transitions instead; second it does so indirectly, via colors labeling the transitions. E.g. in the game on the left-hand side of Figure 1, starting at q_0, the action sequence $(a_1, b_1)(a_1, b_2)(a_1, b_1)$ yields the state sequence $q_0q_0q_1q_0$ and the color sequence 002.

There are several reasons why these new games are interesting.

- The classical games can be encoded easily into the new ones by using state names as colors. Variants such as the games with colored states, or the colorless games with winning condition on the transitions can also be encoded easily into the new games.
- The converse encoding may increase the state space (to infinity for games with infinitely many actions). Note that the transition-versus-state issue was already studied in the turned-based setting in [10]. Likewise, colorless games are encoded easily in games with colors without size increase, and colors usually lead to more succinct winning conditions.
- Colors are widely used in turn-based games, and for all games they help to study the winning conditions independently from the game structure, and thus to approximate or even characterize nice winning conditions for classes of games (usually simple to check) rather than for single games (usually more accurate but harder to check). This is exemplified by the difference between Theorems 5 and 7 in [27].
- Whereas classical one-state games are trivial, the new one-state games are fairly complex and constitute a nice intermediate object towards the understanding of the more complex general games. Likewise, some one-state (aka stateless) objects from the literature are interesting in their own right: [1] studied one-state multi-objective Markov decision processes; vector addition systems (VAS, [17]) are still studied despite the vector addition systems with states (VASS, [16]); the Minkowski games [24] defined with finite sets are a special case of the one-state games from this article.

The main results.

- If W is closed under *interleaving* and *prefix removal*, and if states and colors are finitely many, the existence of a Player 1 winning (finite-memory) strategy is equivalent to the existence of winning (finite-memory) strategies in finitely many derived one-player games.

- If, in addition, W is *factor-prefix complete* and there are finitely many actions, either Player 1 has a winning strategy from one state, or every Player 2 constant (stronger than positional!), positive, stochastic strategy is almost-sure winning from all vertices. This is *semi-random determinacy*.
- One-state games enjoy a stronger conclusion than in the previous item under somewhat weaker assumptions: if the winning condition is *factor-set complete* and closed under interleaving, if Player 2 has finitely many actions, either Player 1 has a winning strategy, or every Player 2 constant, positive stochastic strategy is almost-sure winning.

The finitary flavor of the above characterizations yields decidability and memory sufficiency, in the rough range of double exponentials in the number of states times the number of colors.

In the context of semi-random determinacy, a neutral, random Player 2 is therefore as bad for Player 1 as a hostile environment. Also, the victory is clear-cut in the above results: no need for approximate optimal strategies, no need for the notion of value, etc. This is due to the assumptions, and it is legitimate to wonder how restrictive they are.

Several classical winning conditions from computer science are closed under interleaving, see Section 5. The Muller condition is not, but the parity condition is, so the first characterization result extends to the concurrent Muller games via the Last Appearance Record (LAR), as done in [28]. So, closedness under interleaving is not as restrictive as it may seem.

Fewer classical winning conditions are factor-prefix complete (defined in Section 3.2), but the boundedness condition from [24] and a variant of the ω-regular languages are both closed under interleaving and factor-prefix complete. The variant is as follows: each produced color requests some combinations of colors to occur in the future. In winning plays, the number of currently unsatisfied requests should be uniformly bounded over time. It may be relevant even as a business model: at every time unit the system can pay penalties for every currently unsatisfied request, which may be covered by greater, albeit bounded, instantaneous income.

The above variant relates to the notion of *fairness*, which requires that co-finitely many requests are eventually satisfied. The *finitary fairness* [2] additionally requires uniformly bounded response time. This idea was used in [12] to study temporal logic, and in [9] to study finitary parity games. Requiring uniformly bounded response time (or variants thereof) to study games has been further used later, e.g. in [5]. However, these notions of fairness do not enjoy closedness under interleaving and factor-prefix completeness. (Details in Section 5.)

Related works. The semi-random determinacy implies the *bounded limit-one property* from [11] for the new games: if one state has positive value, one state has value one.

Corollary 4 generalizes the nice Theorem 4 from [18]. Note that the convexity of winning conditions defined in [18] is a essentially the same as the interleaving closedness defined here.

This article also shares similarities with [14]: both use abstract winning conditions, and both characterize the existence of winning strategies in two-player games by the existence of winning strategies in finitely many derived one-player games. Several articles adopted a similar approach: [19] and [20] reduce multi-player multi-outcome Borel games to simpler two-player win/lose Borel games, and characterize the preferences and structures that guarantee the existence of Nash equilibrium in infinite tree-games; [21] does the same to characterize the preferences that guarantee the existence of subgame perfect equilibrium (at low levels of the Borel hierarchy); [23] and [27] do the same to almost characterize the existence of finite-memory Nash equilibrium in games on finite graphs; [22] reduces one-shot concurrent two-player multi-outcome games to simpler one-shot concurrent two-player win/lose games, with applications to generalized Muller games and generalized "parity" games.

One of the benefits of abstraction is that it leads to more general results: e.g. [23] noted that the lexicographic product of mean-payoff and reachability objectives cannot be encoded into real-valued payoffs, and [27] proved it.

Structure of the article. Section 2 gives basic definitions. Section 3 presents the main results and additional definitions. Section 4 discusses the key elements of the proofs. Section 5 presents applications.

2 Definitions

The folklore Observation 1 below will be used extensively to lift properties from finite words to infinite words. It will be first explicitly invoked, and then only implicitly used.

▶ **Observation 1.** *Let $f : S^* \to T^*$ be such that $u \sqsubseteq v \Rightarrow f(u) \sqsubseteq f(v)$, where $\sqsubseteq$ is the prefix relation. Then f can be uniquely extended to $S^* \cup S^\omega \to T^* \cup T^\omega$ such that $f(\rho_{\leq n}) \sqsubseteq f(\rho)$ for all $n \in \mathbb{N}$ and $\rho \in S^\omega$.*

Games. A *game (with colors and states)* is a tuple $\langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ such that

- A_1 and A_2 are non-empty sets (of actions for Player 1 and Player 2),
- Q is a non-empty set (of states),
- $q_0 \in Q$ (is the initial state),
- $\delta : Q \times A_1 \times A_2 \to Q$ (is the state update function).
- C is a non-empty set (of colors),
- $\mathrm{col} : Q \times A_1 \times A_2 \to C$ (is a color trace),
- $W \subseteq C^\omega$ (is the winning condition for Player 1)

Histories. The *full histories* (*full runs*) of such a game are the finite (infinite) words over $A_1 \times A_2$, the Player 2 *histories* (Player 2 *runs*) are the finite (infinite) words over A_2, and the Player 1 *histories* (Player 1 *runs*) are the finite (infinite) words over A_1.

