Parity Games and Propositional Proofs

Arnold Beckmann[⋆1], Pavel Pudlák[⋆⋆2], and Neil Thapen[⋆⋆2]

[1] Department of Computer Science, College of Science, Swansea University, Swansea SA2 8PP, UK. a.beckmann@swansea.ac.uk

[2] Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, 115 67 Praha 1, Czech Republic. {pudlak,thapen}@math.cas.cz

Abstract. A propositional proof system is *weakly automatizable* if there is a polynomial time algorithm which separates satisfiable formulas from formulas which have a short refutation in the system, with respect to a given length bound. We show that if the resolution proof system is weakly automatizable, then parity games can be decided in polynomial time. We also define a combinatorial game and prove that resolution is weakly automatizable if and only if one can separate, by a set decidable in polynomial time, the games in which the first player has a positional winning strategy from the games in which the second player has a positional winning strategy.

1 Introduction

Parity games, mean payoff games and simple stochastic games are three classes of two player games, played by moving a token around a finite graph. In particular parity games have important applications in automata theory, logic, and verification [11]. The main computational problem for all of these games is to decide, given an instance of a game, which player has a positional winning strategy. From this point of view, parity games are reducible to mean payoff games, and mean payoff games are reducible to simple stochastic games [19, 23]. It is known that the decision problem for simple stochastic games is reducible to a search problem in the intersection of the classes PLS and PPAD [6, 13] (which are believed to be incomparable [4]). None of the decision problems is known to be in P, despite intensive research work on developing algorithms for them. For several of the existing algorithms, exponential lower bounds on their runtime have been given recently [9, 10].

Automatizability is an important concept for automated theorem proving. Call a propositional proof system *automatizable* if there is an algorithm which, given a tautology, produces a proof in time polynomial in the size of its smallest proof—this time condition is the best we can hope for, assuming NP $\neq$ coNP. Automatizability is a very strict notion. For example, Alekhnovich and Razborov [1]

⋆ This research was partially done while the authors were visiting fellows at the Isaac Newton Institute for the Mathematical Sciences in the programme "Semantics & Syntax".

⋆⋆ Partially supported by grant IAA100190902 of GA AV ČR.

have shown that resolution is not automatizable under a reasonable assumption in parameterised complexity theory. *Weak automatizability* is a relaxation of automatizability, where proofs of tautologies can be given in an arbitrary proof system, and only the time of finding such proofs is restricted to polynomial in the size of the smallest proof in a given proof system. This characterisation of weak automatizability is equivalent to the existence of a polynomial time algorithm which separates satisfiable formulas from formulas which have a short refutation in the system with respect to a given length bound.

Two recent papers have shown a connection between weak automatizability and the above mentioned games. Atserias and Maneva showed that if a certain proof system (called PK_1 in our notation) is weakly automatizable, then the decision problem for mean payoff games is in P [3]. Huang and Pitassi strengthened this to the decision problem for simple stochastic games [12]. In this paper we extend these results to resolution and parity games. In Sect. 2 below we show that if resolution is weakly automatizable, then parity games can be decided in polynomial time.

In order to obtain a kind of reverse direction of this result, in Sect. 3 we define a new game, the *point-line game*, also about moving a token around a finite graph. We show that its complexity is equivalent to that of resolution, in a certain sense. In particular, resolution is weakly automatizable if and only if one can separate, by a set in P, the games in which the first player has a positional winning strategy from the games in which the second player has a positional winning strategy.

The essential part of the argument in Sect. 2, together with one direction of Sect. 3, is to show that there is a polynomial-size propositional proof that winning strategies cannot exist simultaneously for both players in a game. Propositional proofs are complicated combinatorial objects, and constructing them by hand can be difficult. Instead, we work with weak first-order bounded arithmetic theories which capture the logical content of these proof systems, and rely on known translations of these to do the hard work of actually constructing the propositional proofs for us. These translations go back to Paris and Wilkie [17]. Later work has given finer results about the logical depth of the propositional proofs. The main result we need, a first order-theory which translates into polynomial-size resolution, is essentially due to Krajíček [14–16].

The full version of this paper, in preparation, will extend our methods to give simplified proofs of the results mentioned above relating weak automatizability of the proof system PK_1 to the decision problem for mean payoff and simple stochastic games [3, 12]. Furthermore, it will include a detailed proof of the translation of first order-theories into polynomial-size propositional proofs, with extra information about the fan-in k of connectives located at the maximum depth of propositional formulas, namely that k can be bounded by a constant that we can read directly from the formulas appearing in the first-order proof.

