Rational Behaviour and Strategy Construction
in Infinite Multiplayer Games*

Michael Ummels

Mathematische Grundlagen der Informatik, RWTH Aachen, Germany
E-Mail: ummels@logic.rwth-aachen.de

ABSTRACT.   *We study infinite games played by arbitrarily many players on a directed graph. Equilibrium states capture rational behaviour in these games. Instead of the well-known notion of a Nash equilibrium, we focus on the notion of a subgame perfect equilibrium. We argue that the latter one is more appropriate for the kind of games we study, and we show the existence of a subgame perfect equilibrium in any infinite game with ω-regular winning conditions.*

As, in general, equilibria are not unique, it is appealing to compute one with a maximal payoff. This problem corresponds naturally to the problem of deciding given a game and two payoff vectors whether the game has an equilibrium with a payoff in between the given thresholds. We show that this problem is decidable for games with ω-regular winning conditions played on a finite graph and analyse its complexity. Moreover, we establish that any subgame perfect equilibrium of a game with ω-regular winning conditions played on a finite graph can be implemented by finite-state strategies.

Finally, we consider logical definability. We state that if we fix the number of players together with an ω-regular winning condition for each of them and two payoff vectors the property that a game has a subgame perfect equilibrium with a payoff in between the given thresholds is definable in the modal μ-calculus.

1 Introduction

We study *infinite games of perfect information* [7] played by multiple players on a directed graph. Intuitively, a *play* of such a game evolves by moving a token along edges of the graph. Every vertex of the graph is controlled by precisely one player. Whenever the token arrives at some vertex, the player who controls this vertex must move the token to a successor vertex. Thus a play of such a game is an infinite path through the graph. Plays are mapped to *payoffs*, one for each player. In the simplest case, which we discuss here, payoffs are just 0 and 1, i.e. each player either wins or loses a given play of

---

the game. We allow, however, that a play is won by more than one player or even by no player at all.

Infinite games have been successfully applied in the verification and synthesis of reactive systems. Such a system is usually modelled as a game between the system and its environment where the environment's objective is the complement of the system's objective, so the environment is considered hostile. Therefore, traditionally, the research in this area has mostly looked at two-player games where each play is won by precisely one of the two players, so-called *two-player zero-sum games*. However, motivated by the modelling of distributed systems, interest in the general case has increased in recent years [2, 3].

Example 1. Consider a scenario where three agents are competing for a resource that can only be used by at most two of them using the following protocol: At first, agent 1 decides whether to grant the other agents 2 and 3 access to the resource or to pass control to agent 2. If control is passed to agent 2, she can decide whether to share access to the resource with agent 1 or to grant agent 3 exclusive access to the resource. The situation is naturally modelled by the following game with its arena depicted in Figure 1; round vertices are controlled by player 1; boxed vertices are controlled by player 2; player 3 does not control any vertex; player 1 wins if vertex 5 is visited (infinitely often); player 2 wins if vertex 4 or vertex 5 is visited (infinitely often); player 3 wins if vertex 3 or vertex 4 is visited (infinitely often); the initial vertex is 1.
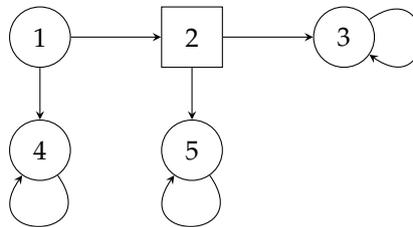

Figure 1. A game with three players.

Different *solution concepts* [17] have been proposed to model rational behaviour in games. The classical solution concept offered by game theory is the one of a *Nash equilibrium* [16]. In a Nash equilibrium no player can receive a better payoff by unilaterally changing her strategy. For instance, the game described in Example 1 has two Nash equilibrium payoffs:

1. Players 1 and 2 win; a Nash equilibrium with this payoff is the combination of strategies where player 1 moves from vertex 1 to vertex 2 and player 2 moves from vertex 2 to vertex 5.
2. Players 2 and 3 win; a Nash equilibrium with this payoff is the combi-

nation of strategies where player 1 moves from vertex 1 to vertex 4 and player 2 moves from vertex 2 to vertex 3.

Intuitively, the second equilibrium is not rational because if player 1 moved from vertex 1 to vertex 2 instead player 2 should change her strategy and move to vertex 5 instead because this is then the only way for her to win. However, in the definition of a Nash equilibrium it is not taken into account that players can change their strategies during a play.

An equilibrium concept that respects this possibility is the notion of a subgame perfect equilibrium [20]. For being a subgame perfect equilibrium, a choice of strategies is not only required to be optimal for the initial vertex but for every possible initial history of the game (including histories not reachable in the equilibrium play). In the example the second Nash equilibrium is not a subgame perfect equilibrium because moving from vertex 2 to vertex 3 is not optimal for player 2 after the play has reached vertex 2.

Subgame perfect equilibria have been well studied in the context of *finite games*. In particular, Kuhn [10] showed that every finite game has a subgame perfect equilibrium. Yet, we think that the concept is also worth to be analysed in the context of infinite games because the possibility of changing strategies during a play is not unique to finite games. In this paper we show the existence of subgame perfect equilibria for infinite games with parity winning conditions, a standard form of ω-regular winning conditions, and we remark that the same holds for the greater class of Borel objectives. This generalises a result by Chatterjee et al. [3] about the existence of Nash equilibria in infinite games. Based on the proof, we also develop an algorithm for computing a subgame perfect equilibrium of a game with parity winning conditions.

We then turn to the potentially harder problem of finding a subgame perfect equilibrium with a maximal payoff. This problem is closely related to the problem of deciding given a game and two payoff vectors whether the game has a subgame perfect equilibrium with a payoff in between the given thresholds. Using a translation into tree automata, we show that the latter problem is decidable for games with ω-regular winning conditions played on a finite graph. In particular, we show that for games with Rabin objectives the problem is decidable in exponential time in general and in polynomial time if the number of players and the number of Rabin pairs are bounded. Moreover, we show that the problem is 2ExpTime-complete for games with LTL objectives.

Naturally, we are also interested in the complexity of strategies realising an equilibrium. We show that for games with ω-regular winning conditions played on a finite graph any subgame perfect equilibrium can be implemented by finite-state strategies. This is the best one can hope for because, even for games with Büchi objectives, positional strategies, in general, do not suffice to implement any Nash or subgame perfect equilibrium.

We conclude this paper with a section on logical definability. It is well

known that for any fixed number m the property that the first player wins a two-player zero-sum parity game with m different priorities is definable in the modal μ-calculus. We state a natural generalisation of this fact for multiplayer games: If we fix the number of players together with an ω-regular winning condition for each of them and two payoff vectors, the property that a game has a subgame perfect equilibrium with a payoff in between the given thresholds is definable in the modal μ-calculus as well.

2 Infinite Multiplayer Games

The definition of an infinite (two-player zero-sum) game played on a directed, coloured graph [24] easily generalises to the multiplayer setting. Formally, we define an *infinite multiplayer game* as a tuple $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (W_i)_{i \in \Pi})$ where

- Π is a finite set of *players*;
- (V, E) is a (not necessarily finite) directed graph;
- $(V_i)_{i \in \Pi}$ is a partition of V;
- $\chi : V \to C$ for some set C;
- $W_i \subseteq C^\omega$ for all $i \in \Pi$.

