Electronic Colloquium on Computational Complexity, Report No. 92 (2013)

Parity Games and Propositional Proofs

Arnold Beckmann*
Department of Computer Science
College of Science
Swansea University
Swansea SA2 8PP, UK
a.beckmann@swansea.ac.uk

Pavel Pudlák*† and Neil Thapen*†
Institute of Mathematics
Academy of Sciences of the Czech Republic
Žitná 25, 115 67 Praha 1, Czech Republic
{pudlak,thapen}@math.cas.cz

June 18, 2013

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 give simple proofs that the same holds for depth-1 propositional calculus (where resolution has depth 0) with respect to mean payoff and simple stochastic games. We define a new type of 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.

Our main technique is to show that a suitable weak bounded arithmetic theory proves that both players in a game cannot simultaneously have a winning strategy, and then to translate this proof into propositional form.

*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

ISSN 1433-8092

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.

Parity games have important applications in automata theory, logic, and verification [17]—for example, the model checking problem for the modal μ-calculus is polynomial time equivalent to solving parity games [14]. To the best of our knowledge, they originated in the study of the non-emptiness problem for parity automata, a notion to which they are equivalent [14]. Mean payoff games were introduced by Ehrenfeucht and Mycielski [12], and are useful in design and analysis for various on-line problems [33]. Condon initiated the study of simple stochastic games for analysing randomised space-bounded alternating Turing machines [10]. They are restrictions of stochastic games, which were introduced by Shapley [29].

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 [27, 33]. 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, which are believed to be incomparable [11, 19, 6]. 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 [15, 16].

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] 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 in polynomial time, 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 [18]. In this paper we extend and simplify these results.

In Section 2 below we show that if resolution is weakly automatizable, then parity games can be decided in polynomial time.

In Section 3 we give a proof that if PK_1 is weakly automatizable, then mean payoff games can be decided in polynomial time. This is the main result of [3], and also follows from [18] or from our Section 4 by the reducibility of mean payoff games to simple stochastic games. However these proofs are rather indirect. Here we give a direct, natural proof of this result, using the approach from the previous section and the development of binary arithmetic from Appendix B.

In Section 4 we show a similar result for PK_1 and simple stochastic games. This is the main result of [18] but again we hope that our proof is simpler.

Finally in Section 5 we define a new game, the *point-line game*, about moving a token around a finite graph. We show that its complexity is equivalent to that of resolution, in a certain sense. Namely, 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 Sections 2, 3 and 4, together with one direction of Section 5, 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 [25]. 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 due to Krajíček [22, 20, 21]; however as far as we know there is no paper with a self-contained presentation of the proof (and our theory is slightly different). For these reasons we include our own presentation of these translations as Appendix A.

The remaining parts of Section 1 contain some necessary preliminaries about propositional proofs, bounded arithmetic, disjoint NP pairs, and binary arithmetic. The technical details of our formalisation of binary arithmetic are in Appendix B.

Our main new result is the reduction of the decision problem for parity games to the weak automatizability of resolution. Finding a polynomial time algorithm to solve parity games is a long-standing open problem, so our result can be viewed as evidence that either resolution is not weakly automatizable, or 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. A first step towards investigating the feasibility of this would be to determine the hardness, as defined by Kullmann [24], of the set of clauses produced by our reduction.

Finally, a natural question is whether there is a converse to the results of Sections 2 and 4. That is, is it the case that polynomial time decidability of parity games implies weak automatizability of resolution, or that polynomial time decidability of simple stochastic games implies weak automatizability of PK_1? We are inclined to think that the answer is no, in both cases.

In the case of parity games and resolution, all we can say is that we do have a kind of converse in Section 5, but we conjecture that the point-line games defined there are not reducible to parity games, one reason being that the canonical pair of a parity game consists of an NP set and its complement (also an NP set), while the canonical pair of a point-line game usually does not have this property.

In the case of of simple stochastic games and PK_1, we can say a little more. The result in Section 4 about simple stochastic games only needs as much of U_3-IND (and hence of the proof system PK_1 – see below) as suffices to prove that every Δ_2 ordering of a bounded set has a least element. We have an indirect reason to believe that this is not the full strength of U_3-IND. Namely, the recent paper [9] considers an analogous ordering principle, that every polynomial-time ordering of a bounded set has a least element, and shows that this is provable not only in the theory T^2_2 but also in the theory T^1_2+sWPHP(P^{NP}), which extends T^1_2 by a kind of approximate counting and is presumably incomparable with T^2_2. Hence the polynomial-time ordering principle is unlikely to be as strong as T^2_2, and we may expect a similar situation with the Δ_2 ordering principle in U_3-IND.

1.1 Constant depth proof systems

The propositional proof system PK is defined as follows. The formulas of PK 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 and negated variables 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 $\bigvee\Phi \vee \bigvee\Psi = \bigvee(\Phi \cup \Psi)$. The *depth* $\mathrm{dp}(\varphi)$ of a formula φ is the maximal nesting of $\bigwedge$ and $\bigvee$ in φ. Thus literals have depth 0, and $\mathrm{dp}(\bigwedge\Phi) = \mathrm{dp}(\bigvee\Phi) = 1 + \max_{\varphi\in\Phi} \mathrm{dp}(\varphi)$.