Finding a polynomial time algorithm to solve parity games is a long-standing open problem, so it is tempting to interpret our main result about parity games and resolution as evidence either that resolution is not weakly automatizable, or at least that if it is, then this will be hard to prove. On the other hand, modern SAT solvers typically use algorithms which, given a formula, generate either a satisfying assignment or what is essentially a resolution proof that the formula is unsatisfiable. Thus it seems that a necessary condition for a formula to be tractable by these SAT solvers is that the formula is either satisfiable, or has a short resolution refutation. Our reduction can be used to translate a parity game into a formula that satisfies at least this necessary condition. Hence, a possible application is to try to combine our reduction with a SAT solver, to obtain a new algorithm for solving parity games.

1.1 Resolution Proof Systems

For $k \geq 1$, the propositional proof system $\mathrm{Res}(k)$ is defined as follows. Propositional formulas are formed from propositional variables $p_0, p_1, p_2, \ldots$, negation $\neg$, and unbounded fan-in conjunctions and disjunctions $\bigwedge$ and $\bigvee$. Variables are called *atoms*, and atoms and negated atoms are together called *literals*. Formulas are then defined inductively: each literal is a formula, and if Φ is a finite non-empty set of formulas then $\bigwedge\Phi$ and $\bigvee\Phi$ are formulas. For a formula φ, we use $\neg\varphi$ as an abbreviation for the formula formed from φ by interchanging $\bigwedge$ and $\bigvee$ and interchanging atoms and their negations. We treat the binary connectives $\wedge$ and $\vee$ as the obvious set operations, for example $\bigvee\Phi \vee \bigvee\Psi = \bigvee(\Phi \cup \Psi)$. If a formula is a conjunction, we will sometimes treat it as the set of its conjuncts, and vice versa.

A k-DNF is a disjunction of conjunctions of literals, where each conjunction is of size at most k. Each line in a $\mathrm{Res}(k)$-proof is a k-DNF, usually written as the list of disjuncts separated by commas. The rules of $\mathrm{Res}(k)$ are as follows, where Γ, Δ stand for sets of formulas, possibly empty, A, B for formulas, and a_i for literals:

$$\wedge\text{-introduction} \quad \frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \wedge B}$$

$$\text{weakening} \quad \frac{\Gamma}{\Gamma, \Delta} \qquad\qquad \text{cut} \quad \frac{\Gamma, a_1 \wedge \ldots \wedge a_m \qquad \Gamma, \neg a_1, \ldots, \neg a_m}{\Gamma}$$

We also allow introduction of logical axioms $\overline{\;a, \neg a\;}$ for atoms a.

A $\mathrm{Res}(k)$ *refutation* of a set of disjunctions Γ is a sequence of disjunctions ending with the empty disjunction, such that each line in the proof is either in Γ, or a logical axiom, or follows from earlier disjunctions in the sequence by a rule. The system $\mathrm{Res}(1)$ is called *resolution* and is denoted by Res.

We will also consider the proof system PK_1, which is defined in the same way as $\mathrm{Res}(k)$ but now dropping the bound on the number of literals in conjunctions. That is, lines in PK_1 proofs are unrestricted DNFs, instead of k-DNFs in case of $\mathrm{Res}(k)$.

1.2 Bounded Arithmetic

We could obtain the results of this paper by a careful use of the conventional Buss-style bounded arithmetic theory T^2_1 [5]. However, these would introduce unnecessary complications to deal with sharply bounded quantification, so instead we will work with simpler systems.

For $r \in \mathbb{N}$, we will say that a function $f\colon \mathbb{N}^r \to \mathbb{N}$ is *polynomially bounded* if there is some polynomial p such that $f(\bar{x}) \leq p(\bar{x})$ for all $\bar{x}$. Let L be the language consisting of the constant symbols 0 and 1, and, for every $r \in \mathbb{N}$, a function symbol for every polynomially bounded function $\mathbb{N}^r \to \mathbb{N}$ and a relation symbol for every relation on $\mathbb{N}^r$. If the reader is uncomfortable with such a large language, it can be replaced by any reasonably rich language extending $\{0, 1, +, \cdot, <\}$ as long as all functions in the language are polynomially bounded. Let BASE be the set of true universal L-sentences. We will use this as our base theory.

We extend L to a language $L^+ = L \cup \bar{R}$ by adding a tuple $\bar{R}$ of finitely many new relation symbols. We will use these to stand for edges in a graph, or strategies in a game, or whatever other objects we need to reason about.