The structure $G = (V, (V_i)_{i \in \Pi}, E, \chi)$ is called the *arena* of $\mathcal{G}$; χ is called the *colouring* of G, and W_i is called the *winning condition* of player $i \in \Pi$. For the sake of simplicity, we assume that $uE := \{v \in V : (u, v) \in E\} \neq \varnothing$ for all $u \in V$, i.e. each vertex of G has at least one outgoing edge. We say that $\mathcal{G}$ is *finitely coloured* if $\chi : V \to C$ for a finite set C. Finally, we call $\mathcal{G}$ a *zero-sum game* if the sets W_i define a partition of V^ω. Thus if $\mathcal{G}$ is an infinite two-player zero-sum game with players 0 and 1 it suffices to define V_0 and W_0, and we just write $\mathcal{G} = (V, V_0, E, \chi, W_0)$.

A *play* or *history* of $\mathcal{G}$ is an infinite or finite path in G, respectively. We say that a play π is *won* by player $i \in \Pi$ if $\chi(\pi) \in W_i$. The *payoff* of a play π of $\mathcal{G}$ is the vector $\mu(\pi) \in \{0, 1\}^\Pi$ defined by $\mu(\pi)_i = 1$ if π is won by player i. A *strategy of player i in $\mathcal{G}$* is a total function $\sigma : V^*V_i \to V$ assigning to each nonempty sequence wv of vertices ending in a vertex v of player i another vertex $\sigma(wv)$ such that $(v, \sigma(wv)) \in E$. We say that a play π of $\mathcal{G}$ is *consistent* with a strategy σ of player i if $\pi(k+1) = \sigma(\pi(0) \dots \pi(k))$ for all $k < \omega$ with $\pi(k) \in V_i$. A *strategy profile of $\mathcal{G}$* is a tuple $(\sigma_i)_{i \in \Pi}$ where σ_i is a strategy of player i in $\mathcal{G}$.

A strategy σ of player i in $\mathcal{G}$ is called *positional* if σ depends only on the current vertex, i.e. if $\sigma(wv) = \sigma(v)$ for all $w \in V^*$ and $v \in V_i$. More generally, σ is called a *finite-state strategy* if the equivalence relation $\sim_\sigma$ on V^* defined by $w \sim_\sigma w'$ if $\sigma(wz) = \sigma(w'z)$ for all $z \in V^*V_i$ has finite index. In other words, a finite-state strategy is a strategy that can be implemented by a finite automaton with output. A strategy profile $(\sigma_i)_{i \in \Pi}$ of $\mathcal{G}$ is called *positional*

or a *finite-state strategy profile* if each σ_i is positional or a finite-state strategy, respectively.

It is sometimes convenient to designate an initial vertex $v_0 \in V$ of the game. We call the tuple $(\mathcal{G}, v_0)$ an *initialised infinite multiplayer game*. A *play (history) of* $(\mathcal{G}, v_0)$ is a play (history) of $\mathcal{G}$ starting with v_0. A strategy (strategy profile) of $(\mathcal{G}, v_0)$ is just a strategy (strategy profile) of $\mathcal{G}$. A strategy σ of some player i in $(\mathcal{G}, v_0)$ is *winning* if every play of $(\mathcal{G}, v_0)$ consistent with σ is won by player i. A strategy profile $(\sigma_i)_{i \in \Pi}$ of $(\mathcal{G}, v_0)$ determines a unique play of $(\mathcal{G}, v_0)$ consistent with each σ_i, called the *outcome of* $(\sigma_i)_{i \in \Pi}$ and denoted by $\langle (\sigma_i)_{i \in \Pi} \rangle$ or, in the case that the initial vertex is not understood from the context, $\langle (\sigma_i)_{i \in \Pi} \rangle_{v_0}$. In the following we will often use the term *game* to denote an *(initialised) infinite multiplayer game*.

For a game $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (W_i)_{i \in \Pi})$ and a history h of $\mathcal{G}$, let the game $\mathcal{G}|_h = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (W_i|_h)_{i \in \Pi})$ be defined by $W_i|_h = \{\alpha \in C^\omega : \chi(h) \cdot \alpha \in W_i\}$. For an initialised game $(\mathcal{G}, v_0)$ and a history hv of $(\mathcal{G}, v_0)$, we call the initialised game $(\mathcal{G}|_h, v)$ the *subgame of* $(\mathcal{G}, v_0)$ *with history hv*. For a strategy σ of player $i \in \Pi$ in $\mathcal{G}$, let $\sigma|_h : V^* V_i \to V$ be defined by $\sigma|_h(wv) = \sigma(hwv)$. Obviously, $\sigma|_h$ is a strategy of player i in $\mathcal{G}|_h$.

A strategy profile $(\sigma_i)_{i \in \Pi}$ of a game $(\mathcal{G}, v_0)$ is called a *Nash equilibrium* if for any player $i \in \Pi$ and all her possible strategies σ' in $(\mathcal{G}, v_0)$ the play $\langle \sigma', (\sigma_j)_{j \in \Pi \setminus \{i\}} \rangle$ is won by player i only if the play $\langle (\sigma_j)_{j \in \Pi} \rangle$ is also won by her. The strategy profile $(\sigma_i)_{i \in \Pi}$ is called a *subgame perfect equilibrium (SPE)* if $(\sigma_i|_h)_{i \in \Pi}$ is a Nash equilibrium of $(\mathcal{G}|_h, v)$ for every history hv of $(\mathcal{G}, v_0)$.

Winning conditions

We have introduced winning conditions as abstract sets of infinite sequences over the set of colours. In verification winning conditions are usually *ω-regular sets* specified by formulae of the logic S1S (monadic second-order logic on infinite words) or LTL (linear-time temporal logic) referring to unary predicates P_c indexed by the set C of colours, which is assumed to be finite. Special cases are the following well-studied winning conditions:

- *Büchi* (given by $F \subseteq C$): defines the set of all $\alpha \in C^\omega$ such that $\alpha(k) \in F$ for infinitely many $k < \omega$.
- *Parity* (given by a *priority function* $\Omega : C \to \omega$): defines the set of all $\alpha \in C^\omega$ such that the least number occurring infinitely often in $\Omega(\alpha)$ is even.
- *Rabin* (given by a set Ω of *pairs* (G_i, R_i) where $G_i, R_i \subseteq C$): defines the set of all $\alpha \in C^\omega$ such that there exists an index i such that $\alpha(k) \in G_i$ for infinitely many $k < \omega$ but $\alpha(k) \in R_i$ only for finitely many $k < \omega$.

Note that the Büchi condition is a special case of the parity condition with two priorities and that the parity condition is a special case of the Rabin condition. Also note that Büchi, parity and Rabin conditions are *prefix independent*, i.e.

for every $\alpha \in C^\omega$ and $w \in C^*$ it is the case that α satisfies the condition if and only if $w\alpha$ does.

We call a finitely coloured game $\mathcal{G}$ a *multiplayer S1S, LTL, Büchi, parity or Rabin game* if the winning condition of each player is of type S1S, LTL, Büchi, parity or Rabin, respectively.[1] Any of these games is called an *ω-regular game*. It is well known that the complement of a Rabin condition is again expressible as a Rabin condition if and only if it is equivalent to a parity condition. Thus any two-player zero-sum Rabin game is also a two-player zero-sum parity game. Observe that $\mathcal{G}|_h = \mathcal{G}$ for every history h of $\mathcal{G}$ if $\mathcal{G}$ is a multiplayer Büchi, parity or Rabin game because the winning conditions in these games are prefix independent.

