Extending Finite Memory Determinacy to Multiplayer Games

Stéphane Le Roux
Département d'Informatique
Université Libre de Bruxelles, Belgique
Stephane.Le.Roux@ulb.ac.be

Arno Pauly
Département d'Informatique
Université Libre de Bruxelles, Belgium
Arno.Pauly@cl.cam.ac.uk

We show that under some general conditions the finite memory determinacy of a class of two-player win/lose games played on finite graphs implies the existence of a Nash equilibrium built from finite memory strategies for the corresponding class of multi-player multi-outcome games. This generalizes a previous result by Brihaye, De Pril and Schewe. For most of our conditions we provide counterexamples showing that they cannot be dispensed with.

Our proofs are generally constructive, that is, provide upper bounds for the memory required, as well as algorithms to compute the relevant winning strategies.

1 Introduction

The usual model employed for synthesis are sequential two-player win/lose games played on finite graphs. The vertices of the graph correspond to states of a system, and the two players jointly generate an infinite path through the graph (the *run*). One player, the protagonist, models the aspects of the system under the control of the designer. In particular, the protagonist will win the game iff the run satisfies the intended specification. The other player is assumed to be fully antagonistic, thus wins iff the protagonist loses. One then would like to find winning strategies of the protagonist, that is, a strategy for her to play the game in such a way that she will win regardless of the antagonist's moves. Particularly desirable winning strategies are those which can be executed by a finite automaton.

Classes of games are distinguished by the way the winning conditions (or more generally, preferences of the players) are specified. Typical examples include:

- Muller conditions, where only the set of vertices visited infinitely many times matters;
- Parity conditions, where each vertex has a priority, and the winner is decided by the parity of the least priority visited infinitely many times;
- Energy conditions, where each vertex has an energy delta (positive or negative), and the protagonist loses if the cumulative energy values ever drop below 0;
- Discounted payoff conditions, where each vertex has a payoff value, and the outcome is determined by the discounted sum of all payoffs visited with some discount factor $0 < \lambda < 1$;
- Combinations of these, such as energy parity games, where the protagonist has to simultaneously ensure that the least parity visited infinitely many times is odd and that the cumulative energy value is never negative.

Our goal is to dispose of two restrictions of this setting: First, we would like to consider any number of players; and second allow them to have far more complicated preferences than just preferring winning over losing. The former generalization is crucial in a distributed setting (also e.g. [4, 6]): If different designers control different parts of the system, they may have different specifications they would like to enforce, which may be partially but not entirely overlapping. The latter seems desirable in a broad range of contexts. Indeed, rarely is the intention for the behaviour of a system formulated entirely in black and white: We prefer a programm just crashing to accidently erasing our hard-drive; we prefer a programm to complete its task in 1 minute to it taking 5 minutes, etc. We point to [13] for a recent survey on such notions of quality in synthesis.

Rather than achieving this goal by revisiting each individual type of game and proving the desired results directly (e.g. by generalizing the original proofs of the existence of winning strategies), we shall provide a transfer theorem: In Theorem 5, we will show that (under some conditions), if the two-player win/lose version of a game is finite memory determined, the corresponding multi-player multi-outcome games all have finite memory Nash equilibria.

This result is more general than a similar one obtained by BRIHAYE, DE PRIL and SCHEWE [4],[20, Theorem 4.4.14]. A particular class of games covered by our result but not the previous one are (a variant of) energy parity games as introduced by CHATTERJEE and DOYEN [8]. The high-level proof idea follows earlier work by the authors on equilibria in infinite sequential games, using Borel determinacy as a blackbox [15][1] – unlike the constructions there (cf. [16]), the present ones however are constructive and thus give rise to algorithms computing the equilibria in the multi-player multi-outcome games given suitable winning strategies in the two-player win/lose versions.

Echoing DE PRIL in [20], we would like to stress that our conditions apply to the preferences of each player individually. For example, some players could pursue energy parity conditions, whereas others have preferences based on Muller conditions: Our results apply just as they would do if all players had preferences of the same type.

For several of the conditions in our main theorem we also provide examples showing that they cannot be removed.

2 Background

A two-player win/lose game played on a finite graph is specified by a directed graph (V,E) where every vertex has an outgoing edge, a starting vertex $v_0 \in V$, two sets $V_1 \subseteq V$ and $V_2 := V \setminus V_1$, and a *winning condition* $W \subseteq V^\omega$. Starting from v_0, the players move a token along the graph, ω times, with player $a \in \{1,2\}$ picking and following an outgoing edge whenever the current vertex lies in V_a. Player 1 wins iff the infinite sequence of visited vertices is in W.

For $a \in \{1,2\}$ let $\mathscr{H}_a$ be the set of finite paths in (V,E) starting at v_0 and ending in V_a. Let $\mathscr{H} := \mathscr{H}_1 \cup \mathscr{H}_2$ be the possible *finite histories* of the game, and let $[\mathscr{H}]$ be the infinite ones. For clarity we may write $[\mathscr{H}_g]$ instead of $[\mathscr{H}]$ for a game g. A *strategy* of player $a \in \{1,2\}$ is a function of type $\mathscr{H}_a \to V$ such that $(v, s(hv)) \in E$ for all $hv \in \mathscr{H}_a$. A pair of strategies (s_1,s_2) for the two players induces a run $\rho \in V^\omega$: Let $s := s_1 \cup s_2$ and set $\rho(0) := v_0$ and $\rho(n+1) := s(\rho(0)\rho(1)\dots\rho(n))$. For all strategies s_a of player a let $\mathscr{H}(s_a)$ be the finite histories in $\mathscr{H}$ that are compatible with s_a, and let $[\mathscr{H}(s_a)]$ be the infinite ones. A strategy s_a is said to be winning if $[\mathscr{H}(s_a)] \subseteq W$, *i.e.* a wins regardless of her opponent's moves.

A *strategic implementation* for player a using m bits of memory is a function $\sigma : V \times \{0,1\}^m \to V \times \{0,1\}^m$ that describes the two simultaneous updates of player a upon arrival at a vertex v if its memory content was M just before arrival: $(v,M) \mapsto \pi_2 \circ \sigma(v,M)$ describes the memory update and $(v,M) \mapsto \pi_1 \circ \sigma(v,M)$ the choice for the next vertex. This choice will be ultimately relevant only if $v \in V_a$, in which case we require that $(v, \pi_1 \circ \sigma(v,M)) \in E$.

[1]Precursor ideas are also present in [14] and [18] (the specific result in the latter was joint work with Neymann).