Each line in a PK-proof is a disjunction, sometimes called a *cedent*, usually written as the list of disjuncts separated by commas. The rules of PK are as follows, where Γ, Δ stand for sets of formulas, possibly empty:

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

$$\text{weakening} \quad \frac{\Gamma}{\Gamma, \Delta} \qquad \text{cut} \quad \frac{\Gamma, A_1 \wedge \ldots \wedge A_m \qquad \Gamma, \neg A_1, \ldots, \neg A_m}{\Gamma}$$

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

We will also sometimes use "formula" to refer to a set of disjunctions. Semantically, this behaves the same as the conjunction of those disjunctions. A PK *refutation* of such 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.

We will write PK_d for the sub-system of PK in which every formula in a cedent has depth d (or less), and $\mathrm{PK}_{d,k}$ for the system in which the formulas have depth $d+1$, but where all gates at depth d have fan-in at most k. The system PK_0 is called *resolution* and is denoted by $\mathrm{Res}(1)$ or simply Res. The system $\mathrm{PK}_{0,k}$, for $k \geq 2$, is denoted by $\mathrm{Res}(k)$.

There is obviously a potential confusion in what is meant by a depth-d proof system, since in each case each line of a proof, viewed as a single disjunction, is of depth one level higher than the formulas occurring in it. We will try to avoid this by always explicitly referring to a system as resolution, $\mathrm{Res}(k)$, PK_d or $\mathrm{PK}_{d,k}$.

1.2 Bounded arithmetic

We could obtain the results of this paper by a careful use of the conventional Buss-style bounded arithmetic theories T^2_1 and T^3_1 [7] augmented with uninterpreted predicate and relation symbols, which can be viewed as second order variables. However, this would introduce unnecessary complications to deal with sharply bounded quantification. Instead we will work with simpler systems. Furthermore, we use our theories in such a way that only the complexity of formulas with the new predicate and relation symbols matters, allowing us to include all true arithmetical formulas as axioms. Therefore, rather than describing a modification of Buss's theories, we give a new definition.

For $r \in \mathbb{N}$, we will say that a function $f : \mathbb{N}^r \to \mathbb{N}$ 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 [32], 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. We add a further technical requirement, to make sure that our translation of these formulas into propositional form works smoothly: the quantifier-free part must have the form of a CNF if d is odd, or a DNF if d is even. However we emphasise that, since any quantifier-free formula is logically equivalent to one in either form, this requirement can usually be ignored in practice and we will ignore it in the first-order proofs we use for our main results in this paper. 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 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 non-empty U_d subset (with parameters) 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. We will call a relation or formula Δ_d if in the model under consideration it is expressible both by an E_d and by a U_d formula. We will say that a formula is Δ_d over a theory if the theory proves that the E_d and U_d versions are equivalent.

The following is proved in the same way as the corresponding principles in the usual bounded arithmetic theories.

Lemma 1.1. *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*,* U_d-MAX *and* E_{d+1}-MAX *are equivalent.*

Furthermore if $d \geq 1$, then U_{d+1}-IND *proves that every Δ_d partial ordering on a bounded interval has a least element.* □

We now give our version of the Paris-Wilkie translation of first-order proofs in bounded arithmetic into small propositional proofs [25]. Our goal is to construct $\mathrm{PK}_{d,k}$ refutations out of U_{d+2}-IND proofs.

For an L^+ formula ϕ, and an assignment α to the free variables of ϕ, we will define a PK formula $\langle\phi\rangle_\alpha$. This translation will evaluate L formulas as true or false, translate atomic formulas about the relations $\bar{R}$ into propositional variables, translate propositional connectives as themselves, and turn bounded quantifiers $\forall$ and $\exists$ into respectively $\bigwedge$ and $\bigvee$. For each relation symbol in $\bar{R}$ of arity s, we fix a propositional variable $r_{i_1,\dots,i_s}$ for each tuple of numbers $i_1, \dots, i_s$. We assume that all these propositional variables, for all relation symbols in $\bar{R}$, are pairwise distinct.

Before giving the formal definition, we introduce some notation. We identify the empty set of formulas with a new symbol $\top$ (for the truth value *true*). We identify the set $\{\bigvee \emptyset\}$ containing just the empty disjunction with a new symbol $\bot$ (for *false*). 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 1.2. 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 $r_{i_1,\dots,i_s}$ where each $i_j = \langle t_j\rangle_\alpha$, and let $\langle\neg R(\bar{t})\rangle_\alpha$ be the negated variable $\neg r_{i_1,\dots,i_s}$.

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 $\top$.