Adapting notation from Wilmers [22], we define a *strict* U_d *formula* to be one consisting of d alternating blocks of bounded quantifiers, beginning with a universal block, followed by a quantifier-free L^+ formula. To obtain optimal results about the depth of the propositional translations of these formulas, we add a technical requirement: the quantifier-free part should have the form of a CNF if d is odd, or a DNF if d is even. Any quantifier-free formula is logically equivalent to one in either form, so in the first-order proofs we construct in this paper we can ignore this requirement. A U_d *formula* is a subformula of a strict U_d formula. The strict E_d formulas and the E_d formulas are defined dually.

We remark that we will almost always work with bounded rather than unbounded quantifiers, and we will often not write the bounds if they are obvious, for example if we are quantifying over the vertices of a given finite graph.

For $d \geq 0$, we define U_d-IND to be BASE together with the usual induction scheme

$$\forall a,\ \phi(0) \wedge \forall x{<}a[\phi(x) \to \phi(x+1)] \to \phi(a)$$

for each U_d formula $\phi(x)$, which may also contain other parameters. The theory E_d-IND is defined similarly.

Similarly we define U_d-MIN to be the usual scheme asserting that any nonempty U_d (with parameters) subset of an interval $[0, a)$ has a least element. The schemes E_d-MIN, U_d-MAX and E_d-MAX are the obvious variants of this.

Lemma 1. *For $d \geq 0$, the following hold over* BASE*:*

1. E_d-IND *is equivalent to* U_d-IND
2. E_d-MAX *is equivalent to* E_d-MIN
3. U_d-MAX *is equivalent to* U_d-MIN
4. U_{d+1}-IND *proves* U_d-MAX *and* E_{d+1}-MAX. □

We now define a version of the Paris-Wilkie translation of first-order proofs in bounded arithmetic into small propositional proofs [17]. We will use this as a tool for constructing resolution refutations out of U_2-IND proofs. For each relation symbol in $\bar{R}$ of arity s, we fix a propositional variable $r_{i_1,\ldots,i_s}$ for each tuple of numbers $i_1, \ldots, i_s$. We assume that all these propositional variables, for all relation symbols in $\bar{R}$, are pairwise distinct.

Let $\top$ and $\bot$ denote the truth values *true* and *false*, respectively. An assignment α is a total map from first-order variables to numbers, in which at most finitely many variables are assigned non-zero values. For an assignment α, a variable x and a number n, we write $\alpha[x \mapsto n]$ for the assignment which maps x to n and leaves the mapping of all other variables unchanged. We write $[x \mapsto n]$ for the assignment which maps x to n and all other variables to 0.

Definition 2. *We compute propositional translations as follows.*

1. *Any L formula ϕ has a definite truth value under α. If ϕ evaluates to true we let $\langle\phi\rangle_\alpha$ be $\top$, and if it evaluates to false we let $\langle\phi\rangle_\alpha$ be $\bot$.*
2. *For t an L-term, we let $\langle t\rangle_\alpha$ be the evaluation of t under α.*
3. *For R an s-ary relation symbol in $\bar{R}$, and $\bar{t}$ an s-tuple of L-terms, we let $\langle R(\bar{t})\rangle_\alpha$ be the propositional variable $r_{i_1,\ldots,i_s}$ where each $i_j = \langle t_j\rangle_\alpha$, and let $\langle \neg R(\bar{t})\rangle_\alpha$ be the negated variable $\neg r_{i_1,\ldots,i_s}$.*
4. *We let $\langle\phi \wedge \psi\rangle_\alpha$ be $\langle\phi\rangle_\alpha \wedge \langle\psi\rangle_\alpha$ and let $\langle\phi \vee \psi\rangle_\alpha$ be $\langle\phi\rangle_\alpha \vee \langle\psi\rangle_\alpha$.*
5. *We let $\langle\forall x < t\ \phi(x)\rangle_\alpha$ be $\bigwedge\{\langle\phi\rangle_{\alpha[x\mapsto m]} : m < \langle t\rangle_\alpha\}$. Bounded existential quantifiers are similarly translated into disjunctions.*

Finally we simplify by inductively removing $\top$ from conjunctions, removing $\bot$ from disjunctions, replacing conjunctions containing $\bot$ with $\bot$, and replacing disjunctions containing $\top$ with $\top$.