We say that two initialised games $(\mathcal{G}, v_0)$ and $(\mathcal{G}', v_0')$ are *equivalent* if for any (finite-state) Nash or subgame perfect equilibrium of $(\mathcal{G}, v_0)$ there exists a (finite-state) Nash or subgame perfect equilibrium of $(\mathcal{G}', v_0')$, respectively, with the same payoff and, vice versa, for any (finite-state) Nash or subgame perfect equilibrium of $(\mathcal{G}', v_0')$ there exists a (finite-state) Nash or subgame perfect equilibrium of $(\mathcal{G}, v_0)$, respectively, with the same payoff. As for two-player zero-sum games (see, for example, [21]), any ω-regular multiplayer game can be reduced to an equivalent multiplayer parity game.

Proposition 2. Any ω-regular multiplayer game $(\mathcal{G}, v_0)$ is equivalent to a multiplayer parity game $(\mathcal{G}', v_0')$. If $\mathcal{G}$ is a multiplayer LTL game with k players, n vertices and winning conditions of size $\leq m$, then $\mathcal{G}'$ has $n \cdot 2^{2^{O(m)+\log k}}$ vertices and $2^{O(m)}$ priorities for each player.

3 Existence of Subgame Perfect Equilibria

The aim of this section is to show that any ω-regular multiplayer game has a subgame perfect equilibrium. By Proposition 2, it suffices to consider multiplayer parity games. In the case of two-player zero-sum games, parity games are *positionally determined*, i.e. one of the two players not only has a winning strategy but a positional one.

Theorem 3 (Emerson-Jutla [4], Mostowski [14])**.** Two-player zero-sum parity games are positionally determined.

Moreover, positional winning strategies can always be chosen uniformly, i.e. independently of the initial vertex (see, for example, [24]). Hence any two-player zero-sum parity game has a positional subgame perfect equilibrium.

Corollary 4. Any two-player zero-sum parity game has a positional subgame perfect equilibrium.

---

[1]Our notation differs here from the usual notation for two-player zero-sum games where a Büchi or Rabin game is a game where the winning condition of the first player is a Büchi or Rabin condition, respectively.

Using Martin's determinacy theorem for two-player zero-sum Borel games [12], Chatterjee et al. [3] showed that any multiplayer game with Borel winning conditions has a Nash equilibrium. Rephrased for parity games, roughly speaking, their proof goes as follows: Given a multiplayer parity game $(\mathcal{G}, v_0)$, for each player i, consider the two-player zero-sum parity game $(\mathcal{G}_i, v_0)$ where player i plays against the coalition of all other players. By Corollary 4, this game has a subgame perfect equilibrium consisting of a strategy σ_i of player i and a strategy σ_{-i} of the coalition, i.e. for every vertex v of $\mathcal{G}$ either σ_i or σ_{-i} is winning in $(\mathcal{G}_i, v)$. In the equilibrium player i plays her strategy σ_i as long as no other player j deviates from her strategy σ_j in which case she switches to the coalition strategy σ_{-j}. In game theory this type of strategy is known under the term *threat strategy* and has its origin in the theory of repeated games (cf. [17, Chapter 8]). To make the Nash equilibrium a subgame perfect equilibrium, we do not consider threat strategies in the original game but in a game arising as a fixed point of a deflationary operator defined on the original game.

Theorem 5. Any multiplayer parity game has a subgame perfect equilibrium.

Proof. Let $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (\Omega_i)_{i \in \Pi})$ be a multiplayer parity game. For each ordinal α we define a set $E^\alpha \subseteq E$ beginning with $E^0 = E$ and

$$E^\lambda = \bigcap_{\alpha < \lambda} E^\alpha$$

for limit ordinals λ. To define $E^{\alpha+1}$ from E^α, we consider for each player $i \in \Pi$ the two-player zero-sum parity game $\mathcal{G}_i^\alpha = (V, V_i, E^\alpha, \chi, \Omega_i)$ where player i plays against the coalition of all other players. By Corollary 4, we can fix a positional subgame perfect equilibrium $(\sigma_i^\alpha, \sigma_{-i}^\alpha)$ of this game where σ_i^α is a strategy of player i and σ_{-i}^α is a strategy of the coalition. Let X_i^α be the set of all $v \in V$ such that σ_i^α is winning in $(\mathcal{G}_i^\alpha, v)$. For vertices $v \in V_i \cap X_i^\alpha$ we delete all outgoing edges except the one taken by the strategy σ_i^α, i.e. we define

$$E^{\alpha+1} = E^\alpha \setminus \bigcup_{i \in \Pi} \{(u,v) \in E : u \in V_i \cap X_i^\alpha \text{ and } v \neq \sigma_i^\alpha(u)\}.$$

Obviously, the sequence $(E^\alpha)_{\alpha \in \mathrm{On}}$ is nonincreasing. Thus we can fix the least ordinal ξ with $E^\xi = E^{\xi+1}$ and define $\sigma_i = \sigma_i^\xi$ and $\sigma_{-i} = \sigma_{-i}^\xi$. Moreover, for each player $j \neq i$ let $\sigma_{j,i}$ be the positional strategy of player j in $\mathcal{G}$ that is induced by σ_{-i}. Player i's equilibrium strategy τ_i is defined as follows: Player i plays σ_i as long as no other player deviates. Whenever some player $j \neq i$ deviates from her equilibrium strategy τ_j, player i switches to $\sigma_{i,j}$. Then $(\tau_i)_{i \in \Pi}$ is a subgame perfect equilibrium of $(\mathcal{G}, v_0)$ for any initial vertex $v_0 \in V$.   Q.E.D.

More generally, Theorem 5 holds for games with (quasi-)Borel winning conditions [13]. The proof is similar to the proof for parity games but based

on Martin's determinacy theorem for (quasi-)Borel sets [12, 13]. However, Martin's theorem can only guarantee the existence of an arbitrary subgame perfect equilibrium in a two-player zero-sum game with (quasi-)Borel winning conditions, not necessarily a positional one. To ensure that the proof works, we have to assume that the arena of the game under consideration is a forest. Over a forest any strategy is obviously equivalent to a positional one. The justification for this assumption is that we can always replace the arena of an arbitrary game by its unravelling from the initial vertex, ending up in an equivalent game. See [23, Chapter 3] for the full proof.

Theorem 6. Any multiplayer game with (quasi-)Borel winning conditions has a subgame perfect equilibrium.

Naturally, we are interested in the complexity of strategies realising a subgame perfect equilibrium. It is easy to see that for parity games played on a finite arena the subgame perfect equilibrium constructed in the proof of Theorem 5 is, in fact, a finite-state one. This leaves open the existence of a positional subgame perfect equilibrium as it is guaranteed in the two-player zero-sum case (even for games with an infinite arena). We are only able to give a partial answer to this question, namely in the case of only two players 1 and 2. Indeed, it is easy to see that the positional strategies $\sigma_{1,2}$ and $\sigma_{2,1}$ as defined in the proof of Theorem 5 form a subgame perfect equilibrium in this case.

Theorem 7. Any two-player parity game has a positional subgame perfect equilibrium.

A Simple Algorithm

Knowing that there always exists a subgame perfect equilibrium in an ω-regular multiplayer game, the next challenge is to compute one. Algorithm 1 is a simple procedure for computing a subgame perfect equilibrium of a multiplayer parity game derived from the proof of Theorem 5 in a straightforward way. Thus its correctness follows immediately.