Together with an initial memory content $M_\varepsilon \in \{0,1\}^m$, a strategic implementation provides a finite memory strategy. The memory content after some history is defined by induction: $M_\sigma(M_\varepsilon, \varepsilon) := M_\varepsilon$ and $M_\sigma(M_\varepsilon, hv) := \pi_2 \circ \sigma(v, M_\sigma(M_\varepsilon, h))$ for all $hv \in \mathscr{H}$. The *finite-memory strategy* s_a induced by the strategic implementation σ together with initial memory content M_ε is defined by $s_a(hv) := \pi_1 \circ \sigma(v, M_\sigma(M_\varepsilon, h))$ for all $hv \in \mathscr{H}_a$. If not stated otherwise, we will assume the initial memory to be 0^m.

A (general) game played on a finite graph is specified by a directed graph (V,E), a set of agents A, a cover $\{V_a\}_{a\in A}$ of V via pairwise disjoint sets, the starting vertex v_0, and for each player a a preference relation $\prec_a \subseteq [\mathscr{H}] \times [\mathscr{H}]$. The notions of strategies and induced runs generalize in the obvious way. In particular, instead of a pair of strategies (one per player), we consider families $(s_a)_{a\in A}$, which are called strategy profiles.

The concept of a winning strategy no longer applies though. Instead, we use the more general notion of a Nash equilibrium: A family of strategies $(s_a)_{a\in A}$ is a Nash equilibrium, if there is no player $a_0 \in A$ and alternate strategy s'_{a_0} such that a would prefer the run induced by $(s_a)_{a\in A\setminus\{a_0\}} \cup (s'_a)_{a\in\{a_0\}}$ to the run induced by $(s_a)_{a\in A}$. Intuitively, no player can gain by unilaterally deviating from a Nash equilibrium. Note that the Nash equilibria in two-player win/lose games are precisely those pairs of strategy where one strategy is a winning strategy.

The transfer of results from the two-player win/lose case to the general case relies on the idea that each general game induces a collection of two-player win/lose games, namely the threshold games of the future games, as below.

Definition 1 (Future game and one-vs-all threshold game)**.**
Let $g = \langle (V,E), v_0, A, \{V_a\}_{a\in A}, (\prec_a)_{a\in A}\rangle$ be a game played on a finite graph.

- Let $a_0 \in A$ and $\rho \in [\mathscr{H}]$, the one-vs-all threshold game $g_{a_0,\rho}$ for a_0 and ρ is the win-lose two-player game played on (V,E), starting at v_0, with vertex subsets V_{a_0} and $\bigcup_{a\in A\setminus\{a_0\}} V_a$, and with winning set $\{\rho' \in [\mathscr{H}] \mid \rho \prec_{a_0} \rho'\}$ for Player 1.
- Let $v \in V$. For paths hv and vh' in (V,E) let $hv\hat{} vh' := hvh'$.
- For all $h \in \mathscr{H}$ with last vertex v let $g^h := \langle (V,E), v, A, \{V_a\}_{a\in A}, (\prec^h_a)_{a\in A}\rangle$ be called the future game of g after h, where for all $\rho, \rho' \in [\mathscr{H}_{g^h}]$ we set $\rho \prec^h_a \rho'$ iff $h\hat{}\rho \prec_a h\hat{}\rho'$. If s is a strategy in g, let s^h be the strategy in g^h such that $s^h(h') := s(h\hat{}h')$ for all $h' \in \mathscr{H}_{g^h}$.

Observation 2. Let $g = \langle (V,E), v_0, A, \{V_a\}_{a\in A}, (\prec_a)_{a\in A}\rangle$ be a game played on a finite graph.

1. g and its thresholds games have the same strategies.
2. for all $h, h' \in \mathscr{H}$ ending with the same vertex the games g^h and $g^{h'}$ have the same (finite-memory) strategies.
3. g, its future games, and their thresholds games have the same strategic implementations.
4. If a strategy s_a in g is finite-memory, for all $h \in \mathscr{H}$ the strategy s^h_a is also finite-memory.

Proof. We only prove the fourth claim. Since s_a is a finite-memory strategy, it comes from some strategic implementation σ with initial memory M_ε. We argue that σ with initial memory $M_\sigma(M_\varepsilon, h))$ implements s^{hv}_a: First, $s^{hv}_a(v) = s_a(hv) = \pi_1 \circ \sigma(v, M_\sigma(M_\varepsilon, h)) = \pi_1 \circ \sigma(v, M_\sigma(M_\sigma(M_\varepsilon, h), \varepsilon))$; second, for all $h'v' \in \mathscr{H}^{hv}$ we have $s^{hv}_a(vh'v') = s_a(hvh'v') = \pi_1 \circ \sigma(v', M_\sigma(M_\varepsilon, hvh')) = \pi_1 \circ \sigma(v', M_\sigma(M_\sigma(M_\varepsilon, h), vh'))$. □

We will employ some additional restrictions on preferences: a preference relation $\prec \subseteq [\mathscr{H}] \times [\mathscr{H}]$ is called *prefix-linear*, if $\rho \prec \rho' \Leftrightarrow h\hat{}\rho \prec h\hat{}\rho'$ for all $\rho, \rho', h\hat{}\rho \in [\mathscr{H}]$. It is *prefix-independent*, if $\rho \prec \rho' \Leftrightarrow h\hat{}\rho \prec \rho'$ and $\rho' \prec \rho \Leftrightarrow \rho' \prec h\hat{}\rho$ for all $\rho, \rho', h\hat{}\rho \in [\mathscr{H}]$. Clearly, a prefix-independent preference is prefix-linear.

As a further generalization, we will consider *automatic-piecewise prefix-linear* preferences $\prec$. Here, there is an equivalence relation on $\mathscr{H}$ with equivalence classes (pieces for short) in $\overline{\mathscr{H}}$ and satisfying three constraints: First, the histories in the same piece end with the same vertex. Second, there exists a deterministic finite automaton, without accepting states, that reads histories and such that two histories are equivalent iff reading them leads to the same states. Third, for all $h\hat{}\rho, h\hat{}\rho', h'\hat{}\rho, h'\hat{}\rho' \in [\mathscr{H}]$, if $\overline{h'} = \overline{h} \in \overline{\mathscr{H}}$, then $h\hat{}\rho \prec h\hat{}\rho' \Leftrightarrow h'\hat{}\rho \prec h'\hat{}\rho'$.