Now let ϕ be a strict U_{d+2} formula, whose quantifier-free part is a k-CNF, if d is odd, and k-DNF, if d is even. Then the translation $\langle\phi\rangle_\alpha$ is a conjunction of disjunctions, where each disjunction is of depth $d+2$, but all of its gates at depth $d+1$ have fan-in at most k. We write $\langle\phi\rangle_\alpha^d$ for the set of these disjunctions. We must also distinguish the case where $\langle\phi\rangle_\alpha$ is simply $\top$ or $\bot$, where we let $\langle\phi\rangle_\alpha^d$ be respectively $\top$ or $\bot$.

The purpose of this extra step is to allow our translated formula to be usable in the propositional proof systems defined above, since $\langle\phi\rangle_\alpha^d$ now has exactly the form of a set of cedents of $\mathrm{PK}_{d,k}$. For example, let ϕ be the formula $\forall x\!<\!z\,\exists y\!<\!z\,R(x,y)\wedge R(y,x)$. Then $\langle\phi\rangle_{[z\mapsto n]}$ is the formula $\bigwedge_{i<n}\bigvee_{j<n} r_{ij}\wedge r_{ji}$. On the other hand $\langle\chi\rangle^0_{[z\mapsto n]}$ is the set of disjunctions $\{\bigvee_{j<n} r_{ij}\wedge r_{ji} : i<n\}$ which is in the right form to, for example, appear as the set of initial cedents in a Res(2) derivation.

Theorem 1.3. *Let $d\in\mathbb{N}$, with $d\geq 0$. Suppose that $\phi_1(x),\dots,\phi_\ell(x)$ are strict U_{d+2} formulas, with x the only free variable, such that U_{d+2}-IND proves $\forall x\,\neg(\phi_1(x)\wedge\ldots\wedge\phi_\ell(x))$. Then for some $k\in\mathbb{N}$ the family*

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

has polynomial size $\mathrm{PK}_{d,k}$ refutations.

Furthermore, we can take k to be the maximum k such that the quantifier-free parts of the formulas ϕ_i and the induction formulas used in the proof translate into k-DNFs if d odd, or k-CNFs if d is even. □

The cases we will need for the main results of this paper are $d=0$ and $d=1$. We will not need the part of the theorem giving extra information about k (because the canonical pair of $\mathrm{PK}_{d,k}$ is independent of k, up to polynomial equivalence—see below).

We remark that the theorem, as written, does not give optimally tidy results when the formulas we are refuting are of lower complexity than the induction formulas used in the proof. For example, in Sections 3 and 4 we will give refutations of U_2 formulas in U_3-IND. A U_2 formula translates naturally into a set of k-DNFs, but to apply the theorem we must consider these rather as U_3 formulas (by padding), which translate into something messier. For our applications in this paper, this does not matter. Furthermore this is an easy thing to fix, as the equivalence between the direct translation of such formulas and the padded version has short derivations.

We give a self-contained proof of Theorem 1.3 in Appendix A.

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 [28]. Our presentation follows [26]. 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

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

We say that $\mathcal{P}$ is *weakly automatizable* if the canonical pair of $\mathcal{P}$ is polynomially separable. 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 θ and ϕ 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

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

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 1.4.

1. *The interpolation pair of* PK_0 *(resolution) is polynomially separable.*
2. *For every* $d \geq 0$*, the canonical pairs of the proof systems*

   $$\mathrm{PK}_d, \mathrm{PK}_{d,2}, \mathrm{PK}_{d,3}, \ldots$$

   are equivalent, and are also equivalent to the interpolation pairs of

   $$\mathrm{PK}_{d,2}, \mathrm{PK}_{d,3}, \mathrm{PK}_{d,4}, \ldots \quad \textit{and of} \quad \mathrm{PK}_{d+1}.$$

Proof These relations are well-known. For the sake of completeness we recall the ideas of their proofs. Feasible interpolation for resolution was proved in [21]. For part 2, let us denote polynomial reducibility between pairs by $\preceq$. The reductions $\mathbf{C}_{\mathrm{PK}_d} \preceq \mathbf{C}_{\mathrm{PK}_{d,k}}$ and $\mathbf{I}_{\mathrm{PK}_{d,k}} \preceq \mathbf{I}_{\mathrm{PK}_{d+1}}$ are trivial. The converse reductions are easy, using extension axioms. (Note that in the case of the interpolation pairs we may introduce extension axioms only for sets of the same kind of variable.) The reduction $\mathbf{I}_{\mathrm{PK}_{d,k}} \preceq \mathbf{C}_{\mathrm{PK}_{d,k}}$ is defined by mapping (ϕ, θ, π) to $(\phi, |\pi|)$ and observing that from π and a satisfying assignment for θ we obtain a refutation of ϕ by substituting the assignment and simplifying the refutation π. The only nontrivial reduction is $\mathbf{C}_{\mathrm{PK}_d} \preceq \mathbf{I}_{\mathrm{PK}_{d+1}}$, which is defined by mapping $(\phi, 1^m)$ to $(\phi, \rho_\phi^m, \pi_\phi^m)$, where ρ_ϕ is a formula that says that a string of length m of propositional variables encodes a PK_d refutation of ϕ, and π_ϕ^m is a PK_{d+1} refutation of $\phi \wedge \rho_\phi^m$. (This extends an argument of [2] for the case $d = 0$.) □

It is not known if any other pairs are polynomially separable. In [5] it is proven that for some small d_0 and all $d \geq d_0$, all pairs $\mathbf{I}_{\mathrm{PK}_d}$ are not polynomially separable, assuming that factoring Blum integers or computing the Diffie-Hellman function is sufficiently hard.