Strategies. A Player 1 strategy is a function from A_2^* to A_1. Informally, it requires Player 1 to remember exactly how Player 2 has played so far, and it tells Player 1 how to play.

Induced histories. The function h is defined inductively below. As arguments it expects a strategy and a Player 2 history in A_2^*, and it returns a full history: the very full history that, morally, should happen if Player 1 followed the given strategy while Player 2 played the given Player 2 history. Namely, $h(s, \epsilon) := \epsilon$ and $h(s, \beta \cdot b) := h(s, \beta) \cdot (s(\beta), b)$.

By Observation 1 the function h is extended to expect opponents runs in A_2^ω and return full runs: $h(s, \boldsymbol{\beta})$ is the only action run whose prefixes are the $h(s, \boldsymbol{\beta}_{\leq n})$ for $n \in \mathbb{N}$.

Extending the update and trace functions. The state update function δ is extended to $\Delta : (A_1 \times A_2)^* \to Q$ inductively: $\Delta(\epsilon) := q_0$ and $\Delta(\rho \cdot (a, b)) := \delta(\Delta(\rho), a, b)$. Using Δ, the trace function col is naturally lifted to full histories by induction: $\mathrm{col}(\epsilon) := \epsilon$ and $\mathrm{col}(\rho \cdot (a, b)) := \mathrm{col}(\rho) \cdot \mathrm{col}(\Delta(\rho), a, b)$. The trace function is further extended to full runs by Observation 1. When considering several games, indices may be added to the corresponding Δ and col.

Winning strategies. A Player 1 strategy s is winning if $\mathrm{col} \circ h(s, \boldsymbol{\beta}) \in W$ for all $\boldsymbol{\beta} \in A_2^\omega$. If there is a Player 1 winning strategy in a game, one says that Player 1 wins the game.

Memory. A Player 1 strategy s is said be implementable with memory M, or memory size $\log_2|M|$, if there exist a set M and $m_0 \in M$, and two functions $\sigma : Q \times M \to A_1$ and $\mu : Q \times M \times A_2 \to M$ such that $s(\beta) = \sigma(\Delta \circ h(s,\beta), m(\beta))$, where m is defined inductively by $m(\epsilon) := m_0$ and $m(\beta b) := \mu(\Delta \circ h(s,\beta), m(\beta), b)$. If M is finite, s is called a finite-memory strategy. Note that every Player 1 strategy is implementable with memory A_2^ω.

One-player games. Intuitively, a *one-player game (with colors and states)* amounts to a game where Player 2 has only one strategy available, i.e. $|A_2| = 1$. Formally, it is a tuple $\langle A_1, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ such that A_1, Q, and C are non-empty sets, $q_0 \in Q$, $\delta : Q \times A_1 \to Q$, $\mathrm{col} : Q \times A_1 \to C$, and $W \subseteq C^\omega$. In this context, the *full histories* (*full runs*) of such a game are the finite (infinite) words over A_1, and the Player 2 *histories* of Player 1 are the natural numbers (telling how many rounds have been played). There is only one Player 2 *run*, namely ω. Then, a Player 1 strategy is a function from $\mathbb{N}$ to A_1, and the notation for the induced full histories is overloaded: $h(s, 0) := \epsilon$ and $h(s, n + 1) := h(s, n) \cdot s(n)$. By Observation 1 the function h is (again) extended: $h(s,\omega)$ is the only action run whose prefixes are the $h(s, n)$ for $n \in \mathbb{N}$. A Player 1 strategy s is winning if $\mathrm{col} \circ h(s,\omega) \in W$.

Prefix removal. A set of infinite sequences is closed under prefix removal if the tails of the sequences from the set are again in the set. Formally, $W \subseteq C^\omega$ is closed under prefix removal if the following holds: $\forall(\gamma, \boldsymbol{\gamma}) \in C^* \times C^\omega$, $\gamma \cdot \boldsymbol{\gamma} \in W \Rightarrow \boldsymbol{\gamma} \in W$. Note that closedness under prefix removal is weaker than the prefix independence assumed in [6], [13], and [18].

Interleaving. Interleaving two infinite sequences consists in enumerating sequentially (prefixes of) the two sequences to produce a new infinite sequence. For example, interleaving $(2n)_{n\in\mathbb{N}}$ and $(2n + 1)_{n\in\mathbb{N}}$ can produce the sequences $(n)_{n\in\mathbb{N}}$ (perfect alternation), $1 \cdot 0 \cdot 3 \cdot 5 \cdot 2 \cdot 7 \cdot 4 \cdot 6 \cdot (n + 8)_{n\in\mathbb{N}}$, and $(2n)_{n\in\mathbb{N}}$ (by enumerating the first sequence only), but not the sequences $(4n)_{n\in\mathbb{N}}$ or $0 \cdot 1 \cdot 4 \cdot 3 \dots$.

Delayed response. Consider a game $g = \langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ with finite Q and C. For every $q \in Q$ let $E_1^q, \dots, E_{k_q}^q$ be the elements of $\{(\mathrm{col}, \delta)(q, a, A_2) \mid a \in A_1\}$, where $(\mathrm{col}, \delta)(q, a, A_2) := \{(\mathrm{col}(q, a, b), \delta(q, a, b)) \mid b \in A_2\}$ for all $a \in A_1$. The elements of $\otimes_{q\in Q, i\leq k_q} E_i^q$ are called the Player 2 *delayed responses*. Intuitively, a Player 2 delayed response amounts to a Player 2 positional strategy in (and only in) a sequentialized version of the game. In every round of this version, Player 1 chooses an action first, then Player 2 chooses an action (or more precisely some color and state among the pairs he could induce by choosing an action). E.g. $[(0, q_0)(0, q_0)]; [(1, q_0)(2, q_1)]$ is a delayed response for Figure 1. It means that at state q_0, Player 2 selects $(0, q_0)$ for both actions of Player 1, and at state q_1 it selects $(1, q_0)$ if Player 1 chooses action a_1. Note that delayed responses are not Player 2 (positional) strategies in the concurrent game, e.g. as $[(0, q_0)(0, q_0)]$ is not achievable in any column.