Definition 3. A preference relation $\prec$ is automatic-piecewise Mont if there is an equivalence relation on $\mathscr{H}$ that is decidable by a finite automaton (with the first two constraints as for automatic-piecewise prefix linearity) such that the following holds. For every run $h_0\hat{}\rho \in [\mathscr{H}]$ that is regular (as a singleton language) and for every family $(h_n)_{n\in\mathbb{N}}$ of paths in (V,E) such that $h_0\hat{}h_1\hat{}\dots\hat{}h_n \in \overline{h_0}$ for all n, and such that $h_0\hat{}\dots\hat{}h_n\hat{}\rho \in [\mathscr{H}]$ for all n, if $h_0\hat{}\dots\hat{}h_n\hat{}\rho \prec h_0\hat{}\dots\hat{}h_{n+1}\hat{}\rho$ for all n then $h_0\hat{}\rho \prec h_0\hat{}h_1\hat{}h_2\hat{}h_3\dots$.

For a cycle h starting and ending in some $v \in V$, let $h^{[0]} = v$ and $h^{[n+1]} = h^{[n]}\hat{}h$, and finally $h^{[\omega]} = \lim_{n\to\infty} h^{[n]}$. We call $\prec$ automatic-piecewise regular-Mont, if for any regular $h_0\hat{}\rho \in [\mathscr{H}]$ and cycle h in (V,E), if $\forall n \in \mathbb{N} \quad h_0\hat{}h^{[n]} \in \overline{h_0}$ and $\forall n \in \mathbb{N} \quad h_0\hat{}h^{[n]}\hat{}\rho \prec h_0\hat{}h^{[n+1]}\hat{}\rho$, then $h_0\hat{}\rho \prec h_0\hat{}h^{[\omega]}$.

Definition 4 (Strict weak order)**.** Recall that a relation $\prec$ is called a *strict weak order* if it satisfies:

$$\begin{array}{ll} \forall x, & \neg(x \prec x) \\ \forall x,y,z, & x \prec y \wedge y \prec z \Rightarrow x \prec z \\ \forall x,y,z, & \neg(x \prec y) \wedge \neg(y \prec z) \Rightarrow \neg(x \prec z) \end{array}$$

Strict weak orders capture in particular the situation where each player cares only about a particular aspect of the run (e.g. her associated personal payoff), and is indifferent between runs that coincide in this aspect but not others (e.g. the runs with identical associated payoffs for her, but different payoffs for the other players).

3 The transfer theorem

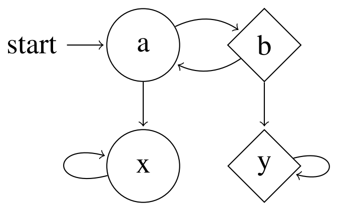
We will start this section with the statement of our first main result, showing how to transfer finite-memory determinacy from class of two-player win/lose games to the corresponding multi-player multi-outcome version. We informally sketch the proof. This is followed by the technical definitions and lemmata used in the formal proof, and then the proof itself. The section is completed by a discussion of the relevance and a comparison to prior results.

Theorem 5. Consider a game played by a set of players A on a finite graph such that

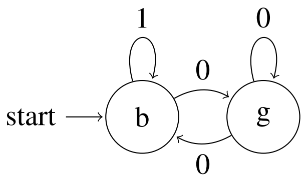
1. The $\prec_a$ are automatic-piecewise prefix-linear Mont strict weak orders with k pieces.
2. All one-vs-all threshold games of all future games are determined via strategies using m bits of memory.

Then the game has a Nash equilibrium in finite-memory strategies requiring $|A|(m+2\log k)+1$ bits of memory.

Definition 6 below rephrases Definitions 2.3 and 2.5 from [14]: The guarantee of a player is the smallest set of runs that is upper-closed w.r.t. the strict-weak-order preference of the player and includes every incomparability class (of the preference) that contains any run compatible with a given strategy of the player in the subgame at any given finite history of the game. The best guarantee of a player consists of the intersection of all her guarantees over the set of strategies.



(a)



(b)

Figure 1

More plainly spoken, the best guarantee for a player at some history is the set of runs such that the player cannot unilaterally enforce something better (for him). We will then show that each player has indeed a strategy enforcing her guarantee. Note that the notion of best guarantee for a player does not at all depend on the preferences of the other players; and as such, it is rather strenuous to consider such runs to be optimal in some sense (cf. Example 7). However, we can construct a Nash equilibrium by starting with a strategy profile where everyone is realizing their guarantee, and then adding punishments against any deviators.

Definition 6 (Player (best) future guarantee)**.** Let g be the game $\langle (V,E), v_0, A, \{V_a\}_{a\in A}, (\prec_a)_{a\in A}\rangle$ where $\prec_a$ is a strict weak order for some $a \in A$. For all $h \in \mathscr{H}$ and strategies s_a for a in g^h let $\gamma_a(h,s_a) := \{\rho \in [\mathscr{H}_{g^h}] \mid \exists \rho' \in [\mathscr{H}_{g^h}(s_a)], \neg(\rho \prec^h_a \rho')\}$ be the player future guarantee by s_a in g^h. Let $\Gamma_a(h) := \bigcap_{s_a} \gamma_a(h,s_a)$ be the best future guarantee of a in g^h.

Example 7. Let the underlying graph be as in Figure 1a, where circle vertices are controlled by Player 1 and diamond vertices are controlled by Player 2. The preference relation of Player 1 is $(ab)^\omega \succ_1 a(ba)^n x^\omega \succ_1 (ab)^n y^\omega$ and the preference relation of Player 2 is $(ab)^\omega \succ_2 (ab)^n y^\omega \succ_2 a(ba)^n x^\omega$ (in particular, both players care only about the tail of the run).

Then $\Gamma_1(a) = \{(ab)^\omega\} \cup \{a(ba)^n x^\omega \mid n \in \mathbb{N}\}$ and $\Gamma_2(a) = [\mathscr{H}]$. Player 1 realizing her guarantee means for her to move to x immediately, thus forgoing any chance of realizing the run $(ab)^\omega$. The Nash equilibrium constructed in the proof of Theorem 5 will be Player 1 moving to x and Player 2 moving to y. Note that in this particular game, the preferences of Player 2 have no impact at all on the Nash equilibrium that will be constructed.

Lemma 8. Let g be a game on a graph, let $\prec_a$ be a strict weak order preference for some player a, let $h \in \mathscr{H}$, let s_a be a strategy for a in g^h, let $h' \in \mathscr{H}(s_a)$, and let s'_a be a strategy for a in $g^{h\hat{}h'}$.