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.

Essentially all our results are based on the following observation, which we state as a lemma.

Lemma 1.5. *Suppose that, for a class of games, we can construct in polynomial time for every game G from the class a pair of propositional formulas* Win_0 *and* Win_1 *in disjoint variables such that each* Win_i *is satisfiable if player i has a positional winning strategy. Suppose also that there exists a polynomial size refutation of* $\mathrm{Win}_0 \cup \mathrm{Win}_1$ *in a proof system $\mathcal{P}$. Then the canonical pair of the class of games is polynomially reducible to the canonical pair of $\mathcal{P}$.*

In particular, if we can construct polynomial size $\mathrm{PK}_{d,k}$ *refutations of* $\mathrm{Win}_0 \cup \mathrm{Win}_1$ *for some k, then we get a polynomial reduction to the canonical pair of* PK_d.

Proof Let p be the polynomial bound on the size of the refutation. The reduction of the canonical pair of the game to the canonical pair of the proof system is given by the map

$$G \mapsto (\mathrm{Win}_0, 1^p) \ .$$

The second part of the lemma follows from Proposition 1.4. □

Remark. We remark that an alternative proof is to observe that the refutation π of $\mathrm{Win}_0 \cup \mathrm{Win}_1$ which we construct in Theorem 1.3 is actually constructible in polynomial time, and that the mapping $G \mapsto (\mathrm{Win}_0, \mathrm{Win}_1, \pi)$

is thus a polynomial reduction from the canonical pair of the class of games to the interpolation pair of $\mathcal{P}$. In particular, if we can construct in polynomial time PK_{d+1} refutations of $\mathrm{Win}_0 \cup \mathrm{Win}_1$, then we also get a polynomial reduction to the canonical pair of PK_d, using Proposition 1.4.

An important observation is that the formulas Win_i are not limited to directly describing player i's strategy, with propositional variables only for that strategy. We are free to add any variables we like, and to say about them anything we like, as long as we stay inside the logical complexity allowed by the proof system (which is no real limitation, as we can add extension variables). We take advantage of this to add variables and clauses that will make the refutation of $\mathrm{Win}_0 \cup \mathrm{Win}_1$ easier. For example, in the case of parity games, Win_0 has extra variables and clauses describing a certain reachability relation $R^{\sigma}_{\min}(x, y, z)$ arising from player 0's strategy σ. What we cannot do is add new variables that depend both on player 0's and on player 1's strategies.

Remark. We should note that in this paper we are concerned with *polynomial* reductions and separations and with proofs of polynomial length. In particular, for disjoint NP pairs, we will simply write *reducible* or *equivalent* rather than polynomially reducible or polynomially equivalent. We do not consider here the many natural and important questions about *quasi-polynomial* reductions and proofs, in particular concerning the proof system Res(log) [22] and the usual systems S^i_2 and T^i_2 of bounded arithmetic. We remark however that the system of narrow resolution, which would sit at around the level PK_{-1} in our hierarchy, is known to be quasi-polynomially automatizable [4].

1.4 Binary arithmetic

In Sections 3 and 4, refuting $\mathrm{Win}_0 \cup \mathrm{Win}_1$ will require reasoning about basic binary arithmetic, that is, about sums and ordering of n-bit numbers whose bits are given by n propositional variables (in the propositional setting) or by an oracle (in the first-order setting). For their results about mean payoff games, the authors of [3] develop a very sophisticated family of fixed depth formulas (disjunctions of k-CNFs, for a constant k) to sum constantly many binary numbers, and show their properties are provable in $\mathrm{PK}_{1,k}$. Similar formulas are used in [18] for simple stochastic games.

We show that, for our purposes, this complicated construction is not necessary. This is because the binary numbers we reason about always fall into two disjoint sets, those arising from player 0's strategy and those arising from player 1's strategy. While we will have to compare numbers from one side with numbers from the other, in our proofs we will never have to consider *sums* in which we mix the two sides together. So for every tuple of player 0's numbers which we will need to sum together in a proof, we can add variables to Win_0 expressing the value of the sum, along with formulas expressing that this value is calculated correctly. We will usually also need to add some extra variables to witness the intermediate steps of the calculation.

We do our formalisation of binary arithmetic in the first-order setting. We use $n+1$ bits to represent an integer in the range $[-2^n, 2^n)$ in two's complement form. Everything necessary can be formalised straightforwardly in U_2-IND, with no surprises. We include details in Appendix B, and summarise below the properties we need.