Obviously, the running time of the algorithm depends on the running time of the algorithm we use for computing a positional subgame perfect equilibrium of a two-player zero-sum parity game. The best known algorithm for this problem, which also computes the winning regions (i.e. the set of vertices from which each player has a winning strategy) of the game, is due to Jurdziński [9]. For a game with at most n vertices, m edges and $d \geq 2$ different priorities, Jurdziński's algorithm runs in time

$$\mathrm{O}\left(dm\left(\frac{n}{\lfloor d/2\rfloor}\right)^{\lfloor d/2\rfloor}\right).$$

Note that each strategy τ_i can be implemented by a finite automaton of size $\mathrm{O}(|\Pi|^2|V|)$. Thus we have the following theorem.

8

Algorithm 1. Computing a finite-state SPE of a multiplayer parity game.

---

input multiplayer parity game $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (\Omega_i)_{i \in \Pi})$

$E_{\text{new}} := E$

repeat

   $E_{\text{old}} := E_{\text{new}}$

  for each $i \in \Pi$ **do**

     Compute a positional SPE (σ_i, σ_{-i}) of $\mathcal{G}_i = (V, V_i, E_{\text{old}}, \chi, \Omega_i)$

     $W_i := \{v \in V : \sigma_i \text{ is winning in } (\mathcal{G}_i, v)\}$

     $E_{\text{new}} := E_{\text{new}} \setminus \{(u, v) \in E : u \in V_i \cap W_i \text{ and } v \neq \sigma_i(u)\}$

  end for

until $E_{\text{new}} = E_{\text{old}}$

for each $i \in \Pi$ **do**

  Compute equilibrium strategy τ_i of player i

end for

output $(\tau_i)_{i \in \Pi}$

Theorem 8. Computing a finite-state subgame perfect equilibrium of a multiplayer parity game with k players, n vertices, m edges and at most $d \geq 2$ priorities for each player can be done in time

$$\mathrm{O}\left(kdm^2 \left(\frac{n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor} + k^3 n^2\right).$$

In particular, Theorem 8 says that we can compute a subgame perfect equilibrium of a multiplayer parity game in polynomial time for classes of games with a bounded number of priorities. Moreover, if there exists a polynomial-time algorithm for computing a positional subgame perfect equilibrium of a two-player zero-sum parity game then Algorithm 1 can be made to run in polynomial time as well. Hence the problem of computing a subgame perfect equilibrium in an arbitrary multiplayer parity game is computationally not much harder than the corresponding problem for two-player zero-sum parity games.

4 Complexity

One can easily construct games where Algorithm 1 computes an equilibrium with a payoff of $(0, \ldots, 0)$ although there is an equilibrium with a payoff of $(1, \ldots, 1)$. This is unsatisfactory because, if we think of verification, we want as many components as possible to fulfil their specification. Therefore it seems desirable to find an equilibrium with a maximal payoff (a *maximal subgame perfect equilibrium*). This *maximisation problem* naturally corresponds to the decision problem SPE defined as follows:[2]

---

[2] Here, $\leq$ denotes the product ordering on $\{0,1\}^k$, i.e. $x \leq y$ if $x_i \leq y_i$ for all i.

Given an ω-regular multiplayer game $(\mathcal{G}, v_0)$ played on a finite arena and thresholds $x, y \in \{0,1\}^k$, decide whether $(\mathcal{G}, v_0)$ has a subgame perfect equilibrium with a payoff $\geq x$ and $\leq y$.

Note that we can find the payoff of a maximal subgame perfect equilibrium with k queries to the decision problem if k is the number of players. To solve the problem SPE, we use a reduction to the problem of deciding whether a given tree automaton defines a nonempty tree language.

Theorem 9. The problem of deciding given a multiplayer Rabin game $(\mathcal{G}, v_0)$ played on a finite arena and thresholds $x, y \in \{0,1\}^k$ whether $(\mathcal{G}, v_0)$ has a subgame perfect equilibrium with a payoff $\geq x$ and $\leq y$ is in EXPTIME. If the number of players and pairs is bounded, the problem is in P.

Proof sketch. Without loss of generality, we can assume that $\mathcal{G}$ is *binary*, i.e. every vertex of $\mathcal{G}$ has at most two successors. Then we can arrange all plays of $(\mathcal{G}, v_0)$ in an infinite binary tree with labels from the vertex set V. Given a strategy profile $(\sigma_i)_{i \in \Pi}$ of $(\mathcal{G}, v_0)$, we enrich this tree with a second label component that takes the value 0 or 1 if the strategy profile prescribes going to the left or right successor, respectively.

The algorithm works as follows: We construct two *alternating parity tree automata*. The first one checks whether some arbitrary tree with labels from the alphabet $V \times \{0,1\}$ is indeed a tree originating from a strategy profile of $(\mathcal{G}, v_0)$, and the second one checks for a tree originating from a strategy profile $(\sigma_i)_{i \in \Pi}$ of $(\mathcal{G}, v_0)$ whether $(\sigma_i)_{i \in \Pi}$ is a subgame perfect equilibrium with a payoff in between the given thresholds. The first automaton is actually a nondeterministic tree automaton with trivial acceptance (every run of the automaton is accepting) and has $O(|V|)$ states. The second automaton has $O(kd)$ states and $O(1)$ priorities where k is the number of players and d is the maximum number of pairs in a player's winning condition. An equivalent nondeterministic parity tree automaton has $2^{O(kd \log kd)}$ states and $O(kd)$ priorities [15]. Finally, we construct the product automaton of the first nondeterministic parity tree automaton with the one constructed from the alternating one. As the former automaton works with trivial acceptance, the construction is straightforward and leads to a nondeterministic parity tree automaton with $O(|V|) \cdot 2^{O(kd \log kd)}$ states and $O(kd)$ priorities. Obviously, the tree language defined by this automaton is nonempty if and only if $(\mathcal{G}, v_0)$ has a subgame perfect equilibrium with a payoff in between the given thresholds. By [5], nonemptiness for nondeterministic parity tree automata can be decided in time polynomial in the number of states and exponential in the number of priorities. Q.E.D.

As any ω-regular multiplayer game can be reduced to an equivalent multiplayer parity game (and thus also to a multiplayer Rabin game), Theorem 9 implies the decidability of SPE. For LTL games the reduction gives an algorithm running in doubly exponential time. As the problem of deciding

the winner in a two-player zero-sum LTL game is already 2ExpTime-complete [18], this is optimal.

Corollary 10. The problem SPE is decidable. For multiplayer LTL games the problem is 2ExpTime-complete.

We point out another consequence of our reduction. By Rabin's basis theorem [19], every regular, nonempty tree language contains a *regular tree*, i.e. a tree with only finitely many nonisomorphic subtrees. It is easy to see that a tree $t : \{0,1\}^* \to V \times \{0,1\}$ originating from a strategy profile $(\sigma_i)_{i \in \Pi}$ is regular if and only if each σ_i is a finite-state strategy. Thus we have the following theorem.

Theorem 11. Let $(\mathcal{G}, v_0)$ be an ω-regular multiplayer game played on a finite arena and $x \in \{0,1\}^k$. Then $(\mathcal{G}, v_0)$ has a subgame perfect equilibrium with payoff x if and only if $(\mathcal{G}, v_0)$ has a finite-state subgame perfect equilibrium with payoff x.

Intuitively, the theorem says that any subgame perfect equilibrium of an ω-regular multiplayer game played on a finite arena can be implemented by finite-state strategies. This is the best one can hope for because an arbitrary Nash or subgame perfect equilibrium, in general, cannot be implemented by positional strategies.