Theorem 3. *Suppose that $\phi_1(x), \ldots, \phi_\ell(x)$ are U_2 formulas, with x the only free variable, such that U_2-IND proves $\forall x\, \neg(\phi_1(x) \wedge \ldots \wedge \phi_\ell(x))$. Then for some $k \in \mathbb{N}$ the family*

$$\Phi_n := \langle\phi_1(x)\rangle_{[x\mapsto n]} \cup \cdots \cup \langle\phi_\ell(x)\rangle_{[x\mapsto n]}$$

has polynomial size $\mathrm{Res}(k)$ refutations. □

1.3 Disjoint NP Pairs

A *disjoint* NP *pair* is simply a pair of disjoint NP sets. In the context of proof complexity, these were first studied by Razborov in [20]. Our presentation follows [18]. A pair (A, B) is *polynomially reducible* to a pair (C, D) if there is a polynomial time function f, defined on all strings, such that $f[A] \subseteq C$ and $f[B] \subseteq D$. A pair (A, B) is *polynomially equivalent* to a pair (C, D) if polynomial reducibility holds in both directions. A pair (A, B) is *polynomially separable* if there is a polynomial time function which takes the value 0 on strings in A and the value 1 on strings in B.

If $\mathcal{P}$ is a propositional proof system, the *canonical pair* $\mathbf{C}_\mathcal{P}$ of $\mathcal{P}$ is the pair (A, B) where

$$A = \{(\phi, 1^m) : \phi \text{ is satisfiable}\}$$
$$B = \{(\phi, 1^m) : \phi \text{ has a } \mathcal{P}\text{-refutation of size at most } m\}.$$

We say that $\mathcal{P}$ is *weakly automatizable* if $\mathbf{C}_{\mathcal{P}}$ is polynomially separable. In other words, $\mathcal{P}$ is weakly automatizable if there is a polynomial time algorithm which separates satisfiable formulas from formulas which have a short refutation in the system with respect to a given length bound. This definition of weakly automatizability is equivalent to others in the literature (see [2]).

To define the *interpolation pair* $\mathbf{I}_{\mathcal{P}}$ of $\mathcal{P}$, let $\Delta_{\mathcal{P}}$ be the set of triples (ϕ, θ, π) where ϕ and θ are propositional formulas in disjoint variables and π is a $\mathcal{P}$-refutation of $\phi \wedge \theta$. Then $\mathbf{I}_{\mathcal{P}}$ is the pair (A, B) where

$$A = \{(\phi, \theta, \pi) \in \Delta_{\mathcal{P}} : \ \phi \text{ is satisfiable}\}$$
$$B = \{(\phi, \theta, \pi) \in \Delta_{\mathcal{P}} : \ \theta \text{ is satisfiable}\}.$$

Given a triple $(\phi, \theta, \pi) \in \Delta_{\mathcal{P}}$, at least one of ϕ and θ must be unsatisfiable. We say that $\mathcal{P}$ *has feasible interpolation* if there is a polynomial time function which, given such a triple as input, outputs 0 if ϕ is unsatisfiable and 1 if θ is unsatisfiable. It is easy to show that $\mathcal{P}$ has feasible interpolation if and only if $\mathbf{I}_{\mathcal{P}}$ is polynomially separable.

Proposition 4 ([2]).
1. *Resolution has feasible interpolation.*
2. *The following list of* NP *pairs are pairwise equivalent: The canonical pairs of* Res, Res(2), Res(3), *…, and the interpolation pairs of* Res(2), Res(3), Res(4), *…, and of* PK_1. □

Finally, we define the *canonical pair* of a class of two-player games to be the pair (A_0, A_1) where A_i is the set of games in which player i has a positional winning strategy. Naturally, for this to make sense we need there to be a definition of what a positional strategy is, and for it to be possible to recognise a positional winning strategy in NP.

2 Parity Games

Following Stirling [21] we will describe parity games in a simplified form, which is linear-time equivalent to the usual definition. A *parity game* G is given by a finite directed graph with vertices V and edges E satisfying the following properties. The set V is the disjoint union of two sets V_0 and V_1 which we think of as the vertices belonging respectively to player 0 and to player 1. The graph has a designated *start vertex* s, and every vertex has at least one outgoing edge. We identify V with the interval $[n] = \{0, \ldots, n-1\}$ where $n = |V|$. Below when we talk about the "least" vertex we mean the least with respect to the usual order on $[n]$. Without loss of generality, $s = 0$.

The game begins with a pebble placed on the start vertex s. On each turn, the pebble is moved from its current vertex v along an edge in the graph. If $v \in V_0$ then player 0 chooses which edge to move it along. If $v \in V_1$ then player 1 chooses. A *play* of the game is the infinite sequence $v_1, v_2, \ldots$ of vertices visited by the pebble. To decide the winner of a play, let v be the least vertex which occurs infinitely often. If $v \in V_0$ then player 0 wins and if $v \in V_1$ then player 1 wins.

A *positional strategy* σ for player 0 is a map $\sigma\colon V_0 \to V$ such that $(x, \sigma(x))$ is an edge in E for each $x \in V_0$. Similarly, a positional strategy τ for player 1 is a map $\tau\colon V_1 \to V$ such that $(x, \tau(x)) \in E$ for each $x \in V_1$.