Derived one-player games. Let t be a Player 2 delayed response. The one-player game $g(t) := \langle A_1, Q, q_0, \delta_t, C, \mathrm{col}_t, W\rangle$ is defined by $(\mathrm{col}_t, \delta_t)(q, a) := t_{q,(\mathrm{col},\delta)(q,a,A_2)}$, the projection of t on the (q, E_i^q)-component such that $E_i^q = (\mathrm{col}, \delta)(q, a, A_2)$. Intuitively, $g(t)$ is the game obtained by letting Player 2 fix his strategy (to realize) t in the sequentialized version of g. For example, the game on the left-hand side of Figure 1 applied to the delayed response $[(0, q_0)(0, q_0)]; [(1, q_0)(2, q_1)]$ yields the game on the right-hand side of Figure 1.

3 Main results

Section 3.1 characterizes the existence of Player 1 winning strategies and gives a complexity result. Section 3.2 defines additional concepts and uses the above characterization to characterize the existence of Player 2 everywhere-winning stochastic strategies. Section 3.3 studies the special case of one-state games and presents the semi-random determinacy.

3.1 Existence of Player 1 winning strategies

Theorem 2 below characterizes the existence of Player 1 winning strategies in a game via the existence of winning strategies in finitely many derived one-player games. Theorem 3 afterwards drops the assumption on closedness under prefix removal from Theorem 2, but at the cost of a universal quantification over the starting state of the game. In Theorems 2 and 3, the finiteness and the closedness assumptions are used only to prove the $2 \Rightarrow 1$ implications.

▶ **Theorem 2.** *Consider a game $g = \langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$. If Q and C are finite, and W is closed under interleaving and prefix removal, the following are equivalent.*
1. Player 1 *wins g.*
2. Player 1 *wins $g(t)$ for all delayed responses t.*

If A_1 is finite and Player 1 *wins, she can do it with memory size $O(f(|A_1|, |Q|, |C|) \cdot (|C \times Q|)^{|Q|2^{|C\times Q|}})$, where $f(|A_1|, |Q|, |C|)$ is a sufficient memory size to win the one-player games using A_1, Q and C.*

▶ **Theorem 3.** *Consider games $g_q = \langle A_1, A_2, Q, q, \delta, C, \mathrm{col}, W\rangle$ parametrized by $q \in Q$. If A_1, A_2, and Q are finite, if W is factor-prefix complete and closed under interleaving and prefix removal, the following are equivalent.*
1. Player 1 *wins g_q for all $q \in Q$.*
2. Player 1 *wins $g_q(t)$ for all $q \in Q$ and delayed responses t.*

If the above holds, Player 1 *wins every g_q with memory size as in Theorem 2.*

In games that are (or encode) turn-based games, the delayed responses are Player 2 positional strategies. So, restricting Theorems 2 and 3 to turn-based games yields Corollaries 4 and 5, respectively. Note that Corollary 4 generalizes Theorem 4 from [18] by only assuming closedness under prefix removal instead of prefix independence. This is significant since the safety condition is closed under interleaving and prefix removal, but is not prefix independent.

▶ **Corollary 4.** *Consider a game $g = \langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ encoding a turn-based game. If Q and C are finite, and W is closed under interleaving and prefix removal, either* Player 1 *has a winning strategy or* Player 2 *has a positional winning strategy.*

▶ **Corollary 5.** *Consider games $g_q = \langle A_1, A_2, Q, q, \delta, C, \mathrm{col}, W\rangle$ parametrized by $q \in Q$ and encoding a turn-based games. If Q and C are finite, and W is closed under interleaving, either* Player 1 *wins all g_q, or* Player 2 *has a positional winning strategy for some g_q.*

The characterizations from Theorems 2 and 3 yields decidability results and rough algorithmic complexity estimates in Corollary 6 below. Note that checking all the possible strategies using memory size given by Theorems 2 and 3 would be slower than Corollary 6.

▶ **Corollary 6.** *Let $\mathcal{C} \neq \emptyset$, let $W \subseteq \mathcal{C}^\omega$ be closed under interleaving and prefix removal (resp. by interleaving), and let $f : \mathbb{N}^3 \to \mathbb{N}$ be such that for all finite $C \subseteq \mathcal{C}$ and all one-player games $\langle A_1, Q, q_0, \delta, C, \mathrm{col}, W\rangle$, it takes at most $f(|A_1|, |Q|, |C|)$ computation steps to decide* *the existence of a (finite-memory) winning strategy in the game. Then for all finite games* $g_{q_0} = \langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ *it takes at most*

$$f(|A_1|, |Q|, |C|) \cdot (|C \times Q|)^{|Q|2^{|C\times Q|}} + |Q||A_1||A_2|$$

computation steps to decide whether Player 1 *wins* g_{q_0} *(with finite memory).*

(resp. $|Q| \cdot f(|A_1|, |Q|, |C|) \cdot (|C \times Q|)^{|Q|2^{|C\times Q|}} + |Q||A_1||A_2|$ *computation steps to decide whether* Player 1 *wins* g_q *(with finite memory) for all* $q \in Q$*.)*

3.2 Existence of Player 2 almost-sure winning random strategies

Consider a game $\langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$.

Probability distribution. A probability distribution on a finite set E is a function $f : E \to [0, 1]$ such that $\sum_{e\in E} f(e) = 1$. Let us call $D(E)$ the set of the probability distributions on E.

Stochastic strategies. A Player 1 (Player 2) stochastic strategy is a function $\sigma : (A_1\times A_2)^* \to D(A_1)$ $(\tau : (A_1 \times A_2)^* \to D(A_2))$.