1. Then $h'\hat{}\gamma_a(h\hat{}h', s^{h'}_a) \subseteq \gamma_a(h,s_a)$ for all $h' \in \mathscr{H}(s_a)$.
2. If $\gamma_a(h\hat{}h', s'_a) \subsetneq \gamma_a(h\hat{}h', s^{h'}_a)$, there exists $\rho \in \gamma_a(h\hat{}h', s^{h'}_a)$ such that $\rho \prec^{h\hat{}h'}_a \rho'$ for all $\rho' \in \gamma_a(h\hat{}h', s'_a)$.
3. If $\gamma_a(h\hat{}h', s'_a) \subseteq \gamma_a(h\hat{}h', s^{h'}_a)$ then $h'\hat{}\gamma_a(h\hat{}h', s'_a) \subseteq \gamma_a(h,s_a)$.

Proof. 1. Let $\rho \in \gamma_a(h\hat{}h', s^{h'}_a)$, so by Definition 6 there exists $\rho' \in [\mathscr{H}(s^{h'}_a)]$ such that $\neg(\rho \prec^{h\hat{}h'}_a \rho')$, *i.e.* $\neg(h'\hat{}\rho \prec^h_a h'\hat{}\rho')$. So $h'\hat{}\rho \in \gamma_a(h,s_a)$ since $h'\hat{}\rho' \in [\mathscr{H}(s_a)] \subseteq \gamma_a(h,s_a)$.

2. Let $\rho \in \gamma_a(h\hat{}h', s^{h'}_a) \setminus \gamma_a(h\hat{}h', s'_a)$, so $\rho \prec^{h\hat{}h'}_a \rho'$ for all $\rho' \in \gamma_a(h\hat{}h', s'_a)$ by Definition 6.

3. Let $\rho \in \gamma_a(h\hat{}h', s'_a)$, so $h'\hat{}\rho \in h'\hat{}\gamma_a(h\hat{}h', s^{h'}_a)$ by assumption, so $h'\hat{}\rho \in \gamma_a(h,s_a)$ by Lemma 8.1.

□

Lemma 9. Let g be a game on a finite graph with strict weak order $\prec_a$ for some $a \in A$, and let $m \in \mathbb{N}$. In each of the threshold games for a of the future games let us assume that

1. if Player 1 has a winning strategy, she has one with memory size m. Then for all $h \in \mathscr{H}$ there exists a strategy s_a with memory size m such that $\gamma_a(h, s_a) = \Gamma_a(h)$.
2. one of the players has a winning strategy with memory size m. Then $\Gamma_a(h)$ has a regular $\prec_a^h$-minimum for all $h \in \mathscr{H}$.

Proof. 1. For all $\rho \in [\mathscr{H}_{g^h}]$, we have $\rho \notin \Gamma_a(h)$ iff Player 1 has a winning strategy in the threshold game for a and ρ in g^h. In this case let s_a^ρ be a winning strategy with memory size m. For a game with n vertices there are at most $(n2^m)^{(n2^m)}$ strategy profiles using m bits of memory (by the σ representation), so the s_a^ρ are finitely many, so at least one of them, which we name s_a, wins the threshold games for all $\rho \notin \Gamma_a(h)$. This shows that $\gamma_a(h, s_a) \subseteq \Gamma_a(h)$, so equality holds.

2. Towards a contradiction let us assume that $\Gamma_a(h)$ has a no $\prec_a^h$-least element, and let $\rho \in \Gamma_a(h)$, so there exists a $\prec_a^h$-smaller $\rho' \in \Gamma_a(h)$. Since a has no winning strategy for the threshold game for ρ' (of the future game at h), the determinacy assumption implies that the coalition of her opponents has one with memory size m. These strategies are finitely many, so one of them, s_{-a}, wins the threshold game for all $\rho \in \Gamma_a(h)$. So $\mathscr{H}_{g^h}(s_{-a}) \cap \Gamma_a(h) = \emptyset$, which contradicts Lemma 9.1. Furthermore, the run induced by s_{-a} and one finite-memory s_a from Lemma 9.1 is regular.

□

Lemma 10. Let g be a game on a finite graph with automatic piecewise prefix-linear strict weak order $\prec_a$ for some $a \in A$. If $h, h' \in H \in \overline{\mathscr{H}}$, then $\gamma_a(h, s_a) = \gamma_a(h', s_a)$ for all strategies s_a for a in g^h, and $\Gamma_a(h) = \Gamma_a(h')$.

Proof. By definition of the future games and automatic-piecewise prefix-linearity $\rho \prec_a^h \rho'$ iff $h\hat{}\rho \prec_a h\hat{}\rho'$ iff $h'\hat{}\rho \prec_a h'\hat{}\rho'$ iff $\rho \prec_a^{h'} \rho'$, so $\prec_a^h = \prec_a^{h'}$. □

Lemma 11. Let g be a game on a finite graph with automatic-piecewise prefix-linear strict weak order $\prec_a$ for some $a \in A$. Let $m \in \mathbb{N}$ and assume that the threshold games for a of the future games are determined *via* size-m strategies. Let $H \in \overline{\mathscr{H}}$.

1. There exists a strategy s_a^H with memory size m such that $\gamma_a(h, s_a^H) = \Gamma_a(h)$ for all $h \in H$.
2. There exists a size-m strategy s_{-a}^H for Player 2 in g^H (*i.e.* g^h for any $h \in H$) that is winning the threshold game for a and ρ in g^H for all $\rho \in \Gamma_a(H)$ (*i.e.* $\Gamma_a(h)$ for any $h \in H$).

Proof. 1. Lemma 9.1 provides a candidate, Lemma 10 shows that it works.

2. Let $h \in H$ and let ρ be one $\prec_a^h$-minimum of $\Gamma_a(h)$ by Lemma 9.2. Since Player 1 has no winning strategy in the threshold game for a and ρ in g^h, there exists a size-m strategy s_{-a}^H that makes Player 2 win. By Lemma 10 this strategy works also for $g^{h'}$ for all $h' \in H$.

□

Lemma 12 below already uses all the assumptions used in Theorem 5, but only for one given player.

Lemma 12. Let g be a game on a finite graph, and let some $\prec_a$ be an automatic-piecewise prefix-linear regular-Mont strict weak order with k pieces. Let $m \in \mathbb{N}$ and assume that the threshold games for a of the future games are determined *via* size-m strategies. There is a strategy s in g such that

$\mathrm{Reg} \cap \gamma_a(h, s^h) = \mathrm{Reg} \cap \Gamma_a(h)$ for all $h \in \mathscr{H}$, and that uses $m + 2\log k$ bits of memory, where Reg denotes the set of all regular runs $\rho \in [\mathscr{H}]$.

If $\prec_a$ is as above, but even fulfills the Mont condition, then we can ensure $\gamma_a(h, s^h) = \Gamma_a(h)$ for all $h \in \mathscr{H}$.