The following theorem has been proven by Emerson [8] independently of a similar result for mean payoff games by Ehrenfeucht and Mycielski [7]; the reduction from parity to mean payoff games was found later by Puri [19].

Theorem 5 (Emerson [8]). *In each parity game, one of the players has a positional winning strategy.* □

From now on we will only discuss positional strategies, so we will usually omit the word "positional". Given a strategy σ for player 0, we will use E^σ to mean the edge relation obtained from E by, for each vertex $v \in V_0$, removing all outgoing edges except for the one chosen in σ. We will similarly use E^τ to mean E restricted by a strategy τ for player 1.

It is straightforward to show that the strategy σ is winning for player 0 if and only if for every vertex t reachable from s in E^σ, for every path from t to t in E^σ, the least vertex on the path is in V_0. To prove our main result in this section, we formalise this characterisation in such a way that we can prove in U_2-IND that player 0 and player 1 cannot simultaneously have winning strategies. In our formalisation below, all quantifiers are implicitly bounded by n.

Expand the language L to include relation symbols E, V_0, V_1, E^σ, $R^\sigma_{\min}$, E^τ, $R^\tau_{\min}$ and a constant symbol n. We will write G to stand for the tuple E, V_0, V_1, n representing the structure of the game. The intended meaning of E^σ is as described above. The intended meaning of the ternary relation $R^\sigma_{\min}(x, y, z)$ is that there is a non-trivial path in E^σ from x to y on which the least vertex visited is z. The relations E^τ and $R^\tau_{\min}$ are similar.

Let $\mathrm{Game}(G)$ be a formula asserting that G is a suitable graph for a parity game, that is, that V_0 and V_1 partition the vertices, and that every vertex has at least one outgoing edge. Let $\mathrm{Strategy}_0(G, E^\sigma)$ be a formula asserting that E^σ represents a strategy for player 0, that is, that every vertex in V_0 has an outgoing edge in E^σ. Let $\mathrm{Strategy}_1(G, E^\tau)$ be a similar formula for player 1. It is clear that these can all be written as U_2 formulas.

Let $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\min})$ be the conjunction of the universal closures of

1. $\mathrm{Strategy}_0(G, E^\sigma)$
2. $E^\sigma(x, y) \wedge z = \min(x, y) \to R^\sigma_{\min}(x, y, z)$
3. $R^\sigma_{\min}(x, y, u) \wedge R^\sigma_{\min}(y, z, v) \wedge w = \min(u, v) \to R^\sigma_{\min}(x, z, w)$
4. $R^\sigma_{\min}(s, x, u) \wedge R^\sigma_{\min}(x, x, v) \to v \in V_0$.

Let $\mathrm{Win}_1(G, E^\tau, R^\tau_{\min})$ be a similar formula for player 1.

Lemma 6. *If player* 0 *has a winning strategy in* G*, then there exist* E^σ *and* $R^\sigma_{\min}$ *satisfying* $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\min})$*. Similarly for player* 1 *and* $\mathrm{Win}_1(G, E^\tau, R^\tau_{\min})$. □

Theorem 7. *Provably in* U_2-IND*, it is impossible to satisfy formulas* $\mathrm{Game}(G)$, $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\min})$ *and* $\mathrm{Win}_1(G, E^\tau, R^\tau_{\min})$ *simultaneously.*

Proof. Let $R^*(x,y)$ be the formula $\exists v,\ R^\sigma_{\min}(x,y,v) \wedge R^\tau_{\min}(x,y,v)$. By condition 3 of Win_0 and Win_1, the relation $R^*(x,y)$ is transitive. Moreover for every x there is at least one y such that $R^*(x,y)$, since we can take y to be the unique successor of x in $E^\sigma \cap E^\tau$ and take v to be $\min(x,y)$.

Let $A(x)$ be the formula $R^*(s,x) \wedge \forall y\!>\!x\, \neg R^*(x,y)$. Using E_1-MAX, let x be maximum such that $R^*(s,x)$. It follows that $A(x)$ holds. Hence using E_2-MIN, we let t be minimum such that $A(t)$. Now using E_1-MAX, let t' be maximum such that $R^*(t,t')$. By the transitivity of R^*, we know that $R^*(s,t')$ and also that for all $y > t'$ we have $\neg R^*(t',y)$. Hence $A(t')$ holds, and therefore $t' \geq t$ by minimality of t. On the other hand, since $A(t)$ and $R^*(t,t')$, we know $t' \leq t$. We conclude that $t' = t$.