Proposition 1.6. *Over* U_2-IND*,*

1. *The usual ordering* $\leq$ *on such integers is provably a* Δ_2 *linear order.*
2. *There is a* U_2 *formula* Sum *such that for integers* X, Y, Z *with* $X, Y \in [-2^{n-1}, 2^{n-1})$*, we have* $X + Y = Z$ *if and only if there is a string* C *such that* $\mathrm{Sum}(X, Y, Z, C)$*.*
3. *Provably, for integers* X, Y, Z, U, V, W *with* $X, Y, U, V \in [-2^{n-1}, 2^{n-1})$*, if* $X \leq U$ *and* $Y \leq V$*, then* $\mathrm{Sum}(X, Y, Z, C)$ *and* $\mathrm{Sum}(U, V, W, D)$ *implies* $Z \leq W$*.* □

2 Parity games

Following Stirling [31] 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 $\sigma : V_0 \to V$ such that $(x, \sigma(x)) \in E$ for each $x \in V_0$. Similarly, a positional strategy τ for player 1 is a map $\tau : V_1 \to V$ such that $(x, \tau(x)) \in E$ for each $x \in V_1$.

The following theorem has been proven by Emerson [13] independently of a similar result for mean payoff games by Ehrenfeucht and Mycielski [12]; the reduction from parity to mean payoff games was found later by Puri [27].

Theorem 2.1 (Emerson [13]). *A player has a winning strategy in a parity game if and only if he 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^σ_{min}, E^τ, R^τ_{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 $R^\sigma_{\mathrm{min}}(x, y, z)$ 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^τ_{min} are similar.

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

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

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

Lemma 2.2. *If player* 0 *has a winning strategy in game G, then there exist E^σ and R^σ_{min} satisfying* $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\mathrm{min}})$. *Similarly for player* 1 *and* $\mathrm{Win}_1(G, E^\tau, R^\tau_{\mathrm{min}})$. □

The converse of Lemma 2.2 is also true. For suppose that we can satisfy $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\mathrm{min}})$, but that player 0 does not have a winning strategy. Then by Theorem 2.1, player 1 must have a winning strategy. Hence by Lemma 2.2, we can also satisfy $\mathrm{Win}_1(G, E^\tau, R^\tau_{\mathrm{min}})$. But by Theorem 2.3 below, we cannot satisfy both.

Theorem 2.3. *Provably in* U_2-IND*, it is impossible to satisfy* $\mathrm{Game}(G)$*,* $\mathrm{Win}_0(G, E^\sigma, R^\sigma_{\min})$ *and* $\mathrm{Win}_1(G, E^\tau, R^\tau_{\min})$ *simultaneously.*

Proof We first describe an informal proof. In the graph $E^\sigma \cap E^\tau$, if we start from s we will eventually reach some vertex t which is on a loop. Let v be the least vertex on this loop. Then we must have $\exists u\, R^\sigma_{\min}(s,t,u)$, $\exists u\, R^\tau_{\min}(s,t,u)$, $R^\sigma_{\min}(t,t,v)$ and $R^\tau_{\min}(t,t,v)$. Hence condition 4 is false in Win_0 if $v \in V_1$ and false in Win_1 if $v \in V_0$.

We cannot use this argument directly in U_2-IND, because we are not able in general to define the reachability relation on $E^\sigma \cap E^\tau$. Instead, 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 $\min(x,y)$.

We will use $R^*(x,y)$ as an approximation of the reachability relation on $E^\sigma \cap E^\tau$ and, as in the informal proof, find a vertex t that is both on a loop and reachable from s, in this approximate sense. 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 2.4. *There is a number $k \in \mathbb{N}$ and a polynomial p such that for every game G, the set of formulas $\langle \mathrm{Win}_0(E^\sigma, R^\sigma_{\min})\rangle_G \cup \langle \mathrm{Win}_1(E^\tau, R^\tau_{\min})\rangle_G$ has a $\mathrm{Res}(k)$ refutation of size $p(n)$.*

Proof Take the refutation given by Theorem 1.3 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. □

Applying Lemma 1.5 to Corollary 2.4 yields the following

Corollary 2.5. *The canonical pair for parity games is reducible to the canonical pair for resolution.* □

Corollary 2.6. *If resolution is weakly automatizable, then parity games can be decided in polynomial time.* □

3 Mean payoff games

A *mean payoff game* G is given by a finite directed graph (V, E) where $V = [n]$ is the disjoint union of sets V_0 and V_1 belonging respectively to player 0 and player 1, there is a designated start vertex s and each vertex has at least one outgoing edge. Furthermore each edge (x, y) is now assigned an integer weight $w(x, y)$ (written as a binary string). The rules for moving the pebble are the same as for a parity game. To decide the winner, let $\nu = \liminf_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} w(v_i, v_{i+1})$ where $v_1, v_2, \dots$ is the infinite sequence of vertices visited by the pebble. If $\nu \geq 0$ then player 0 wins and if $\nu < 0$ then player 1 wins. Strategies and positional strategies are defined as for parity games, and by the following theorem we will again usually omit the word "positional".