Example 12. Consider the following Büchi game with two players 1 and 2 played on the arena depicted in Figure 2: Every vertex is controlled by player 1, and the winning condition of player i is to visit vertex i infinitely often. Obviously, player 1's finite-state strategy of alternating between visits to vertex 1 and vertex 2 induces a subgame perfect equilibrium of the game with payoff $(1,1)$. However, for any positional strategy of player 1 only one player wins the resulting play.
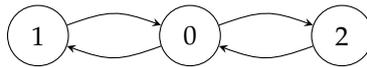

Figure 2. A 2-player Büchi game.

5 Definability

Let us now study the following question: Given winning conditions $W_1, \ldots, W_k \subseteq C^\omega$ defined with respect to a (finite) set C of colours and payoff thresholds x, y, in which logic can we define the class of initialised game arenas that, when equipped with the given winning conditions, admit a subgame perfect equilibrium with a payoff in between the given thresholds? Note that any game arena $G = (V, (V_i)_{i \in \Pi}, E, \chi)$, where $\chi : V \to C$, can be identified with the Kripke structure $(V, (V_i)_{i \in \Pi}, E, (P_c)_{c \in C})$ defined by

$P_c = \{v \in V : \chi(v) = c\}$. Hence any logic that has a semantics for pointed Kripke structures can be used to define a class of initialised game arenas, one possible candidate being the *modal μ-calculus* L-, i.e. basic modal logic extended by least and greatest fixed points. Indeed, it is well known that for any fixed number of priorities the class of initialised two-player game arenas that admit a winning strategy for the first player in the corresponding two-player zero-sum parity game is L_μ-definable [4]. If G is a k-player game arena with colours in C and $W_1, \ldots, W_k \subseteq C^\omega$ are winning conditions, we write $G[W_1, \ldots, W_k]$ for the corresponding game. Then our result can be stated as follows.

Theorem 13. Let $W_1, \ldots, W_k \subseteq C^\omega$ be ω-regular and $x, y \in \{0,1\}^k$. Then there exists a formula $\psi \in L_\mu$ such that the following equivalence holds for every *k*-player game arena *G* with colours in *C* and every vertex *v* of *G*:

$$G, v \models \psi \Leftrightarrow (G[W_1, \ldots, W_k], v) \text{ has a SPE with a payoff} \geq x \text{ and} \leq y.$$

Note that Theorem 13 is closely related to Theorem 9 and Corollary 10. Indeed, it can be shown that the formula ψ as defined in Theorem 13 can be constructed effectively. As the model-checking problem for L_μ on finite Kripke structures is decidable, this gives another method to solve the problem SPE.

As a special case, Theorem 13 implies that for every fixed ω-regular winning condition the class of initialised two-player game arenas that admit a winning strategy for player 0 in the corresponding two-player zero-sum game is definable in L_μ, a fact that, surprisingly, seems not to have been stated anywhere before.

References

1. D. Berwanger, E. Grädel & S. Kreutzer. Once upon a time in the west. Determinacy, complexity and definability of path games. In *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2003*, vol. 2580 of *LNCS*, pp. 226–240. Springer-Verlag, 2003.

2. K. Chatterjee, T. A. Henzinger & M. Jurdziński. Games with secure equilibria. In *Proceedings of the 19th Annual Symposium on Logic in Computer Science, LICS 2004*, pp. 160–169. IEEE Computer Society Press, 2004.

3. K. Chatterjee, R. Majumdar & M. Jurdziński. On Nash equilibria in stochastic games. In *Proceedings of the 13th Annual Conference of the European Association for*

Computer Science Logic, CSL 2004, vol. 3210 of *LNCS*, pp. 26–40. Springer-Verlag, 2004.

4. E. A. Emerson & C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, FoCS '91*, pp. 368–377. IEEE Computer Society Press, 1991.

5. E. A. Emerson, C. S. Jutla & A. P. Sistla. On model-checking for fragments of μ-calculus. In *Proceedings of the 5th International Conference on Computer Aided Verification, CAV '93*, vol. 697 of *LNCS*, pp. 385–396. Springer-Verlag, 1993.

6. E. A. Emerson & A. P. Sistla. Deciding full branching time logic. *Information and Control*, 61(3):175–201, 1984.

7. D. Gale & F. M. Stewart. Infinite games with perfect information. In *Contributions to the Theory of Games II*, vol. 28 of *Annals of Mathematical Studies*, pp. 245–266. Princeton University Press, 1953.

8. D. Janin & I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of the 7th International Conference on Concurrency Theory, CONCUR '96*, vol. 1119 of *LNCS*, pp. 263–277. Springer-Verlag, 1996.

9. M. Jurdziński. Small progress measures for solving parity games. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS 2000*, vol. 1770 of *LNCS*, pp. 290–301. Springer-Verlag, 2000.

10. H. W. Kuhn. Extensive games and the problem of information. In *Contributions to the Theory of Games II*, vol. 28 of *Annals of Mathematical Studies*, pp. 193–216. Princeton University Press, 1953.

11. O. Kupferman & M. Y. Vardi. Church's problem revisited. *The Bulletin of Symbolic Logic*, 5(2):245–263, 1999.

12. D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.

13. D. A. Martin. An extension of Borel determinacy. *Annals of Pure and Applied Logic*, 49(3):279–293, 1990.

14. A. W. Mostowski. Games with forbidden positions. Technical Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland, 1991.

15. D. E. Muller & P. E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141(1–2):69–107, 1995.

16. J. F. Nash Jr. Equilibrium points in *N*-person games. *Proceedings of the National Academy of Sciences of the USA*, 36:48–49, 1950.

17. M. J. Osborne & A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

18. A. Pnueli & R. Rosner. Distributed reactive systems are hard to synthesize. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science, FoCS '90*, pp. 746–757. IEEE Computer Society Press, 1990.

19. M. O. Rabin. Automata on infinite objects and Church's problem. *American Mathematical Society*, 1972.

20. R. Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324 and 667–689, 1965.

21. W. Thomas. On the synthesis of strategies in infinite games. In *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science, STACS '95*, vol. 900 of *LNCS*, pp. 1–13. Springer-Verlag, 1995.

22. W. Thomas. Languages, automata, and logic. In *Handbook of Formal Language Theory*, vol. III, pp. 389–455. Springer-Verlag, 1997.

23. M. Ummels. Rational behaviour and strategy construction in infinite multiplayer games. Diploma Thesis, RWTH Aachen, Germany, 2005.

24. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.

Appendix

Proof of Proposition 2

Let $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (W_i)_{i \in \Pi})$ where each $W_i \subseteq C^\omega$ is ω-regular. Then for each player $i \in \Pi$ we can fix a *deterministic parity automaton* $\mathcal{A}_i$ that recognises W_i (see, for example, [22]). Such an automaton has the form $A_i = (Q_i, C, q_i^0, \delta_i, \Omega_i)$ where Q_i is a finite set of states, q_i^0 is the initial state, $\delta_i : Q_i \times \Sigma \to Q_i$ is the transition function and $\Omega_i : Q \to \omega$ is the priority function. An input word $\alpha \in C^\omega$ is accepted by $\mathcal{A}_i$ if the run $\rho \in Q^\omega$ of $\mathcal{A}_i$ on α, defined by $\rho(0) = q_i^0$ and $\rho(k+1) = \delta(\rho(k), \alpha(k))$ for all $k < \omega$, fulfils the parity condition given by the priority function Ω_i. For a word $w \in C^*$ let $\delta_i^*(w)$ be the state reached by $\mathcal{A}_i$ after reading w, i.e. $\delta_i^*(\varepsilon) = q_i^0$ and $\delta_i^*(wc) = \delta_i(\delta_i^*(w), c)$ for all $w \in C^*$, $c \in C$.