We now have that $R^*(s,t)$ and $R^*(t,t)$. Hence there are vertices u and v such that both $R^\sigma_{\min}(s,t,u) \wedge R^\sigma_{\min}(t,t,v)$ and $R^\tau_{\min}(s,t,u) \wedge R^\tau_{\min}(t,t,v)$ hold. Therefore condition 4 must be false in either Win_0 or Win_1, since either $v \in V_0$ or $v \in V_1$. □

The formula $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\min})$ is a conjunction of U_2 formulas. Suppose we are given a parity game G, with n vertices. Let α map the constant symbol n of our language (which we treat here as a free variable) to the number n. Then for some $k \in \mathbb{N}$ we can translate each such formula ϕ into a conjunction $\langle\phi\rangle_\alpha$ of k-DNFs, with propositional variables for the relations E^σ, $R^\sigma_{\min}$ and for the structure of the game G. We abuse notation and write $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G$ for the propositional formula obtained by taking the set of all the formulas $\langle\phi\rangle_\alpha$ and substituting in, for the propositional variables describing the structure of G, the values given by the actual game G.

In other words, $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G$ is the propositional formula obtained by translating Win_0 and substituting in the real values of G. It is satisfiable if and only if player 0 has a winning strategy in G. The formula $\langle \mathrm{Win}_1(E^\tau, R^\tau_{\min})\rangle_G$ is similar.

Corollary 8. *There is a number $k \in \mathbb{N}$ and a polynomial p such that for every game G, the formula $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G \cup \langle \mathrm{Win}_1(E^\tau, R^\tau_{\min})\rangle_G$ has a $\mathrm{Res}(k)$ refutation of size $p(n)$.*

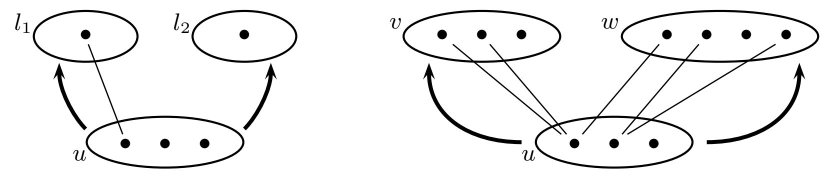
Proof. Take the proof given by Theorem 3, and substitute in the real values of G. Observe that G satisfies $\mathrm{Game}(G)$, so all the initial formulas coming from $\mathrm{Game}(G)$ vanish. □

Corollary 9. *The canonical pair for parity games is reducible to the canonical pair for resolution.*

Proof. Let p and k be as in Corollary 8. By Proposition 4, it is enough to show reducibility to the canonical pair for $\mathrm{Res}(k)$. The reduction function is given by

$$G \mapsto (\, \langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G,\ 1^{p(n)}\,).$$

If player 0 has a winning strategy for G then $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G$ is satisfiable. On the other hand, if player 1 has a winning strategy for G then $\langle \mathrm{Win}_1(E^\tau, R^\tau_{\min})\rangle_G$



Vertex u connected to leaves l_1 and l_2 with points and lines

Non-leaf vertices with points and lines

Fig. 1. Components of point-line game graphs.

is satisfiable, and substituting the satisfying assignment into the Res(k) refutation from Corollary 8 yields the required refutation of $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G$ of size $p(n)$. □

Corollary 10. *If resolution is weakly automatizable, then parity games can be decided in polynomial time.* □

3 A Game Equivalent to Resolution

In this section we will define the *point-line game* and prove the following:

Theorem 11. *The canonical pair for the point-line game is equivalent to the canonical pair for resolution.*

An instance of the point-line game is given by a finite directed acyclic graph (V, E) with some extra structure. Namely, the set V is the disjoint union of sets V_0, V_1 and F, where vertices in V_0 and V_1 belong respectively to player 0 and player 1, and F contains exactly the leaf vertices, that is, those of out-degree 0. There is a designated start vertex s of in-degree 0. Each vertex v contains a set S_v of *points*. The start vertex is empty (contains no points) and every leaf contains exactly one point. Vertices do not share points. If there is an edge (u, v) in E, then some points in u may be connected to some points in v by *lines*. A point in u may have lines out to many points in v, but each point in v has a line in from at most one point in u, as in Fig. 1. During the game some points will be assigned *colours*, either black, for player 0, or white, for player 1.

The game starts with a pebble on s. At the beginning of a general turn, the pebble is on some vertex u and every point in u has a colour. As before, the player who owns vertex u moves the pebble along an outgoing edge to a new vertex v. Every point p in v that is connected by a line to some point q in u is then coloured with q's colour. Every other point in v is coloured with the colour of the player who did not move. The game ends when the pebble reaches a leaf w. The winner is the player whose colour is on the single point in w.