Proof. We define a strategic implementation for s in pseudocode in Algorithm 1. The algorithm uses in particular that by Lemma 11.1 for any piece $\overline{h}$ there is a strategic implementation using m bits for a strategy $s_a^{\overline{h}}$ such that $\gamma_a(h, s_a^{\overline{h}}) = \Gamma(h)$ for all $h \in \overline{h}$. An index of one of these strategic implementations is always stored, and the combined strategic implementation then follows the stored strategy as long as this one continues to realize the guarantee. If this is no longer the case, all of the k strategic implementations are checked, and one is chosen that does realize the guarantee, and this one is followed from there onwards.

Claim: The strategy implemented by the Algorithm 1 indeed satisfies the criteria.

Proof. Let $h_0 \in \mathscr{H}$. To show that $\gamma_a(h_0, s^{h_0}) = \Gamma_a(h_0)$ ($\mathrm{Reg} \cap \gamma_a(h_0, s^{h_0}) = \mathrm{Reg} \cap \Gamma_a(h_0)$), let $\rho \in \mathscr{H}(s^{h_0})$ ($\rho \in \mathrm{Reg} \cap \mathscr{H}(s^{h_0})$) and let us make a case distinction. First case, a changes strategies finitely many times along ρ. Let $h_1, \dots, h_n$ be such that for all $1 \le i \le n$ the i-th update along ρ occurs at history $h'_i := h_0\hat{\ }h_1\hat{\ }\dots\hat{\ }h_i$. Applying Lemma 8.3 n times yields $h_1\hat{\ }\dots\hat{\ }h_n\hat{\ }\gamma_a(h'_n, t_{h'_n}) \subseteq \dots \subseteq h_1\hat{\ }\gamma_a(h'_1, t_{h'_1}) \subseteq \gamma_a(h_0, t_{h_0}) \subseteq \Gamma_a(h_0)$. So $\rho \in \Gamma_a(h_0)$ since $\rho \in h_1\hat{\ }\dots\hat{\ }h_n\hat{\ }\gamma_a(h'_n, t_{h'_n})$.

Second case, a changes strategies infinitely many times along ρ. Let $(h_n)_{n\ge1}$ be the paths in (V,E) such that the n-th change occurring strictly after h_0 occurs at history $h'_n := h_0\hat{\ }h_1\hat{\ }\dots\hat{\ }h_n$. By Lemmas 9.2 and 10, for all $H \in \overline{\mathscr{H}}$ let ρ_a^H be a regular $\prec_a^H$-minimum of $\Gamma_a(H)$. By Lemma 8.2, for all $1 \le n$ there exists $\rho' \in \gamma_a(h'_{n+1}, t_{h'_n}^{h_{n+1}})$ such that $\rho' \prec_a^{h'_{n+1}} \rho''$ for all $\rho'' \in \gamma_a(h'_{n+1}, t_{h'_{n+1}})$, especially $\rho' \prec_a^{h'_{n+1}} \rho_a^{\overline{h'_{n+1}}}$. Since $h_{n+1}\rho' \in \gamma_a(h'_n, t_{h'_n})$ by Lemma 8.1, we find $\rho_a^{\overline{h'_n}} \prec_a^{h'_n} h_{n+1}\hat{\ }\rho_a^{\overline{h'_{n+1}}}$. By finiteness of $\overline{\mathscr{H}}$ one $H \in \overline{\mathscr{H}}$ occurs infinitely many times as a $\overline{h'_n}$. For all $n \ge 1$ let $h'_{\varphi(n)}$ be the n-th corresponding history.

If ρ is regular, there is some $f : \mathbb{N} \to \mathbb{N}$, finite path h' and cycle h such that $h_0\hat{\ }h'\hat{\ }h^{[n]} = h'_{\varphi(f(n))}$ (and thus $h'\hat{\ }h^{[\omega]} = \rho$). The inequality above can then be rewritten $h_0\hat{\ }h'\hat{\ }h^{[n]}\hat{\ }\rho_a^{\overline{h}} \prec_a h_0\hat{\ }h'\hat{\ }h^{[n+1]}\hat{\ }\rho_a^{\overline{h}}$ for all $n \ge 1$, so $h_0\hat{\ }h'\hat{\ }\rho_a^{\overline{h}} \prec_a h_0\hat{\ }\rho$ by the regular-Mont condition, so $h'\hat{\ }\rho_a^{\overline{h}} \prec_a^{h_0} \rho$.

In the general case, let $h = h'_{\varphi(1)}$, and let h' be such that $h = h_0\hat{\ }h'$. The inequality above can then be rewritten $h'_{\varphi(n)}\hat{\ }\rho_a^{\overline{h}} \prec_a h'_{\varphi(n+1)}\hat{\ }\rho_a^{\overline{h}}$ for all $n \ge 1$, so $h\hat{\ }\rho_a^{\overline{h}} \prec_a h_0\hat{\ }\rho$ by the Mont condition, so $h'\hat{\ }\rho_a^{\overline{h}} \prec_a^{h_0} \rho$.

Since $h'\hat{\ }\rho_a^{\overline{h}} \in \Gamma_a(h_0)$ by the finite case above, $\rho \in \Gamma_a(h_0)$. □

Let us now analyze the pseudo-code in Algorithm 1 and find out how much memory suffices. The algorithm keeps track of the current piece H of history so far, which by assumption requires $\log k$ bits. It also keeps track of an index of the current strategic implementation, which again requires $\log k$ bits (as we need at most one strategic implementation per piece). Finally, we use m bits for the memory content M.

□

Proof of Theorem 5. We combine Lemmas 9.1 and 12 to obtain a finite memory strategy s_a for each player a such that $\mathrm{Reg} \cap \gamma_a(h, s_a^h) = \mathrm{Reg} \cap \Gamma_a(h)$ for all $h \in \mathscr{H}$. The run ρ_{NE} induced by this strategy profile will be the run induced by the Nash equilibrium. As ρ_{NE} is induced by a finite-memory strategy profile, $\rho_{NE} \in \mathrm{Reg}$, and in particular, for any decomposition $\rho_{NE} = h\hat{\ }\rho$ and any player we find that $\rho \in \Gamma_a(h)$. Now we need to ensure that no one has any incentive to deviate.

```
   Data: Current local strategy implementation Strat
   current local memory content Mem
   current piece Piece
1  Function updatePiece is
       input  : A piece $H$ of history and a vertex $v \in V$
       output: The piece of history after $H$ and then $v$
2  end
3  Function updateLocal is
       input  : a strategy implementation s, a local memory content M, the vertex v the play is
                arriving in
       output: the updated local memory content M' and the vertex v' the strategy s wants to move
                to
4  end
5  Function realizesGuarantee is
       input  : a strategy implementation s, a local memory content M, the current piece H
       output: a boolean answer whether (s, M) is realizing the guarantee at H
6  end
7  Function Strategy is
       input  : vertex v the play is arriving in
       output: vertex v' the strategy wants to move to
8      Piece:= updatePiece (Piece, v);
9      (M', v') := updateLocal (Strat, Mem, v);
10     if realizesGuarantee (Strat, M', Piece) then
11         Mem:= M';
12         return v';
13     end if
14     else
15         foreach Strategy implementation s do
16             (M',v') := updateLocal (s, $0^m$, v);
17             if realizesGuarantee (s, M', Piece) then
18                 Strat:= s;
19                 Mem:= M';
20                 return v';
21             end if
22         end foreach
23     end if
24 end
```