We define the multiplayer parity game $\mathcal{G}'$ as follows: The game has the synchronised product $G \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_k$ of the original arena G with each automaton $\mathcal{A}_i$ as its arena. $G \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_k$ has vertices $(v, (q_i)_{i \in \Pi})$ where $v \in V$ and $q_i \in Q_i$, and there is an edge from $(u, (p_i)_{i \in \Pi})$ to $(v, (q_i)_{i \in \Pi})$ if and only if (u, v) is an edge in G and $\delta_i(p_i, \chi(u)) = q_i$. A vertex $(v, (q_i)_{i \in \Pi})$ is controlled by player j if and only if $v \in V_j$. The colour of a vertex $(v, (q_i)_{i \in \Pi})$ is $(q_i)_{i \in \Pi}$, and the priority function for player i in this game is inherited from the automaton $\mathcal{A}_i$. Finally, the initial vertex of the game is $v_0' = (v_0, (q_i^0)_{i \in \Pi})$.

Note that if a strategy profile $(\sigma_i)_{i \in \Pi}$ is a Nash or subgame perfect equilibrium of $(\mathcal{G}, v_0)$ it is also a Nash or subgame perfect equilibrium of $(\mathcal{G}', v_0')$, respectively, with the same payoff. Vice versa, let $(\sigma_i')_{i \in \Pi}$ be a strategy profile of $\mathcal{G}'$. Each strategy σ_i' induces a strategy σ_i of player i in $\mathcal{G}$, defined by $\sigma_i(v_1 \ldots v_k) = v$ for the unique $v \in V$ such that

$$(v, \delta_i^*(\chi(v_1 \ldots v_k))) = \sigma_i'((v_1, \delta_i^*(\varepsilon)) \ldots (v_k, \delta_i^*(\chi(v_1 \ldots v_{k-1})))).$$

Obviously, σ_i is a finite-state strategy if σ_i' is one. Moreover, it is easy to show that if $(\sigma_i')_{i \in \Pi}$ is a Nash or subgame perfect equilibrium of $(\mathcal{G}', v_0')$ then $(\sigma_i)_{i \in \Pi}$ is a Nash or subgame perfect equilibrium of $(\mathcal{G}, v_0)$, respectively, with the same payoff.

For the second part of the proposition, assume that each winning condition of $\mathcal{G}$ is given by an LTL formula of length at most m. Then each automaton $\mathcal{A}_i$ can be realised with $2^{2^{O(m)}}$ states and $2^{O(m)}$ priorities [6]. Thus $\mathcal{G}'$ has $2^{2^{O(m)+\log k}}$ states and $2^{O(m)}$ priorities for each player.   Q.E.D.

Proof of Theorem 5 (continued)

It remains to show that $(\tau_i)_{i \in \Pi}$ is a subgame perfect equilibrium of $(\mathcal{G}, v_0)$ for any initial vertex $v_0 \in V$. First note that σ_i is winning in $(\mathcal{G}_i^{\xi}, v)$ if σ_i^{α} is winning in $(\mathcal{G}_i^{\alpha}, v)$ for some ordinal α because if σ_i^{α} is winning in $(\mathcal{G}_i^{\alpha}, v)$ every play of $(\mathcal{G}_i^{\alpha+1}, v)$ is consistent with σ_i^{α} and therefore won by player i. As $E^{\xi} \subseteq E^{\alpha+1}$, this also holds for every play of $(\mathcal{G}_i^{\xi}, v)$. Now let hv be any history of $(\mathcal{G}, v_0)$. We show that $(\tau_j|_h)_{j \in \Pi}$ is a Nash equilibrium of $(\mathcal{G}, v)$. Towards this, let τ' be any strategy of any player $i \in \Pi$ in $\mathcal{G}$; let $\pi = \langle (\tau_j|_h)_{j \in \Pi} \rangle_v$, and let $\pi' = \langle \tau'|_h, (\tau_j|_h)_{j \in \Pi \setminus \{i\}} \rangle_v$. We show that π is won by player i or that π' is not won by player i. The claim is trivial if $\pi = \pi'$. Thus assume that $\pi \neq \pi'$ and fix the least $k < \omega$ such that $\pi(k+1) \neq \pi'(k+1)$. Clearly, $\pi(k) \in V_i$ and $\tau'(h\pi(0)\dots\pi(k)) \neq \tau_i(h\pi(0)\dots\pi(k))$. Without loss of generality, let $k = 0$. We distinguish the following two cases:

- σ_i is not winning in $(\mathcal{G}_i^{\xi}, v)$. By the definition of each τ_j, π is a play of $(\mathcal{G}_i^{\xi}, v)$. We claim that π is consistent with σ_i, which implies that π is won by player i. Otherwise fix the least $l < \omega$ such that $\pi(l) \in V_i$ and $\sigma_i(\pi(l)) \neq \pi(l+1)$. As σ_i is winning in $(\mathcal{G}_i^{\xi}, v)$, σ_i is also winning in $(\mathcal{G}_i^{\xi}, \pi(l))$. But then $(\pi(l), \pi(l+1)) \in E^{\xi} \setminus E^{\xi+1}$, a contradiction to $E^{\xi} = E^{\xi+1}$.

- σ_i is not winning in $(\mathcal{G}_i^{\xi}, v)$. Then σ_{-i} is winning in $(\mathcal{G}_i^{\xi}, v)$. As $\tau'(hv) \neq \tau_i(hv)$, player i has deviated, and so we have $\pi' = \langle \tau'|_h, (\sigma_{j,i})_{j \in \Pi \setminus \{i\}} \rangle_v$. We claim that π' is a play of $(\mathcal{G}_i^{\xi}, v)$. As σ_{-i} is winning in $(\mathcal{G}_i^{\xi}, v)$, this implies that π' is not won by player i. Otherwise fix the least $l < \omega$ such that $(\pi'(l), \pi'(l+1)) \notin E^{\xi}$ together with the ordinal α such that $(\pi'(l), \pi'(l+1)) \in E^{\alpha} \setminus E^{\alpha+1}$. Clearly, $\pi'(l) \in V_i$. Thus σ_i^{α} is winning in $(\mathcal{G}_i^{\alpha}, \pi'(l))$, which implies that σ_i is winning in $(\mathcal{G}_i^{\xi}, \pi'(l))$. As π' is consistent with σ_{-i}, this means that σ_{-i} is not winning in $(\mathcal{G}_i^{\xi}, v)$, a contradiction.

As $(\tau_j|_h)_{j \in \Pi}$ is a Nash equilibrium of $(\mathcal{G}, v) = (\mathcal{G}|_h, v)$ for every history hv of $(\mathcal{G}, v_0)$, $(\tau_j)_{j \in \Pi}$ is a subgame perfect equilibrium of $(\mathcal{G}, v_0)$.   Q.E.D.