Induced stochastic histories. The function H is defined inductively below. As arguments it expects stochastic strategies σ and τ for Player 1 and a Player 2, respectively, and it returns a function from $(A_1 \times A_2)^*$ to $\mathbb{R}$. Namely, $H(\sigma, \tau)(\epsilon) := 1$, and $H(\sigma, \tau)(\rho \cdot (a, b)) := H(\sigma, \tau)(\rho) \cdot \sigma(\rho)(a) \cdot \tau(\rho)(b)$. It is easy to check that $H(\sigma, \tau)(\rho) \geq 0$ for all $\rho \in (A_1 \times A_2)^*$, and that $\sum_{|\rho|=n} H(\sigma, \tau)(\rho) = 1$ for all $n \in \mathbb{N}$.

Induced probability measure. For every pair $(\sigma, \tau) \in D(A_1)^{(A_1\times A_2)^*} \times D(A_2)^{(A_1\times A_2)^*}$ one defines a probability measure $\lambda(\sigma, \tau)$ on $(A_1 \times A_2)^\omega$ by setting $\lambda(\sigma, \tau)(\rho \cdot (A_1 \times A_2)^\omega) := H(\sigma, \tau)(\rho)$ for all $\rho \in (A_1 \times A_2)^*$. (It is then extended uniquely to measurable sets.)

Almost-sure winning stochastic strategies. A Player 2 stochastic strategy τ is said to be almost-sure winning if $\lambda(\sigma, \tau)(\mathrm{col}^{-1}[W]) = 0$ for all $\sigma \in D(A_1)^{(A_1\times A_2)^*}$. (Recall that $\mathrm{col} : (A_1 \times A_2)^\omega \to C^\omega$ is an extension of $\mathrm{col} : Q \times A_1 \times A_2 \to C$ with notation overload.)

Factor-prefix completeness. Informally, W is factor-prefix complete if the following holds: if the prefixes of an infinite sequence occur as factors arbitrarily far in the tail of a second sequence in W, the first sequence is also in W. (A factor, aka substring, is a subsequence of consecutive elements.) Formally, $W \subseteq C^\omega$ is factor-prefix complete if the following holds: $\forall \gamma \in C^\omega, (\exists \gamma' \in W, \forall n, m \in \mathbb{N}, \exists k \in \mathbb{N}, \gamma_{\leq n} = \gamma'_{m+k} \dots \gamma'_{m+k+n}) \Rightarrow \gamma \in W$.

In Theorem 7 below, a distribution is said to be positive if it assigns only positive masses. A (stochastic) strategy is said to be constant if it is a constant function, i.e. it returns always the same distribution, which is stronger than being Markovian (aka memoryless, positional).

▶ **Theorem 7** (semi-random determinacy). *Consider games* $g_q = \langle A_1, A_2, Q, q, \delta, C, \mathrm{col}, W\rangle$ *parametrized by* $q \in Q$*. If* A_1 *and* A_2 *are finite, if* W *is factor-prefix complete and closed under interleaving and prefix removal, the following are equivalent.*

1. *for all* $q \in Q$*,* Player 1 *has no winning strategies in* g_q*.*
2. *for all* $q \in Q$*,* Player 2 *has a constant, positive, stochastic strategy almost-sure winning* g_q*.*
3. *for all* $q \in Q$ *every* Player 2 *stochastic strategy involving probabilities bounded away from* 0 *(i.e. with positive infimum) almost-sure wins* g_q*.*

So in the setting of Theorem 7, either Player 1 has a winning strategy for some g_q, or every constant, positive strategy is almost-sure winning, hence the determinacy. Also note that semi-random determinacy implies the *bounded limit-one property* from [11] for the new games: if one state has positive value, one state has value one.

3.3 The special case of stateless (i.e. one-state) games

Stateless games. Intuitively, a *stateless game (with colors)* amounts to a game with only one state, i.e. $|Q| = 1$. Formally, it is a tuple $\langle A_1, A_2, C, \mathrm{col}, W\rangle$ such that A_1, A_2, and C are non-empty sets, $\mathrm{col} : A_1 \times A_2 \to C$ (as opposed to $\mathrm{col} : Q \times A_1 \times A_2 \to C$ in the general case), and $W \subseteq C^\omega$. Histories, runs, strategies, and induced histories are defined as in the general case . It is easier to extend the trace function in this context: $\mathrm{col}(\epsilon) := \epsilon$ and $\mathrm{col}(\rho \cdot (a, b)) = \mathrm{col}(\rho) \cdot \mathrm{col}(a, b)$.

Restricting Theorem 3 to stateless games yields a simpler Corollary 8 below. (Note that restricting Theorem 2 would yield a weaker variant of Corollary 8, i.e. additionally assuming closedness under prefix removal.) Memory size and algorithmic complexity estimates could be obtained essentially by replacing $|Q|$ with 1 in Theorem 3 and Corollary 6.

▶ **Corollary 8.** *Consider a game $\langle A_1, A_2, C, \mathrm{col}, W\rangle$ with finite C and interleaving-closed W. Let $C_1, \dots, C_k$ be the elements of $\{\mathrm{col}(a, A_2) \mid a \in A_1\}$. The following are equivalent.*

1. Player 1 *has a winning strategy (resp. finite-memory winning strategy).*
2. $\forall(c_1, \dots, c_k) \in C_1 \times \dots \times C_k$, $W \cap \{c_1, \dots, c_k\}^\omega \neq \emptyset$ *(resp.* $W \cap \{c_1, \dots, c_k\}^\omega \cap \mathrm{reg}_C \neq \emptyset$*), where* reg_C *are the regular infinite sequences over* C.

Restricting Theorem 7 to stateless games cancels the universal quantification over states, but an even stronger version can be obtained: finiteness of A_1 and prefix removal closedness are dropped, and the assumption on factor-prefix completeness is weakened to factor-set completeness, as below.

Factor-set completeness. A language of infinite sequences is called *factor-set complete* if the following holds: if a sequence in the language has factors of unbounded length over some C_0, the language has a sequence over C_0. This is formally defined by contraposition: $W \subseteq C^\omega$ is factor-set complete if for all $C_0 \subseteq C$ and for all $\rho \in W$, we have $W \cap C_0^\omega = \emptyset \Rightarrow \forall \rho \in W, \exists m \in \mathbb{N}, \forall n \in \mathbb{N}, \exists i \in \mathbb{N}, i < m \wedge \rho_{n+i} \notin C_0$.