Algorithm 1: Strategy for player *a* realizing guarantees

For all $a \in A$ and all pieces $H \in \overline{\mathscr{H}}$ of histories ending in V_a let s^H_{-a} be the strategy from Lemma 11.2. For all $a \in A$ let a play as follows: Keep playing according to ρ_{NE} until one player b deviates from ρ_{NE} at some history h_D. If $b = a$, *i.e.*, the last vertex of h_D is in V_a, let a do whatever; otherwise let a play according to $s^{\overline{h_D}}_{-b}$ in g^{h_D}, *i.e.* let a take part in the one-vs-all coalition that ensures that b cannot obtain anything $\prec^{h_D}_a$-better than $\rho_{\overline{h_D}}$ in g^{h_D}, so $\neg(\rho_{\overline{h_D}} \prec^{h_D}_b \rho_D)$, where ρ_D is the new run in g^{h_D} after deviation by b. Let $h_D\hat{}\rho = \rho_{NE}$. By construction of ρ_{NE} and Lemma 12, $\rho \in \Gamma_b(h_D)$, so $\neg(\rho \prec^{h_D}_b \rho_{\overline{h_D}})$. So $\neg(\rho_{NE} \prec_b h_D\hat{}\rho_D)$ since $\prec_b$ is a strict weak order, *i.e.* b has no incentive to perform a deviation.

Each player a needs to run the intended strategies s_b for all other players b, too, in order to be able to detect deviation. This requires $|A|(m + 2\log k)$ bits by Lemma 12. If a deviation is detected, the punishment strategy for that history is executed instead. (Note that the history where the deviation happened includes the information on who deviated; and that the players already keep track of the history in the strategy of Lemma 12.) The punishment phase requires $\log k$ memory to record which strategy to follow, and m bits to follow it. However, the original memory can be repurposed, by using just one extra bit to determine whether deviation ever occurred. This yields the overall memory bound $|A|(m + 2\log k) + 1$. □

4 Discussion

Comparison to previous work

As mentioned above, a similar but weaker result (compared to our Theorem 5) has previously been obtained by BRIHAYE, DE PRIL and SCHEWE [4],[20, Theorem 4.4.14]. They use cost functions rather than preference relations. Our setting of strict weak orders is strictly more general [2]. However, even if both frameworks are available, it is more convenient for us to have results formulated via preference relations rather than cost functions: Cost functions can be translated immediately into preferences, whereas translating preferences to cost functions is more cumbersome. In particular, it can be unclear to what extend *nice* preferences translate into *nice* cost functions. Note also that prefix-linearity for strict weak orders is more general than prefix-linearity for cost functions.

As a second substantial difference, [20, Theorem 4.4.14] requires either prefix-independent cost functions and finite memory determinacy of the induced games, or prefix-linear cost functions and positional subgame-perfect strategies. Their subgame-perfection assumption essentially means that they assume the result of Lemma 12. In particular,[20, Theorem 4.4.14] cannot be applied to energy parity games, where finite prefixes of the run do impact the overall value for the players, and where at least the protagonist requires memory to execute a winning strategy.

The Mont condition is absent in [4], but it can be shown that their other requirements imply a slight weakening of the Mont condition (which still suffices for our proof).

Before [20, 4], it had already been stated by PAUL and SIMON [19] that multi-player multi-outcome Muller games have Nash equilibria consisting of finite memory strategies. As (two-player win/lose) Muller games are finite memory determined [11], and the corresponding preferences are obviously prefix independent, this result is also a consequence of [20, Theorem 4.4.14]. Another result subsumed by [20, Theorem 4.4.14] (and subsequently our main theorem) is found in [3] by BRIHAYE, BRUYÈRE and DE PRIL.

[2]For example, the lexicographic combination of two payoff functions can typically not be modeled as a payoff function, as $\mathbb{R} \times \{0,1\}$ (with lexicographic order) does not embed into $\mathbb{R}$ as a linear order.

Algorithmic considerations

Let us briefly consider the algorithmic price to pay for the extension for the two-player win/lose case to the multi-player multi-outcome situation. Let us assume that for some class of games satisfying the criteria of Theorem 5, computing a winning strategy of a two-player win/lose game of size n has winning strategies takes $f(n)$ time. Let us further assume that, given finite memory strategies of size up to m bits for each player, we can decide in time $g(n,m)$ who is winning. Additionally, let us assume that a multi-player multi-outcome game of size n induces up to $t(n)$ one-vs-all threshold games of size n.

We can find for each $h \in \mathscr{H}$ some strategy s_a with memory size m such that $\gamma_a(h,s_a) = \Gamma_a(h)$ by brute force: We are investigating up to $t(n)$ induced one-vs-all threshold games, and are asking for a winning strategy in each of them, which could take up to $t(n)(f(n)+g(n,m))$ time. In any concrete example, though, a much more efficient construction is to be expected.

In Lemma 12, we need use the result above for each combination of memory content (2^m different values) and piece of the partition (k). Thus, we are spending up to $2^m kt(n)(f(n)+g(n,m))$ time to obtain sufficiently many strategies to realize the guarantee everywhere, which we can combine in linear time to yield the single strategy output strategy.

In the proof of Theorem 5, we invoke Lemma 12 once per player, which costs $O(|A|2^m kt(n)(f(n)+g(n,m))$ time. In addition, we need k many winning strategies in an induced one-vs-all threshold game, for additional cost of $O(kf(n))$ time – in total, we are still at $O(|A|2^m kt(n)(f(n)+g(n,m))$.