Proof of Theorem 7

Let $\mathcal{G} = (\{1, 2\}, V, V_1, V_2, E, \chi, \Omega_1, \Omega_2)$ be a two-player parity game. We show that the positional strategies $\tau_1 = \sigma_{1,2}$ and $\tau_2 = \sigma_{2,1}$ as defined in the proof of Theorem 5 form a subgame perfect equilibrium of $(\mathcal{G}, v_0)$ for any initial vertex v_0. As we have $\mathcal{G}|_h = \mathcal{G}$, $\tau_1|_h = \tau_1$ and $\tau_2|_h = \tau_2$ for all histories h of $\mathcal{G}$, it suffices to show that (τ_1, τ_2) is a Nash equilibrium of $(\mathcal{G}, v)$ for all $v \in V$. Let $v \in V$ and $\pi = \langle \tau_1, \tau_2 \rangle_v$. Moreover, let τ' be any other strategy of player 2 in $(\mathcal{G}, v)$, and let $\pi' = \langle \tau_1, \tau' \rangle_v$. We have to show that π is won by player 2 or that π' is not won by player 2. We distinguish the following two cases (using the same notation as in the proof of Theorem 5):

- σ_2 is winning in $(\mathcal{G}_2^\xi, v)$. By the definition of the strategies τ_1 and τ_2, π is a play of $(\mathcal{G}_2^\xi, v)$. We claim that π is consistent with σ_2, which implies that π is won by player 2. Otherwise fix the least $l < \omega$ such that $\pi(l) \in V_2$ and $\sigma_2(\pi(l)) \neq \pi(l+1)$. As σ_2 is winning in $(\mathcal{G}_2^\xi, v)$, σ_2 is also winning in $(\mathcal{G}_2^\xi, \pi(l))$. But then $(\pi(l), \pi(l+1)) \in E^\xi \setminus E^{\xi+1}$, a contradiction to $E^\xi = E^{\xi+1}$.

- σ_2 is not winning in $(\mathcal{G}_2^\xi, v)$. Then τ_1 is winning in $(\mathcal{G}_2^\xi, v)$. We claim that π' is a play of $(\mathcal{G}_2^\xi, v)$, which implies that π' is not won by player 2. Otherwise fix the least $l < \omega$ such that $(\pi'(l), \pi'(l+1)) \notin E^\xi$ together with the ordinal α such that $(\pi'(l), \pi'(l+1)) \in E^\alpha \setminus E^{\alpha+1}$. Clearly, $\pi'(l) \in V_2$. Thus σ_2^α is winning in $(\mathcal{G}_2^\alpha, \pi'(l))$, which implies that σ_2 is winning in $(\mathcal{G}_2^\xi, \pi'(l))$. As π' is consistent with τ_1, this means that τ_1 is not winning in $(\mathcal{G}_2^\xi, v)$, a contradiction.

The reasoning for player 1 is analogous.   Q.E.D.

Proof of Theorem 9

Let $\mathcal{G} = (\Pi, V, (V_i)_{i \in \Pi}, E, \chi, (\Omega_i)_{i \in \Pi})$ be a multiplayer Rabin game with n vertices, k players, at most d pairs for each player and initial vertex v_0. Furthermore, let $x, y \in \{0,1\}^k$. Without loss of generality, we can assume that $\mathcal{G}$ is binary. Moreover, let us fix a linear ordering $<$ on V, and let $t : \{0,1\}^* \to V$ be the *unravelling* of $\mathcal{G}$ from v_0, i.e. $t(\varepsilon) = v_0$, and $t(w0)$ and $t(w1)$ are the first and the last successor of $t(w)$, respectively. Given a strategy profile $(\sigma_i)_{i \in \Pi}$ of $(\mathcal{G}, v_0)$, let $t_{(\sigma_i)_{i \in \Pi}} : \{0,1\}^* \to V \times \{0,1\}$ be defined by

$$t_{(\sigma_i)_{i \in \Pi}}(w) = \begin{pmatrix} t(w) \\ x \end{pmatrix}$$

for the least $x \in \{0,1\}$ with $t(wx) = \sigma_i(t(\varepsilon) \dots t(w))$ if $t(w) \in V_i$. We call any tree of the form $t_{(\sigma_i)_{i \in \Pi}}$ for some strategy profile $(\sigma_i)_{i \in \Pi}$ of $(\mathcal{G}, v_0)$ a *strategy tree of $(\mathcal{G}, v_0)$*.

As outlined in the proof sketch, we construct two tree automata. The first one checks for an infinite binary tree over the alphabet $V \times \{0,1\}$ whether it is a strategy tree of the game. The automaton is nondeterministic and works with trivial acceptance, i.e. every run of the automaton is accepting. The second one checks for a strategy tree whether the corresponding strategy profile is a subgame perfect equilibrium of the game with a payoff $\geq x$ and $\leq y$. For alternating tree automata we use the same notation as, for example, in [11]. In particular, the transition function of such an automaton is a total function mapping a state and an input letter to a positive Boolean formula with atomic propositions (i, q) where $i \in \{0,1\}$ is a *direction* and q is an automaton state.

Lemma. There exists a nondeterministic tree automaton $\mathcal{A}_{\mathcal{G}, v_0}$ such that a tree $t : \{0,1\}^* \to V \times \{0,1\}$ is accepted by $\mathcal{A}_{\mathcal{G}, v_0}$ if and only if t is a strategy tree of $(\mathcal{G}, v_0)$. $\mathcal{A}_{\mathcal{G}, v_0}$ has $O(n)$ states and works with trivial acceptance.

Proof. On input t the automaton has to check whether the first component of $t(\varepsilon)$ is v_0, whether for each tree node w the vertices given by the first component of $t(w0)$ and $t(w1)$, respectively, are indeed *the* successors of the vertex given by the first component of $t(w)$ and whether for each tree node w the second component of $t(w)$ is 0 if the vertex given by the first component of $t(w)$ has only one successor. The automaton has state set $V \times \{1, 2\}$ where a state (v, n) has the intended meaning that the current vertex is v and that this vertex has n successors. The initial state is $(v_0, |v_0 E|)$. Finally, the transition function δ is defined by

$$\delta((u, n), \begin{pmatrix} v \\ z \end{pmatrix}) = \begin{cases} \bigwedge\limits_{i=1}^{2} (i-1, (v_i, |v_i E|)) & \text{if } u = v, n = |vE| \text{ and } z < n, \\ \text{false} & \text{otherwise} \end{cases}$$

where $v_1 \leq v_2$ are the successors of v.

Lemma. There exists an alternating parity tree automaton $\mathcal{A}_{\mathcal{G},x,y}$ such that a strategy tree $t_{(\sigma_i)_{i \in \Pi}}$ of $(\mathcal{G}, v_0)$ is accepted by $\mathcal{A}_{\mathcal{G},x,y}$ if and only if $(\sigma_i)_{i \in \Pi}$ is a subgame perfect equilibrium of $(\mathcal{G}, v_0)$ with a payoff $\geq x$ and $\leq y$. $\mathcal{A}_{\mathcal{G},x,y}$ has $O(kd)$ states and $O(1)$ priorities.