▶ **Observation 9.** *Factor-prefix completeness implies factor-set completeness (finite alphabets).*

▶ **Theorem 10** (Stateless semi-random determinacy). *Consider a stateless game $\langle A_1, A_2, C, \mathrm{col}, W\rangle$ with finite C and A_2. Let us assume that W is interleaving-closed and factor-set complete. Then either* Player 1 *has a winning strategy, or every* Player 2 *constant, positive, stochastic strategy is almost-sure winning.*

4 The proofs

Theorems 2 and 3 characterize a concurrent game by finitely many one-player games. A natural idea would be to split their proof into two parts: first, reduce the problem to turn-based games via the well-known observation that a player has a winning strategy in a concurrent game iff she has one in the sequential version of the game where she plays first; second, use similar techniques as in [18]. For this to work, the sequential versions of the

	b_1	b_2
a_1	$1, 1$	$2, 1$
a_2	$1, -1$	$0, -1$
a_3	$-1, 0$	$-2, 0$

$1, 1$	$1, 1$	$1, 1$	$1, 1$	$2, 1$
$1, -1$	$1, -1$	$0, -1$	$0, -1$	$1, -1$
$-1, 0$	$-2, 0$	$-1, 0$	$-2, 0$	$-1, 0$

Figure 2 A concurrent Minkowski game and its derived games.

concurrent games must allow for colorless transitions, or a fresh color should be used for the transitions where Player 1 plays. This raises three issues: first, true colors should occur infinitely often in every run in these turn-based games, which would require a more complex notion of turn-based game (and considering only games with strict player alternation does not help, as this property is lost during the induction); second, the winning condition should be rephrased to take the fresh color into account, and so should its closedness properties; third, it would be much more difficult to obtain stronger results for the one-state concurrent games, since the one-state property may be hard to track through the translation into turn-based games. Instead, this article overcomes the concurrency directly thanks to Lemma 11.

▶ **Lemma 11.** *Let $(X_i)_{i\in I}$ be a family of sets. Then*
$\forall f : \prod_{i\in I} X_i \to I, \exists i \in I, \forall x \in X_i, \exists y \in \prod_{i\in I} X_i, y_i = x \wedge f(y) = i.$

Proof. Towards a contradiction, let us assume the negation of the claim, i.e. $\exists f : \prod_{i\in I} X_i \to I, \forall i \in I, \exists x \in X_i, \forall y \in \prod_{i\in I} X_i, y_i \neq x \vee f(y) \neq i$. By collecting one witness $x =: z_i$ for each i, one constructs $z \in \prod_{i\in I} X_i$ such that $\forall y \in \prod_{i\in I} X_i, y_i \neq z_i \vee f(y) \neq i$. In particular, taking $y := z$ yields $z_i \neq z_i \vee f(z) \neq i$ for all i, which contradicts the type of f. ◀

Consider the one-state game g in Figure 2 (to the left), where each cell encloses one vector of the real plane. Player 1's objective is that the sum of the outcome vectors remains bounded, which is closed under interleaving and prefix removal, so g is a concurrent version of the Minkowski games [24]. There are $2^3 = 8$ delayed responses, and five of the corresponding one-player games $g_0, \dots g_7$ are displayed to the right in Figure 2. Player 1 wins $g_0, \dots, g_7$, since for each $i \leq 7$ the vector $(0, 0)$ is in the convex hull of the three vectors defining g_j. The idea is to let Player 1 play g as if she were playing $g_0, \dots, g_7$ in parallel, more specifically in an interleaved way. Then, summing up the eight bounded trajectories yields a bounded trajectory for g.

The main difficulty to play the $g_0, \dots, g_7$ in an interleaved way is that at every stage, Player 1 should pick an action such that whichever action Player 2 chooses, the resulting vector is exactly the expected one by the (fixed) winning strategy for *some* g_j. Let $f : \{1, 2\}^3 \to \{a_1, \dots, a_3\}$ be the function that tells which action should be played currently in each of the $2^3 = 8$ one-player games. By Lemma 11 there exists an action a_i such that the following holds: if Player 2 chooses b_1, there exists g_j expecting the vector in the cell (a_i, b_1), and likewise if Player 2 chooses b_2, there exists g_k expecting the vector in the cell (a_i, b_2).

Let us now quickly mention semi-random determinacy. The proof of Theorem 7 below uses similar techniques as, e.g., a proof in [24].

Proof of 1 $\Rightarrow$ 3 from Theorem 7. Let $p \in]0, \frac{1}{|A_2|}]$ and let τ be a Player 2 stochastic strategy that always assigns probability at least p to every action.

For all $q \in Q$, by contraposition of Theorem 2 let t_q be a delayed response (in g_q) such that Player 1 loses the one-player game $g_q(t_q)$. For all $n \in \mathbb{N}$, anytime a play reaches the state q, the probability that from then on Player 2 follows t_q for n rounds in a row, as if second-guessing Player 1, is greater than or equal to p^n.

Consider a play where Player 2 follows τ. Let q be a state that is visited infinitely often. (Such a state exists since Q is finite.) Thanks to the argument above, for all $n \in \mathbb{N}$, the probability that, at some point, Player 2 follows t_q for n rounds in a row from q on is one. Since the countable intersection of measure-one sets has also measure one, the probability that, for all $n \in \mathbb{N}$, at some point Player 2 follows t_q for n rounds in a row from q on is one.

Let $(\rho^n)_{n\in\mathbb{N}}$ be the corresponding full histories. Since A_1 and A_2 are finite, the tree induced by prefix closure of the $(\rho^n)_{n\in\mathbb{N}}$ is finitely branching, so by Koenig's Lemma it has an infinite path $\boldsymbol{\rho}$, which corresponds to Player 2 following t_q infinitely many rounds in a row. So $\mathrm{col}(\boldsymbol{\rho}) \notin W$. By factor-prefix closedness the original play is also losing for Player 1, i.e. winning for Player 2. ◀

5 Applications

Abstract assumptions need not only be general, they also need to be practical. Section 5.1 shows that the closedness and completeness axioms enjoy nice algebraic properties: individually, w.r.t. Boolean combination, as well as collectively via the derived closure or completion operators. Section 5.2 mentions several classical or recent winning conditions from computer science and tells which of them satisfy the closedness and completeness axioms. Section 5.3 introduces the notion of bounded residual load as an alternative to the finitary fairness [2], and uses it to define a finitary variant of the ω-regular languages that satisfies the closedness and completeness axioms.