As before, a *positional strategy* is a function $\sigma\colon V_0 \to V$ or $\tau\colon V_1 \to V$ assigning a choice of outgoing edge to each of a player's vertices, regardless of the history of the game or the colouring of the current vertex. However in this case, it is not in general true that a winning strategy exists if and only if a positional winning strategy exists. One can give an example of such a game in which neither player has a positional winning strategy, while at the same time one of the players must, as in any finite game, have a (non-positional) winning strategy.

Lemma 12. *Given such a game G and a positional strategy σ for player* 0*, it is decidable in polynomial time whether σ is a winning strategy. Hence the canonical pair for point-line games is a disjoint* NP *pair.*

Proof. We describe a polynomial time algorithm which, working backwards from the leaves, labels each vertex u with either a set $B_u \subseteq S_u$ of points or a symbol "Losing$_0$". This labelling will have the property that if u is labelled "Losing$_0$" then, regardless of the colouring of u, if the pebble reaches u then player 1, playing optimally, will win the game if player 0 plays according to σ. If u is not labelled "Losing$_0$" then if player 0 plays according to σ and player 1 plays optimally, player 0 will win the game from u if and only if all points in B_u are coloured black. Thus σ is a winning strategy for player 0 if and only if the start vertex s is not labelled "Losing$_0$".

The algorithm labels a vertex u using the following rules.

1. If u is a leaf, set B_u to be the (unique) point in u.
2. If $u \in V_1$, suppose that u has children $v_1, \dots, v_k$ and that these have all been labelled. If any child v_i is labelled "Losing$_0$", then label u as "Losing$_0$". Otherwise, let B_u contain every point in u which is connected by a line to some point in B_{v_i} for some child v_i (in other words, let B_u be the union of the pre-images of the sets B_{v_i}).
3. If $u \in V_0$, let $v = \sigma(u)$. Suppose that v has been labelled. If v is labelled "Losing$_0$" then label u as "Losing$_0$". If not, there are two possibilities. If there is a point in B_v that is not connected by a line to any point in u, label u as "Losing$_0$". Otherwise, let B_u be the set of points of u which are connected by a line to some point in B_v. □

Theorem 13. *The canonical pair for the point-line game is reducible to the canonical pair for* $\mathrm{Res}(k)$ *for some $k \in \mathbb{N}$, and hence to the canonical pair for resolution by Proposition 4.*

Proof. (Sketch.) We can write a formula Win_0 which is satisfiable if and only if there is a strategy σ for player 0 and a corresponding labelling of the graph, as in the previous lemma, in which no leaf reachable from s under σ is labelled "Losing$_0$". We can write a similar formula Win_1 wrt. a strategy τ for player 1 and a corresponding labelling. The proof that Win_0 and Win_1 cannot be satisfied simultaneously is then essentially a proof that the labelling algorithm works. We prove, working from the leaves of the graph down to s, that if any node v is reachable from s under both σ and τ, then $B^\sigma_v \cap W^\tau_v$ is non-empty, where W^τ is player 1's version of the relation B^σ and represents points that must be coloured white for player 1 to win using strategy τ. This gives a contradiction when we reach s, which contains no points.

This argument formalises as a U_2 induction (we also need to add relations R^σ and R^τ, for reachability under σ and τ, respectively to Win_0 and Win_1, as in the previous section). Thus, it translates into a $\mathrm{Res}(k)$ refutation, which gives us our result, as in Corollaries 8 and 9. $\square$

The other direction of Theorem 11 can be proven by showing that the interpolation pair for PK_1, which is equivalent to the canonical pair for resolution by Proposition 4, is reducible to the canonical pair for the game.

Theorem 14. *The interpolation pair of* PK_1 *is reducible to the canonical pair for the point-line game.*

Proof. (Sketch.) Starting from a PK_1-refutation of two sets of clauses Φ and Ψ in disjoint sets of variables X and Y, we can construct in polynomial time a game G such that if Φ is satisfiable then player 0 has a positional winning strategy in G, and if Ψ is satisfiable then player 1 has such a strategy.

The game has one vertex for each DNF that forms a line in the proof, and that vertex contains one point for each conjunction in the DNF. Additionally it has one leaf vertex for each literal z arising from a variable in $X \cup Y$. Each such leaf vertex contains a single point.