Let us introduce some realistic but simplifying additional assumptions. We would expect t to be of the form $2^{O(n)}$. The parameter k will be dominated by $2^{O(n)}$ in most situations. Typical finite-memory determined win/lose games might required exponential memory (if measured by number of states), but as we measure m in bits, also 2^m would be absorbed into $2^{O(n)}$. By dropping the distinctions between finding the winning strategy and determining who is winning, and noting that $|A| \leq n$, we arrive at an overall complexity of $2^{\mathcal{O}(n)}f(n)$. The additional factor of 2^n will in some cases worsen the asymptotic runtime significantly (e.g. for parity games, subexponential algorithms are known [12]), but in others, pales against the complexity for solving the two-player win/lose case.

On uniform finite memory determinacy

The requirement in Theorem 5 that there is a uniform memory bound sufficient for all threshold games is not dispensible. Mean-payoff parity games, for example, satisfy all other criteria, yet lack finite memory Nash equilibria, as the following example shows.

Let g be the one-player game in Figure 1b. The payoff of a run that visits the vertex g infinitely often is the limit (inferior or superior) of the average payoff. It is zero if g is visited finitely many times only. For any threshold $t \in \mathbb{R}$, if $t < 1$, the player has a winning finite-memory strategy: cycle p times in b, where $p > \frac{1}{1-t}$, visit g once, cycle p times in b, and so on. If $t \geq 1$, the player has no winning strategy at all. So the thresholds games of g, and likewise for the future game of g, are finite-memory determined. The game has no finite memory Nash equilibrium nonetheless, since the player can get a payoff as closed to 1 as she wants, but not 1.

Note that the preceding example could also be used for discounted-payoff parity games. However, these also fail the automatic-piece prefix-linearity condition.

Uniform finite memory determinacy can sometimes be recovered by considering ε-versions instead: We partition the payoffs in blocks of size ε, and let the player be indifferent within the same block. Clearly any Nash equilibrium from the ε-discretized version yields an ε-Nash equilibrium of the original game. If the original preferences were prefix-independent, the modified preferences still are. Moreover, as there are now only finitely many relevant threshold games per graph, their uniform finite memory

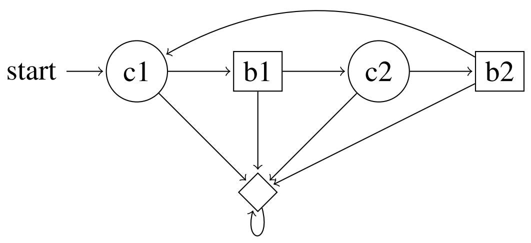


Figure 2: The graph for the game in Example 13

determinacy follows from mere finite memory determinacy. In such a situation, our result allows us to conclude that multi-player multi-outcome games have finite memory ε-Nash equilibria.

On the Mont condition

We can exhibit a prototypic example for how failure of the regular-Mont condition translates into the absence of Nash equilibria:

Example 13 ([3]). The game g in Figure 2 involves Player 1 (2) who owns the circle (box) vertices. Who owns the diamond is irrelevant. The payoff for Player 1 (2) is the number of visits to a box (circle) vertex, if this number is finite, and is -1 otherwise. Let s_1 be the positional strategy where Player 1 chooses $b1$ when in $c1$ and the diamond when in $c2$, and let t_2 be the positional strategy where Player 2 always chooses the diamond. With s_1 Player 1 secures payoff 1, and with t_2 Player 2 makes sure that Player 1 is not getting more than that. Let s_2 be any positional strategy for Player 2, and let t_1 be the positional strategy where Player 1 always choses the diamond. With s_2 Player 1 secures payoff 1, and with t_1 Player 1 makes sure that Player 2 is not getting more than that. Therefore the threshold games of g are positionally determined, and likewise for the future games of g. The game g has no Nash equilibrium nonetheless: in the run induced by a putative NE, one of the players has to choose the diamond at some point (to avoid payoff -1), but by postponing this choice to next time, the player can increase her payoff by 1. This shows the relevance of the Mont condition.

On automatic-piecewise prefix-linearity

While the definition of automatic-piecewise prefix-linearity may seem a bit complicated at first glance, the notion seems to fit in well with finite memory strategies: Intuitively, the requirement is merely that however we split a run into a finite prefix and an infinite tail, the contribution of the prefix to the value for the player factors via something expressible by a fixed (i.e. independent of the length of the prefix) number of bits, and does so via a finite automaton. Whenever this is not satisfied, one would expect that in principle a finite memory strategy may fail (compared to an unrestricted strategy), simply because it cannot properly account for the contribution of the finite prefix it has seen so far.

Of the popular winning conditions, many are actually prefix-independent, such as parity, Muller, mean-payoff, cost-Parity, cost-Streett [10], etc. Clearly, any combination of prefix-independent conditions itself will be prefix-independent. Typical examples of non-prefix independent conditions are reachability, energy, and discounted payoff. We can easily verify that combining a reachability or energy condition with any prefix-linear condition yields an automatic-piecewise prefix-linear condition (provided that energy is bounded).

[3]This example is based on an example communicated to the authors by Axel Haddad and Thomas Brihaye, which in turn is based on a construction in [5].

Discounted payoff can be problematic, though. Here each vertex is assigned a payoff a_v, a discount factor $\delta \in (0,1)$ is chosen, and the value of a run $\rho = v_0 v_1 \dots$ is $\sum_{i=0}^{\infty} v_i \delta^i$. While discounted payoff on its own is of course prefix-linear, it does not combine well with other criteria: For example, in a generalized discounted payoff parity game the question whether we prefer a tail with a better discounted payoff but worse least priority to a tail with worse discounted payoff but better least priority may depend on the precise value of the payoff obtained in the history so far, as well as the length of the history (as later contributions to payoff count less). This is too much information for a finite automaton to remember, thus, generalized discounted payoff parity games do not satisfy the criterion for being automatic-piecewise prefix-linear.

Applications

We shall briefly mention two classes of games covered by our main theorem, but not by the results from [20, 4], (bounded) energy parity games and a variant of reachability + mean-payoff games. For more details, we refer to [17].

Energy games were first introduced in [7]: Two players take turns moving a token through a graph, while keeping track of the *current energy level*, which will be some integer. Each move either adds or subtracts to the energy level, and if the energy level ever reaches 0, the protagonist loses. These conditions were later combined with parity winning conditions in [8] to yield energy parity games as a model for a system specification that keeps track of gaining and spending of some resource, while simultaneously conforming to a parity specification.