5.1 Algebraic properties of the closedness and completeness axioms

Lemma 12 below shows how the axioms behave w.r.t. Boolean combination.

▶ **Lemma 12.**
1. *The set of the factor-set complete languages is closed under union.*
2. *The set of the interleaving-closed languages is closed under intersection.*
3. *The set of the factor-prefix complete languages is closed under intersection and union.*

The set of the interleaving-closed languages is not closed under union: $\{0^\omega\}$ and $\{1^\omega\}$ are closed under interleaving (and by prefix removal), but $\{0^\omega, 1^\omega\}$ is not. The set of the interleaving-closed languages is not closed under complementation: the interleaving of two infinite sequences that are not eventually constant is not eventually constant, but interleaving the eventually constant sequences 0^ω and 1^ω may yield $(01)^\omega$. The set of the factor-set complete languages is not closed under intersection: indeed, both two-element sets $\{0(12)0(12)^2 0(12)^3 0\dots, (12)^\omega\}$ and $\{0(12)0(12)^2 0\dots, (112)^\omega\}$ are factor-set complete, but their intersection $\{0(12)0(12)^2 0\dots\}$ is not. The set of the factor-set (-prefix) complete languages is not closed under complementation: $\{1^\omega\}$ is factor-set (-prefix) complete, but $\{0,1\}^\omega \setminus \{1^\omega\}$ is not.

The closedness under interleaving and prefix removal, and the factor-prefix completeness induce closure operators. If a relevant winning condition fails to satisfy an equaly relevant axiom, such an operator conveniently constructs a (more generous, axiom satisfying) variant of the winning condition. The closure by prefix removal of a set consists in adding the tails of the sequences from the set; the closure by interleaving consists in adding sequences obtained by interleaving the sequences from the set; and the factor-prefix completion consists in adding the sequences whose prefixes occur arbitrarily far in a sequence from the set. Note that factor-set completeness does not induce a canonical closure operator due to the existential quantifier in its definition.

Lemma 13 below shows that the operators behave as expected. This is not for granted in general, as one may need to perform the addition operation an ordinal number of times. Here, one step suffices, which is convenient if computation is of concern.

▶ **Lemma 13.**
1. *Closure by prefix removal yields sets that are closed under prefix removal.*
2. *Closure by interleaving yields sets that are closed under interleaving*
3. *Factor-prefix completion yields sets that are factor-prefix complete.*

Lemma 14 shows that the operators preserve the existing properties. (Lemma 13 is invoked as a proof technique.)

▶ **Lemma 14.**
1. *Closure by prefix removal preserves closedness under interleaving.*
2. *Closure by prefix removal preserves factor-set and factor-prefix completeness.*
3. *Closure by interleaving preserves closedness under prefix removal.*
4. *Closure by interleaving preserves factor-set and factor-prefix completeness.*
5. *Factor-prefix completion preserves closedness under prefix removal.*

5.2 Concrete winning conditions

The non-comprehensive list below displays classical or recent winning conditions from computer science. It especially shows that new winning conditions obtained by conjunction of older winning conditions have been recently studied, e.g. in [7] and [4].

Parity $C := \{0, 1, \dots n\}$ for some $n \in \mathbb{N}$. A sequence is winning iff the least number occurring infinitely many times in the sequence is even.

Muller $C := \{0, 1, \dots n\}$ for some $n \in \mathbb{N}$. Let $M \subseteq \mathcal{P}(C)$ be a set of subsets of C. A sequence is winning iff the numbers occurring infinitely many times in the sequence constitute a set in M.

Mean-payoff $C = \mathbb{R}$, and a sequence is winning iff the limit superior of the partial sums is non-negative: $(u_n)_{n\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$ is winning iff $\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} u_n \geq 0$. (Variants exist with limit inferior or positivity instead of non-negativity.)

Energy $C = \mathbb{R}$, and a sequence is winning iff its partial sums are non-negative: $(u_n)_{n\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$ is winning iff $\forall n \in \mathbb{N}, \sum_{i=0}^{n} u_n \geq 0$.

Boundedness [24] $C = \mathbb{R}^d$, and a sequence is winning iff its partial sums are uniformly bounded: $(u_n)_{n\in\mathbb{N}} \in (\mathbb{R}^d)^{\mathbb{N}}$ is winning iff $\exists b \forall n \in \mathbb{N}, \| \sum_{i=0}^{n} u_n \| \leq b$.

Discounted sum C is a bounded subset of $\mathbb{R}$. Let $0 < \alpha < 1$ and $t \in \mathbb{R}$. A sequence $(u_n)_{n\in\mathbb{N}} \in C^{\mathbb{N}}$ is winning iff $\sum_{n=0}^{+\infty} \alpha^n u_n \geq t$.

Energy-parity [7] $C := \mathbb{R} \times \{0, 1, \dots n\}$ for some $n \in \mathbb{N}$. The winning condition is the conjunction of the energy (first component) and the parity (second component) conditions.

Average energy [4] $C = \mathbb{R}$. The objective is to maintain a non-negative energy while keeping the average level of energy below a threshold $t \in \mathbb{R}$: a sequence $(u_n)_{n\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$ is winning iff $(\forall n \in \mathbb{N}, \sum_{i=0}^{n} u_n \geq 0) \wedge \limsup_{n\to+\infty} \frac{1}{n} \sum_{i=0}^{n} \sum_{j=0}^{i} u_j \leq t$.

▶ **Observation 15.**
1. *The parity, mean-payoff, energy, boundedness, energy-parity, and average energy conditions are all closed under interleaving. (It uses Lemma 12.2 to deal with energy-parity and average energy.)*

2. *The Muller and discounted sum conditions are not closed under interleaving.*
3. *The boundedness condition is factor-prefix complete; the others are not.*
4. *The energy condition (thus also energy-parity and average energy) and the discounted sum condition are not closed under prefix removal; the others are.*

▶ **Corollary 16.** *The turn-based safety-mean-payoff-parity games are half-positionally determined. (By Corollary 4 and Section 5.1.)*

It may be disappointing that the Muller condition is not even closed under interleaving, but Proposition 17 below extends Theorem 2 to the concurrent Muller games. Using results from [11] is likely to yield a better algorithmic complexity, though, but the point here is mainly that Theorem 2 can be extended.