The structure of the game is similar to that of the proof. The edges reflect the structure of the proof, and two points are connected by a line if the corresponding conjunctions stand in a natural direct ancestor relation. The vertices corresponding to cut and $\wedge$-introduction rules belong to player 0 if an X variable is involved in the rule, and to player 1 if it is a Y variable. Vertices corresponding to clauses from Φ belong to player 0, similarly for Ψ and player 1.

The game is constructed so that the following is true. Suppose player 0 knows an assignment A to the X variables that satisfies Φ. Then he can use A to make choices in the game guaranteeing that, whenever the pebble moves to a non-leaf vertex u, then for every point p in u which corresponds to a conjunction whose X-literals are all satisfied by A, p gets coloured black. This means that when the game reaches a node corresponding to an initial clause of the proof, then if the clause is from Φ at least one point will be black, and if it is from Ψ then all the points will be black. Either way, player 0 will win. We have the symmetrical property for player 1. $\square$

A question motivated by our results is to find a direct reduction of parity games to point-line games with positional strategies. Using such a reduction one may be able to define a subclass of point-line games that always have positional strategies, for which one could try to find a polynomial time algorithm instead of working directly with parity games.

References

1. Alekhnovich, M., Razborov, A.A.: Resolution is not automatizable unless W[P] is tractable. SIAM J. Comput. 38(4), 1347–1363 (2008)

2. Atserias, A., Bonet, M.L.: On the automatizability of resolution and related propositional proof systems. Inform. and Comput. 189(2), 182–201 (2004)
3. Atserias, A., Maneva, E.: Mean-payoff games and propositional proofs. Inform. and Comput. 209(4), 664–691 (2011)
4. Buresh-Oppenheim, J., Morioka, T.: Relativized NP search problems and propositional proof systems. In: Proceedings of the 19th IEEE Annual Conference on Computational Complexity. pp. 54–67. IEEE (2004)
5. Buss, S.R.: Bounded arithmetic, Studies in Proof Theory. Lecture Notes, vol. 3. Bibliopolis, Naples (1986)
6. Condon, A.: On algorithms for simple stochastic games. In: Advances in computational complexity theory (New Brunswick, NJ, 1990), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 13, pp. 51–71. Amer. Math. Soc., Providence, RI (1993)
7. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Internat. J. Game Theory 8(2), 109–113 (1979)
8. Emerson, E.A.: Automata, tableaux, and temporal logics. In: Logics of programs (Brooklyn, N.Y., 1985), Lecture Notes in Comput. Sci., vol. 193, pp. 79–88. Springer, Berlin (1985)
9. Friedmann, O.: An exponential lower bound for the latest deterministic strategy iteration algorithms. Log. Methods Comput. Sci. 7(3), 3:19, 42 (2011)
10. Friedmann, O.: Recursive algorithm for parity games requires exponential time. RAIRO Theor. Inform. Appl. 45(4), 449–457 (2011)
11. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, logics, and infinite games, Lecture Notes in Computer Science, vol. 2500. Springer-Verlag, Berlin (2002)
12. Huang, L., Pitassi, T.: Automatizability and simple stochastic games. (Revised version – an earlier version appeared in Automata, languages and programming, Part I, vol. 7655 of Lecture Notes in Comput. Sci., Springer, Heidelberg, 2011, pp. 605–607)
13. Juba, B.: On the Hardness of Simple Stochastic Games. Master's thesis, Carnegie Mellon University (2005)
14. Krajíček, J.: Lower bounds to the size of constant-depth Frege proofs. J. Symbolic Logic 59, 73–86 (1994)
15. Krajíček, J.: Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. J. Symbolic Logic 62, 457–486 (1997)
16. Krajíček, J.: On the weak pigeonhole principle. Fund. Math. 170(1-2), 123–140 (2001)
17. Paris, J., Wilkie, A.: Counting problems in bounded arithmetic. In: Methods in mathematical logic (Caracas, 1983), Lecture Notes in Math., vol. 1130, pp. 317–340. Springer, Berlin (1985)
18. Pudlák, P.: On reducibility and symmetry of disjoint NP pairs. Theoret. Comput. Sci. 295(1-3), 323–339 (2003)
19. Puri, A.: Theory of hybrid systems and discrete event structures. Ph.D. thesis, University of California, Berkeley (1995)
20. Razborov, A.A.: On provably disjoint NP-pairs. Tech. Rep. RS-94-36, Basic Research in Computer Science Center, Aarhus, Denmark (November 1994)
21. Stirling, C.: Modal and Temporal Properties of Processes. Texts in Computer Science, Springer (2001)
22. Wilmers, G.: Bounded existential induction. J. Symbolic Logic 50(1), 72–90 (1985)
23. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theoret. Comput. Sci. 158(1-2), 343–359 (1996)