Theorem 3.1 (Ehrenfeucht and Mycielski [12]). *A player has a winning strategy in a mean payoff game if and only if he has a positional winning strategy.* □

Given a strategy σ for player 0, and vertices x and y such that y is reachable from x in E^σ, let $u^\sigma_{\inf}(x, y) \in \{-\infty\} \cup \mathbb{Z}$ be the infimum, over all non-trivial paths π from x to y in E^σ, of the total weight of π. We claim that if $u^\sigma_{\inf}(x, y) > -\infty$ then $u^\sigma_{\inf}(x, y) \geq Mn$, where $M < 0$ is a lower bound on the weight of the edges. This is because if π has no loops, then Mn bounds the weight of π; removing loops of positive weight does not increase the weight of π; and if π has a loop of negative weight, then paths exist with arbitrarily low weights, so $u^\sigma_{\text{inf}}(x, y) = -\infty$.

A strategy σ for player 0 is winning if and only if, for every vertex t reachable from s in E^σ, $u^\sigma_{\text{inf}}(t, t) \geq 0$. Similarly given a strategy τ for player 1 we can define $u^\tau_{\text{sup}}(x, y)$ and show that τ is winning if and only if, for every vertex t reachable from s in E^τ, $u^\tau_{\text{sup}}(t, t) < 0$.

Expand the language L to include a tuple G of relation symbols E, V_0, V_1, W and a constant symbol n, together describing the game. We add relation symbols E^σ, R^σ, and U^σ_{inf} for player 0's strategy and some relations derived from it. Here the intended meaning of $R^\sigma(x, y)$ is that y is reachable from x in E^σ, and the intended meaning of $U^\sigma_{\text{inf}}(x, y, i)$ is the ith bit of the binary number $u^\sigma_{\text{inf}}(x, y)$ (we also reserve one bit to express whether $u^\sigma_{\text{inf}}(x, y)$ is infinite). We further add a collection C^σ_U of relation symbols to code the computations of the sum $u^\sigma_{\text{inf}}(x, y) + u^\sigma_{\text{inf}}(y, z)$ for all triples x, y, z. We similarly add relation symbols E^τ, R^τ, U^τ_{sup} and C^τ_U for player 1.

Let Game(G) be a formula asserting that G is a suitable graph for a mean payoff game. Let Strategy$_0(G, E^\sigma)$ and Strategy$_1(G, E^\tau)$ be as before. Again, all three formulas are U_2 formulas. Let $\mathrm{Win}_0(G, E^\sigma, R^\sigma, U^\sigma_{\text{inf}}, C^\sigma_U)$ be the conjunction of the universal closures of:

1. Strategy$_0(G, E^\sigma)$
2. All sums used in the proof are computed correctly
3. $E^\sigma(x, y) \to R^\sigma(x, y) \wedge [u^\sigma_{\text{inf}}(x, y) \leq w(x, y)]$
4. $R^\sigma(x, y) \wedge R^\sigma(y, z) \to R^\sigma(x, z) \wedge [u^\sigma_{\text{inf}}(x, z) \leq u^\sigma_{\text{inf}}(x, y) + u^\sigma_{\text{inf}}(y, z)]$
5. $R^\sigma(s, x) \to [u^\sigma_{\text{inf}}(x, x) \geq 0]$.

Here condition 2 is a conjunction of U_2 formulas involving the formula Sum. Conditions 3, 4 and 5 are also U_2, since the ordering relation is provably Δ_2. Note that the formalisation of 4 does not involve Sum, but rather the part of C^σ_U which is stated in condition 2 to code the value of the sum $u^\sigma_{\text{inf}}(x, y) + u^\sigma_{\text{inf}}(y, z)$.

Let $\mathrm{Win}_1(G, E^\tau, R^\tau, U^\tau_{\text{sup}}, C^\tau_U)$ be a dual formula for player 1, with the ordering reversed in 3 and 4 and with 5 replaced by

5. $R^\tau(s, x) \to [u^\tau_{\text{sup}}(x, x) < 0]$.

Theorem 3.2. *Provably in* U_3-IND*, it is impossible to satisfy* Game(G), $\mathrm{Win}_0(G, E^\sigma, R^\sigma, U^\sigma_{\text{inf}}, C^\sigma_U)$ *and* $\mathrm{Win}_1(G, E^\tau, R^\tau, U^\tau_{\text{sup}}, C^\tau_U)$ *simultaneously.*

Proof Let $R^*(x, y)$ be the formula

$$R^\sigma(x, y) \wedge R^\tau(x, y) \wedge [u^\sigma_{\text{inf}}(x, y) \leq u^\tau_{\text{sup}}(x, y)].$$

Using the properties of binary arithmetic we can show that the relation R^* is transitive. Also, as before, for all x there exists some y for which $R^*(x, y)$.