Proof. W.l.o.g., assume that $\Omega_i = \{(G_{i1}, R_{i1}), \ldots, (G_{id}, R_{id})\}$ with $G_{ij} \cap R_{ij} = \emptyset$. We define $\mathcal{A}_{\mathcal{G},x,y}$ as follows: The automaton has state set

$$\Pi \times \{1, \ldots, d\} \times \{G, R, Y\} \times \{+, -, \forall\} \cup \{q_0, q_?\}.$$

Intuitively, if the automaton is in a state $(i, j, *, +)$ it checks whether the acceptance condition of the j-th pair in player i's winning condition is fulfilled in the outcome of the given strategy profile; if it is in a state $(i, j, *, -)$, it checks whether the acceptance condition of the j-th pair in player i's winning condition is violated in the outcome of the given strategy profile; if it is in a state $(i, j, *, \forall)$, it checks whether the acceptance condition of the j-th pair in player i's winning condition is violated for all possible choices of player i; if it is in state $q_?$, it chooses a subgame; q_0 is the initial state of the automaton. In the following for a vertex $v \in V$ let

$$\delta_{ij}(v) = \begin{cases} G & \text{if } \chi(v) \in G_{ij}, \\ R & \text{if } \chi(v) \in R_{ij}, \\ Y & \text{otherwise.} \end{cases}$$

Now we specify the transition function δ. In a state $(i, j, *, +)$ or $(i, j, *, -)$, the automaton just follows the move given by the input tree. Thus we define

$$\delta((i, j, *, +), \begin{pmatrix} v \\ z \end{pmatrix}) = (z, (i, j, \delta_{ij}(v), +))$$

and

$$\delta((i, j, *, -), \begin{pmatrix} v \\ z \end{pmatrix}) = (z, (i, j, \delta_{ij}(v), -)).$$

In a state $(i, j, *, \forall)$ the automaton universally chooses both possible moves at vertices controlled by player i and follows the move given by the input tree at all other vertices. Thus we define

$$\delta\left((i, j, *, \forall), \binom{v}{z}\right) = \bigwedge_{z'}(z', (i, j, \delta_{ij}(v), \forall))$$

if $v \in V_i$, and

$$\delta\left((i, j, *, \forall), \binom{v}{z}\right) = (z, (i, j, \delta_{ij}(v), \forall))$$

if $v \notin V_i$. When choosing a subgame, the automaton universally chooses to continue choosing a subgame or to check whether the player who controls the current vertex cannot positively deviate. In the latter case the automaton nondeterministically guesses whether to show that the outcome of the given strategy profile is won by this player or to show that this player cannot win by choosing any other strategy (which can be assumed to differ from her strategy at the current history). Thus we define

$$\delta\left(q_?, \binom{v}{z}\right) = \bigwedge_{z'}(z', q_?)$$
$$\wedge \left[\bigvee_{j}(z, (i, j, \delta_{ij}(v), +)) \vee \bigwedge_{j}(1 - z, (i, j, \delta_{ij}(v), \forall)) \right],$$

where $v \in V_i$. In the initial state the automaton universally chooses to choose a subgame (which may be the game itself) or to show that the strategy profile given by the input tree has the correct payoff. In the latter case the automaton universally chooses a player i with $x_i = 1$ or $y_i = 0$ and shows that player i wins or loses the outcome of the given strategy profile, respectively. Thus we define

$$\delta\left(q_0, \binom{v}{z}\right) = \delta\left(q_?, \binom{v}{z}\right)$$
$$\wedge \bigwedge_{x_i=1} \bigvee_{j}(z, (i, j, \delta_{ij}(v), +)) \wedge \bigwedge_{y_i=0} \bigwedge_{j}(z, (i, j, \delta_{ij}(v), -)).$$

It remains to specify the priority function Ω. For states of the form $(i, j, c, +)$, the priority function is defined by

$$\Omega(i, j, c, +) = \begin{cases} 2 & \text{if } c = G, \\ 1 & \text{if } c = R, \\ 3 & \text{if } c = Y. \end{cases}$$

Thus an infinite sequence of $+$-*states* is accepted iff the sequence contains infinitely many 'G-states' but only finitely many R-*states*. For states of the form $(i, j, c, -)$ or $(i, j, c, \forall)$, the priority function is defined by

$$\Omega(i, j, c, *) = \begin{cases} 1 & \text{if } c = G, \\ 0 & \text{if } c = R, \\ 2 & \text{if } c = Y. \end{cases}$$

Thus an infinite sequence of $-$-*states* or $\forall$-*states* is accepted iff the sequence contains infinitely many *R-states* or only finitely many *G-states*. Finally, we set $\Omega(q_?) = \Omega(q_0) = 0$. Thus an infinite sequence of ?-*states* is always rejected.

To get a nondeterministic tree automaton $\mathcal{A}_{\mathcal{G},v_0,x,y}$ accepting precisely those trees $t : \{0,1\}^* \to V \times \{0,1\}$ such that $t = t_{(\sigma_i)_{i \in \Pi}}$ for a subgame perfect equilibrium $(\sigma_i)_{i \in \Pi}$ with a payoff of $\geq x$ and $\leq y$, we make the automaton $\mathcal{A}_{\mathcal{G},x,y}$ nondeterministic and subsequently construct the product with the automaton $\mathcal{A}_{\mathcal{G},v_0}$. As an alternating parity tree automaton with m states and r priorities can be transformed into an equivalent nondeterministic parity tree automaton with $2^{O(mr \log mr)}$ states and $O(mr)$ priorities [15], the resulting automaton $\mathcal{A}_{\mathcal{G},v_0,x,y}$ has $O(n) \cdot 2^{O(kd \log kd)}$ states and $O(kd)$ priorities.

It follows that the automaton $\mathcal{A}_{\mathcal{G},v_0,x,y}$ defines a nonempty tree language if and only if $(\mathcal{G}, v_0)$ has a subgame perfect equilibrium with a payoff of $\geq x$ and $\leq y$. Thus it suffices to check wether the tree language defined by $\mathcal{A}_{\mathcal{G},v_0,x,y}$ is nonempty. For a nondeterministic parity tree automaton with m states and r priorities, this can be done in time $O(m^r)$ [5]. Finally, note that the nonemptiness of $\mathcal{A}_{\mathcal{G},v_0,x,y}$ does not depend on the choice of the linear order $<$.   Q.E.D.

Proof of Theorem 13

We use the same proof technique as deployed in [1] where a similar theorem for two-player zero-sum *path games* has been shown. The main idea is to make use of the fact that the property expressed by the right-hand side of the equivalence stated in Theorem 13 is invariant under bisimulation. This allows us to restrict ourselves to trees as game arenas with their respective root as initial vertex (by replacing an arbitrary arena with its unravelling from the initial vertex). Furthermore, it has been shown by Janin and Walukiewicz [8] that every bisimulation-invariant class of trees that is definable in MSO is already definable in L_μ. Thus it suffices to give an MSO formula φ that describes the above property over game arenas that are trees. A, without loss of generality, positional strategy σ for player j in a game played on a tree $T = (V, (V_i)_{i=1,\ldots,k}, E, (P_c)_{c \in C})$ with root $\varepsilon \in V$ can be identified with the set $\{\sigma(v) : v \in V_j\}$. Conversely, a set X can be identified with a strategy of player j in such a game if X satisfies the MSO formula

$$\mathrm{Strat}_j(X) := \forall v \big[(V_j v \to \exists! u (Evu \land Xu)) \land (\neg V_j v \to \forall u (Evu \to \neg Xu)) \big].$$

It is an easy exercise to write down an MSO formula $\vartheta(X_1, \ldots, X_k)$ expressing that strategies $X_1, \ldots, X_k$ define a subgame perfect equilibrium of the game $(T[W_1, \ldots, W_k], \varepsilon)$ with a payoff in between the given thresholds. Then the desired formula φ is $\exists X_1 \ldots \exists X_k (\bigwedge_i \mathrm{Strat}_i(X_i) \land \vartheta)$.   Q.E.D.

19