In both [7] and [8] the energy levels are a priori not bounded from above. This is a problem for the applicability of Theorem 5, as unbounded energy parity preferences are not automatic-piecewise prefix-linear in the general case. In [2], two versions of bounded energy conditions were investigated: Either any energy gained in excess of the upper bound is just lost (as in e.g. recharging a battery), or gaining energy in excess of the bound leads to a loss of the protagonist (as in e.g. refilling a fuel tank without automatic spill-over prevention). We are only concerned with the former. As a finite automaton can easily keep track of the energy level between 0 and the upper bound, bounded energy parity preferences are automatic-piecewise prefix linear. We can also show that these games are uniformly finite-memory determined, and with some extra work obtain (see [17]):

Corollary 14. All multiplayer multioutcome energy parity games have Nash equilibria in finite memory strategies. Let A be the set of players, n the size of the graph and W the largest energy delta. Further let E be the maximum difference between the energy maximum and the energy minimum for some player. Then $1 + |A| n E^{|A|} \log(2nW) + (|A|^2 + |A|) \log nE$ bits of memory suffice.

In our second example, each player has both a mean-payoff goal and a reachability objective. Maximizing the mean-payoff, however, takes precedence, and the reachability objective only becomes relevant as a tie-breaker. These conditions are not expressible via some payoff or cost function[4], but are still automatic-piecewise prefix-independent strict weak orders. As the two-player win/lose games can easily be reduced to mean-payoff games, we obtain uniform finite-memory determinacy, and thus the existence of finite-memory Nash equilibria by Theorem 5.

We leave the investigation whether winning conditions defined via LTL$[\mathscr{F}]$ or LTL$[\mathscr{D}]$ formulae [1] match the criteria of Theorem 5 to future work. Another area of prospective examples to explore are multi-dimensional objectives as studied e.g. in [21, 9].

[4]Compare the footnote on Page 4.

References

[1] S. Almagor, U. Boker & O Kupferman (2016): *Formalizing and reasoning about quality*. Journal of the ACM, doi:10.1145/2875421.

[2] Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, Nicolas Markey & Jiří Srba (2008): *Infinite Runs in Weighted Timed Automata with Energy Constraints*. In Franck Cassez & Claude Jard, editors: Formal Modeling and Analysis of Timed Systems, Lecture Notes in Computer Science 5215, Springer Berlin Heidelberg, pp. 33–47, doi:10.1007/978-3-540-85778-5_4.

[3] Thomas Brihaye, Veroniqué Bruyère & Julie De Pril (2010): *Equilibria in quantitative reachability games*. In: Proc. of CSR, LNCS 6072, Springer.

[4] Thomas Brihaye, Julie De Pril & Sven Schewe (2013): *Multiplayer Cost Games with Simple Nash Equilibria*. In: Logical Foundations of Computer Science, LNCS, pp. 59–73, doi:10.1007/978-3-642-35722-0_5.

[5] Thomas Brihaye, Gilles Geeraerts, Axel Haddad & Benjamin Monmege (2014): *To Reach or not to Reach? Efficient Algorithms for Total-Payoff Games*. arXiv 1407.5030.

[6] Nils Bulling & Valentin Goranko (2013): *How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (Extended Abstract)*. In: Proc. of Strategic Reasoning, doi:10.4204/EPTCS.112.8. Available at `http://www.arxiv.org/abs/1303.0789`.

[7] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger & Mariëlle Stoelinga (2003): *Resource Interfaces*. In Rajeev Alur & Insup Lee, editors: Embedded Software, Lecture Notes in Computer Science 2855, Springer Berlin Heidelberg, pp. 117–133, doi:10.1007/978-3-540-45212-6_9.

[8] Krishnendu Chatterjee & Laurent Doyen (2012): *Energy parity games*. Theor. Comput. Sci. 458, pp. 49–60, doi:10.1016/j.tcs.2012.07.038.

[9] Lorenzo Clemente & Jean-François Raskin (2015): *Multidimensional beyond Worst-Case and Almost-Sure Problems for Mean-Payoff Objectives*. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015, pp. 257–268, doi:10.1109/LICS.2015.33.

[10] Nathanaël Fijalkow & Martin Zimmermann (2014): *Parity and Streett Games with Costs*. Logical Methods in Computer Science 10(2), doi:10.2168/LMCS-10(2:14)2014.

[11] Yuri Gurevich & L. Harrington (1982): *Trees, automata and games*. In: Proc. STOC, doi:10.1145/800070.802177.

[12] Marcin Jurdzinski, Mike Paterson & Uri Zwick (2008): *A Deterministic Subexponential Algorithm for Solving Parity Games*. SIAM J. Comput. 38(4), pp. 1519–1532, doi:10.1137/070686652.

[13] Orna Kupferman (2016): *On High-Quality Synthesis*. In S. Alexander Kulikov & J. Gerhard Woeginger, editors: 11th International Computer Science Symposium in Russia, CSR 2016, Springer International Publishing, pp. 1–15, doi:10.1007/978-3-319-34171-2_1.

[14] Stéphane Le Roux (2013): *Infinite Sequential Nash Equilibria*. Logical Methods in Computer Science 9(2), doi:10.2168/LMCS-9(2:3)2013.

[15] Stéphane Le Roux & Arno Pauly (2014): *Infinite Sequential Games with Real-valued Payoffs*. In: CSL-LICS '14, ACM, pp. 62:1–62:10, doi:10.1145/2603088.2603120.

[16] Stéphane Le Roux & Arno Pauly (2015): *Weihrauch Degrees of Finding Equilibria in Sequential Games*. In Arnold Beckmann, Victor Mitrana & Mariya Soskova, editors: Evolving Computability, Lecture Notes in Computer Science 9136, Springer, pp. 246–257, doi:10.1007/978-3-319-20028-6_25.

[17] Stéphane Le Roux & Arno Pauly (2016): *Extending finite memory determinacy: General techniques and an application to energy parity games*. arXiv:1602.08912.

[18] Jean François Mertens (1987): *Repeated Games*. In: Proc. Internat. Congress Mathematicians, American Mathematical Society, pp. 1528–1577.

[19] Soumya Paul & Sunil Simon (2009): *Nash Equilibrium in Generalised Muller Games*. In Ravi Kannan & K. Narayan Kumar, editors: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, *Leibniz International Proceedings in Informatics (LIPIcs)* 4, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 335–346, doi:10.4230/LIPIcs.FSTTCS.2009.2330. Available at `http://drops.dagstuhl.de/opus/volltexte/2009/2330`.

[20] Julie De Pril (2013): *Equilibria in Multiplayer Cost Games*. Ph.D. thesis, Université de Mons.

[21] Yaron Velner, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, Alexander Rabinovich & Jean-François Raskin (2015): *The complexity of multi-mean-payoff and multi-energy games*. *Information and Computation* 241, pp. 177 – 196, doi:10.1016/j.ic.2015.03.001. Available at `http://www.sciencedirect.com/science/article/pii/S0890540115000164`.