It follows, by exactly the same argument as in the proof of Theorem 2.3, that there exists a vertex t such that $R^*(s, t) \wedge R^*(t, t)$. The only difference is that now the relation $R^*(x, y)$ is Δ_2 rather than E_1, so we need U_3-IND rather than U_2-IND. We conclude that $R^\sigma(s, t) \wedge R^\tau(s, t) \wedge [u^\sigma_{\mathrm{inf}}(t, t) \leq u^\tau_{\mathrm{sup}}(t, t)]$, violating condition 5 of either Win_0 or Win_1. □

Corollary 3.3 ([3]). *The canonical pair for mean payoff games is reducible to the canonical pair for* PK_1. *Hence if* PK_1 *is weakly automatizable, then we can decide the winner of a mean payoff game in polynomial time.*

Proof The proof is similar to the one given for parity games in Corollaries 2.4 and 2.5. □

4 Simple stochastic games

A *simple stochastic game* (SSG) G is given by a directed graph (V, E) satisfying the following properties. G has a designated *start vertex* s and two sink vertices called the 0-*sink* and the 1-*sink*. The set of non-sink vertices is the disjoint union of three sets V_{max}, V_{min}, V_{ave} called *max*, *min* and *average* vertices. All non-sink vertices have exactly two outgoing edges. As before we assume that $V = [n]$ and $s = 0$.

The game is played by putting a pebble on the start vertex, which is then moved along the edges of G by two players denoted player 1 or "Max", and player 0 or "Min". From a max vertex, player 0 chooses the outgoing edge to move the pebble along, and similarly for min vertices and player 1. At average vertices, the successor vertex is chosen at random with each of the two outgoing edges being chosen with probability $\frac{1}{2}$. Player 1 wins the play if the pebble reaches the 1-sink and player 0 wins if it reaches the 0-sink.

A *strategy* σ for player 0 is a map $\sigma \colon V_{\mathrm{min}} \to V$ such that $(i, \sigma(i)) \in E$ for all min vertices i. Similarly, a strategy τ for player 1 is a map $\tau \colon V_{\mathrm{max}} \to V$ such that $(i, \tau(i)) \in E$ for all max vertices i. We define the *value* $v_{\sigma,\tau}(i)$ *of vertex* i *with respect to strategies* σ *and* τ to be the probability that player 1 wins the game if the pebble begins on i and the players use strategies σ and τ. The *optimal value* $v_{\mathrm{opt}}(i)$ *of* G *at vertex* i is defined as $\max_\tau \min_\sigma v_{\sigma,\tau}(i)$. We define the *value* $\mathrm{val}(G)$ *of* G to be $v_{\mathrm{opt}}(s)$, the optimal value of the start vertex. The *SSG value problem* is to decide, given an SSG G, whether $\mathrm{val}(G) > \frac{1}{2}$.

For $\lambda \in \mathbb{R}$, say that a λ-*solution* of G is a vector $u \in [0, 1]^n$ satisfying, at all vertices i,

$$u(i) = \begin{cases} \lambda \max\{u(j), u(k)\} & \text{if } i \in V_{\max} \text{ and } iE = \{j,k\} \\ \lambda \min\{u(j), u(k)\} & \text{if } i \in V_{\min} \text{ and } iE = \{j,k\} \\ \frac{\lambda}{2}(u(j)+u(k)) & \text{if } i \in V_{\text{ave}} \text{ and } iE = \{j,k\} \\ 0 & \text{if } i \text{ is the 0-sink} \\ 1 & \text{if } i \text{ is the 1-sink} \end{cases}$$

where iE denotes the set of vertices which can be reached from i using an edge in G, that is, $iE = \{j \in V : (i,j) \in E\}$. If $\lambda = 1$ we will call u simply a *solution* of G. It is easy to see that the vector v_{opt} of optimal values of G is a solution of G. However, in general there may also be other solutions.

Proposition 4.1 ([29, 10])**.** *There is a constant c such that for any game G, if we let $m = cn$ and $\lambda = 1 - 2^{-m}$ then G has a unique λ-solution w. Furthermore if $\text{val}(G) \leq \frac{1}{2}$ then $w(s) \leq \frac{1}{2}$ and if $\text{val}(G) > \frac{1}{2}$ then $w(s) \geq \frac{1}{2} + \frac{1}{2} \cdot 4^{-N}$, where $N = n(m+1)$.* □

This follows immediately from the results in [10]. That paper expands G to a λ-*stopping game* G' with N vertices, where $\text{val}(G') > \frac{1}{2}$ if and only if $\text{val}(G) > \frac{1}{2}$. The vector w arises as the restriction of the optimal values of G' to the vertices in G. Since G' is an SSG, the optimal values of G' are rational numbers over a common denominator bounded above by 4^N.

Take λ, m and N as in the proposition. To obtain our reduction of SSGs to the weak automatizability of PK_1, we will prove in U_3-IND that it is impossible to simultaneously have a solution u of G with $u(s) \leq \frac{1}{2}$ and a λ-solution w of G with $w(s) \geq \frac{1}{2} + \frac{1}{2} \cdot 4^{-N}$. Mixing solutions with λ-solutions in this way is not essential, but makes our proof substantially simpler. We are also careful that our proof never involves the sum of a value from u and a value from w. Otherwise we follow [18] in formalising the argument in [10] that λ-stopping games have a unique solution.