▶ **Proposition 17.** *[Similar to [11]] Consider the finite games* $\langle A_1, A_2, Q, q_0, \delta, C, \mathrm{col}, W\rangle$ *where* W *is a Muller condition. Deciding the existence of a* Player 1 *winning (finite-memory) strategy can be done in big* O *of*

$$(|A_1||A_2||C||C|!)^2 \cdot (|Q||C|^2|C|!)^{|Q||C||C|!\left(2^{|Q||C|^2|C|!}\right)}$$

computation steps.

5.3 Bounded residual load

Unlike Theorems 2 and 3, Theorems 7 and 10 are not likely to be extended to include ω-regular languages. Before defining a variant of the ω-regular languages that satisfies the closedness and completeness properties from this article, let us consider notions of fairness that can be defined via a predicate S on $\mathbb{N} \times \mathbb{N} \times C^\omega$. Intuitively $S(n, d, \boldsymbol{\gamma})$ is supposed to mean that the sequence $\boldsymbol{\gamma}$ has satisfied, with delay at most d, a request that was formulated in $\boldsymbol{\gamma}$ at time n.

There are several reasonable ways to express the good behavior of an infinite sequence using the $S(n, d, \boldsymbol{\gamma})$. The classical definition of *fairness* requires that all problems be eventually solved (see F below), or cofinitely many problems (see FCI below), for a usual weakening that ensures prefix independence of the condition. Arguing that this kind of fairness gives no guarantee about response time, [11] strengthened fairness into *finitary fairness*, which requires the existence of a uniform bound on the waiting time (see FF below).

Yet another variant, *bounded residual load* (BRL), is introduced below. It says that $\boldsymbol{\gamma} \in C^\omega$ satisfies S wrt bounded residual load, if the number of problems that have currently not yet been solved is uniformly bounded over time.

1. $F(\boldsymbol{\gamma}) := \forall n \in \mathbb{N}, \exists d \in \mathbb{N},\ S(n, d, \boldsymbol{\gamma})$
2. $FCI(\boldsymbol{\gamma}) := |\{n \in \mathbb{N} \mid \forall d \in \mathbb{N}, \neg S(n, d, \boldsymbol{\gamma})\}| < \infty$
3. $FF(\boldsymbol{\gamma}) := \exists d \in \mathbb{N}, \forall n \in \mathbb{N},\ S(n, d, \boldsymbol{\gamma})$
4. $BRL(\boldsymbol{\gamma}) := \exists b \in \mathbb{N}, \forall n \in \mathbb{N},\ b \geq |\{k \in \mathbb{N} \mid k \leq n \wedge \neg S(k, n-k, \boldsymbol{\gamma})\}|$

▶ **Observation 18.**
1. $FF(\gamma) \Rightarrow F(\gamma) \quad \wedge \quad F(\gamma) \Rightarrow FCI(\gamma)$
2. $FF(\gamma) \Rightarrow BRL(\boldsymbol{\gamma}) \quad \wedge \quad BRL(\boldsymbol{\gamma}) \Rightarrow FCI(\gamma)$
3. F *and* BRL *are incomparable in general.*

The finitary fairness and the like may be too strict for some applications: gladly accepting to wait b time units, but categorically refusing to wait $b+1$ time units sounds unusual indeed. Instead, the system (which is responsible for solving the problems) could pay a penalty for each problem spending each time unit unsolved. Thanks to the bounded residual load, one has then the guarantee that the amount of money to be paid per time unit is bounded.

It is possible to combine the two ideas, though: by setting an acceptable response time and an acceptable uniform bound on the number of missed deadlines. This however, turns out to be equivalent to the simple BRL, which argues for the robustness of the concept.

▶ **Observation 19.** *Let* $BRLD(\boldsymbol{\gamma}) := \exists b, d \in \mathbb{N}, \forall n \in \mathbb{N},\ b \geq |\{k \in \mathbb{N} \mid k \leq n - d \wedge \neg S(k, n-k, \boldsymbol{\gamma})\}|$*, then* $BRLD(\boldsymbol{\gamma}) \Leftrightarrow BRL(\boldsymbol{\gamma})$*.*

A second justification for the BRL is that it has nice properties that the other notions of fairness lack when $S(n, d, \boldsymbol{\gamma})$ is defined to minic ω-regular languages, as shown below. Consider a non-empty set C of colors and a function $\mathcal{C} : C \to \mathcal{P}(C^*)$. A sequence $\boldsymbol{\gamma} \in C^\omega$ is said to satisfy $\mathcal{C}$ from position n after delay d, denoted $S_{\mathcal{C}}(n, d, \boldsymbol{\gamma})$, if the following holds.

$$\exists u \in \mathcal{C}(\boldsymbol{\gamma}_n), \exists (k_1, \dots, k_{|u|}) \in \mathbb{N}^{|u|},\ n < k_1 < \dots < k_{|u|} \leq n + d \wedge \forall i \leq |u|,\ u_i = \boldsymbol{\gamma}_{k_i}$$

Intuitively, each color is a problem or a request, and the problem may be solved in several ways, each way consisting in enumerating suitable colors quickly. (This might very well correspond to the positive fragment of some bounded-time temporal logic.) To simulate the parity condition, one can set $C := \mathbb{N}$ and $\mathcal{C}(2n) := \{\{k\} \mid k \in \mathbb{N}\}$ and $\mathcal{C}(2n+1) := \{\{2k\} \mid k \in \mathbb{N} \wedge k \leq n\}$ for all $n \in \mathbb{N}$. The corresponding $BRL_{\mathcal{C}}$ is the parity condition with bounded residual load.

Lemma 20 below says that however $\mathcal{C}$ may be instantiated, all Theorems 2, 3, 7, and 10 can be applied with the $BRL_{\mathcal{C}}$ winning condition.

▶ **Lemma 20.** *For every non-empty set* C *of colors and every function* $\mathcal{C} : C \to \mathcal{P}(\mathbb{N}^C)$*, the winning condition* $BRL_{\mathcal{C}}$ *is closed under prefix removal and interleaving, and factor-prefix complete.*

Even when $\mathcal{C}$ simulates the parity condition as above, none of the corresponding $F_{\mathcal{C}}$, $FCI_{\mathcal{C}}$, or $FF_{\mathcal{C}}$ is both closed under interleaving and factor-set complete. $FF_{\mathcal{C}}$ is not closed under interleaving: $FF_{\mathcal{C}}((01)^\omega)$ and $FF_{\mathcal{C}}((23)^\omega)$, but $\neg FF_{\mathcal{C}}(\boldsymbol{\gamma})$, where $\boldsymbol{\gamma} := (23)01(23)^2 01 \dots 01(23)^n 01 \dots$ can be obtained by interleaving $(01)^\omega$ and $(23)^\omega$. $FCI_{\mathcal{C}}$ is not factor set-complete: $FCI_{\mathcal{C}}(\boldsymbol{\gamma})$, where $\boldsymbol{\gamma} := 101^2 01^3 \dots 01^n 0 \dots$, but $\neg FCI_{\mathcal{C}}(1^\omega)$ altough factors of 1's occur with arbitrary length in $\boldsymbol{\gamma}$. $F_{\mathcal{C}}$ is neither: first, $F_{\mathcal{C}}((10)^\omega)$ and $F_{\mathcal{C}}(2^\omega)$, but $\neg F_{\mathcal{C}}(1 \cdot 2^\omega)$, altough $1 \cdot 2^\omega$ can be obtained by interleaving $(10)^\omega$ and 2^ω; second, as above for $FCI_{\mathcal{C}}$. Note that the window-parity condition [8],[5] is not closed under interleaving either, as again exemplified by $(01)^\omega$ and $(23)^\omega$.

References

1 Rajeev Alur, Marco Faella, Sampath Kannan, and Nimit Singhania. Hedging Bets in Markov Decision Processes. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:20, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CSL.2016.29.

2 Rajeev Alur and Thomas A. Henzinger. Finitary fairness. *ACM Trans. Program. Lang. Syst.*, 20(6):1171–1194, 1998. doi:10.1145/295656.295659.

3 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Concurrent games with ordered objectives. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures*, pages 301–315, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

4 Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim G. Larsen, and Simon Laursen. Average-energy games. *Acta Informatica*, Jul 2016. doi:10.1007/s00236-016-0274-1.

5 Véronique Bruyère, Quentin Hautem, and Mickael Randour. Window parity games: an alternative approach toward parity games with time bounds. In *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016.*, pages 135–148, 2016. doi:10.4204/EPTCS.226.10.

6 Krishnendu Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388(1):181–198, 2007. doi:10.1016/j.tcs.2007.07.047.

7 Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming*, pages 599–610, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

8 Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin. Looking at mean-payoff and total-payoff through windows. *Inf. Comput.*, 242:25–52, 2015. doi:10.1016/j.ic.2015.03.010.

9 Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in Ω-regular games. *ACM Trans. Comput. Logic*, 11(1):1:1–1:27, nov 2009. doi:10.1145/1614431.1614432.

10 Thomas Colcombet and Damian Niwiński. On the positional determinacy of edge-labeled games. *Theoretical Computer Science*, 352(1):190–196, 2006. doi:10.1016/j.tcs.2005.10.046.

11 Luca de Alfaro and Thomas A. Henzinger. Concurrent omega-regular games. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*, pages 141–154, 2000. doi:10.1109/LICS.2000.855763.

12 Nachum Dershowitz, D. N. Jayasimha, and Seungjoon Park. *Bounded Fairness*, pages 304–317. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi:10.1007/978-3-540-39910-0_14.

13 Hugo Gimbert and Florian Horn. Solving simple stochastic tail games. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 847–862, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=1873601.1873670.

14 Hugo Gimbert and Wiesław Zielonka. Games where you can play optimally without any memory. In *CONCUR 2005 - Concurrency Theory*, volume 3653 of *Lecture Notes in Computer Science*, pages 428–442. Springer Berlin Heidelberg, 2005.

15 Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, and Michael Wooldridge. Nash equilibria in concurrent games with lexicographic preferences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1067–1073, 2017. doi:10.24963/ijcai.2017/148.

16 John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979. doi:10.1016/0304-3975(79)90041-0.

17 Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969. doi:10.1016/S0022-0000(69)80011-5.

18 Eryk Kopczyński. Half-positional determinacy of infinite games. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 336–347, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

19 Stéphane Le Roux. Infinite sequential Nash equilibrium. *Logical Methods in Computer Science*, 9, 2013. Special Issue for the Conference "Computability and Complexity in Analysis" (CCA 2011).

20 Stéphane Le Roux. From winning strategy to nash equilibrium. *Math. Log. Q.*, 60(4-5):354–371, 2014. doi:10.1002/malq.201300034.

21 Stéphane Le Roux. Infinite subgame perfect equilibrium in the hausdorff difference hierarchy. In *Topics in Theoretical Computer Science - The First IFIP WG 1.8 International Conference, TTCS 2015, Tehran, Iran, August 26-28, 2015, Revised Selected Papers*, pages 147–163, 2015. doi:10.1007/978-3-319-28678-5_11.

22 Stéphane Le Roux and Arno Pauly. Infinite sequential games with real-valued payoffs. In *Proceedings of LiCS*, 2014.

23 Stéphane Le Roux and Arno Pauly. Extending finite memory determinacy to multiplayer games. In *Proceedings of the 4th International Workshop on Strategic Reasoning, SR 2016, New York City, USA, 10th July 2016.*, pages 27–40, 2016. doi:10.4204/EPTCS.218.3.

24 Stéphane Le Roux, Arno Pauly, and Jean-François Raskin. Minkowski games. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, pages 50:1–50:13, 2017. doi:10.4230/LIPIcs.STACS.2017.50.

25 Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975. URL: http://www.jstor.org/stable/1971035.

26 Donald A. Martin. The determinacy of blackwell games. *Journal of Symbolic Logic*, 63(4):1565–1581, 1998. doi:10.2307/2586667.

27 Stéphane Le Roux and Arno Pauly. Extending finite-memory determinacy to multi-player games. *Information and Computation*, pages –, 2018. doi:10.1016/j.ic.2018.02.024.

28 Wolfgang Thomas. Languages, automata, and logic. In Salomaa A. Rozenberg G., editor, *Handbook of Formal Languages*. Springer, Berlin, Heidelberg, 1997.