Let $D = 2^{m+4}4^N$. Let $\text{Win}_0(G, U)$ express the following, where as above we use j and k to refer to the two neighbours of i:

1. All sums in u used in the proof are computed correctly
2. $u(i) \geq \max\{u(j), u(k)\}$, for $i \in V_{\max}$
3. $u(i) \geq \min\{u(j), u(k)\}$, for $i \in V_{\min}$
4. $2u(i) \geq u(j) + u(k) - 2$, for $i \in V_{\text{ave}}$
5. $u(\text{0-sink}) = 0$ and $u(\text{1-sink}) = D$
6. $u(s) \leq \frac{D}{2}$.

If $\mathrm{val}(G) \leq \frac{1}{2}$ then $\mathrm{Win}_0(G, U)$ can be satisfied by setting $u = \lfloor D \cdot v_{\mathrm{opt}} \rfloor$ for the vector v_{opt} of optimal values of G.

For $\mathrm{Win}_1(G, W)$, we would like to write something dual to $\mathrm{Win}_0(G, U)$, expressing that w has some of the useful properties of a λ-solution of G. For example, we might choose to write $w(i) \leq \lambda \max\{w(j), w(k)\}$. However, for the sake of simplicity we would rather avoid using any binary multiplication. It turns out that we will only be interested in vertices i with $w(i) \geq 2^{m+3}$, and for such i we have that, for any number a, if $w(i) \leq \lambda a$ then $w(i) \leq a - 8$ (recall that $\lambda = 1 - 2^{-m}$). For our purposes, this last property is enough. Hence we let $\mathrm{Win}_1(G, W)$ express the following:

1. All sums in w used in the proof are computed correctly
2. If $w(i) \geq 2^{m+3}$ then $w(i) \leq \max\{w(j), w(k)\} - 8$, for $i \in V_{\max}$
3. If $w(i) \geq 2^{m+3}$ then $w(i) \leq \min\{w(j), w(k)\} - 8$, for $i \in V_{\min}$
4. If $w(i) \geq 2^{m+3}$ then $2w(i) \leq w(j) + w(k) - 6$, for $i \in V_{\mathrm{ave}}$
5. $w(0\text{-sink}) = 0$ and $w(1\text{-sink}) = D$
6. $w(s) \geq \frac{D}{2} + 2^{m+3}$.

If $\mathrm{val}(G) > \frac{1}{2}$ then $\mathrm{Win}_1(G, W)$ can be satisfied by setting $w = \lfloor D \cdot w' \rfloor$ for the (unique) λ-solution w' of G.

Let $\mathrm{Game}(G)$ assert that G is a suitable graph for an SSG. As in Section 3, $\mathrm{Game}(G)$, $\mathrm{Win}_0(G, U)$ and $\mathrm{Win}_1(G, W)$ are all conjunctions of U_2 formulas.

Theorem 4.2. *Provably in* U_3-IND*, it is impossible to satisfy* $\mathrm{Game}(G)$, $\mathrm{Win}_0(G, U)$ *and* $\mathrm{Win}_1(G, W)$ *simultaneously.*

Proof Define a relation $\succeq$ on the vertices of G by $i \succeq j$ if and only if $w(i) - w(j) \geq u(i) - u(j)$. This is Δ_2 (since we can assume that we are given all differences $w(i) - w(j)$ and $u(i) - u(j)$) and, by the properties of binary arithmetic, is a total order. Therefore, by U_2-MAX there exists a vertex i which is $\succeq$-maximum. Fix such an i.

In particular $i \succeq s$, that is, $w(i) - w(s) \geq u(i) - u(s)$. Hence

$$w(i) - \tfrac{D}{2} - 2^{m+3} \geq u(i) - \tfrac{D}{2}.$$

Since $u(i) \geq 0$, it follows that $w(i) \geq 2^{m+3}$. Also $w(i) \neq u(i)$, so i cannot be a sink. Let j and k be the neighbours of i. We know $w(i) - w(j) \geq u(i) - u(j)$ and $w(i) - w(k) \geq u(i) - u(k)$.

Suppose $i \in V_{\max}$. Without loss of generality we may assume $w(j) \geq w(k)$. From condition 2 of Win_0 we have $u(i) \geq u(j)$ and from condition 2 of Win_1 we have $w(i) \leq w(j) - 8$, giving

$$u(i) - u(j) \geq 0 \geq w(i) - w(j) + 8 > w(i) - w(j)$$

which is impossible.

Suppose $i \in V_{\min}$. Without loss of generality we may assume $u(j) \leq u(k)$. This time, from condition 3 of Win_0 we have $u(i) \geq u(j)$ and from condition 3 of Win_1 we have $w(i) \leq w(j) - 8$, so we are back in the previous case.

Finally suppose $i \in V_{\mathrm{ave}}$. From condition 4 of Win_0 and Win_1 we have

$$u(i) - u(j) + u(i) - u(k) + 2 \geq 0 \geq w(i) - w(j) + w(i) - w(k) + 6$$

which is impossible. □

Corollary 4.3 ([18])**.** *The canonical pair for SSGs is reducible to the canonical pair for* PK_1*. Hence if* PK_1 *is weakly automatizable, then we can decide the SSG value problem in polynomial time.* □

5 A game equivalent to resolution

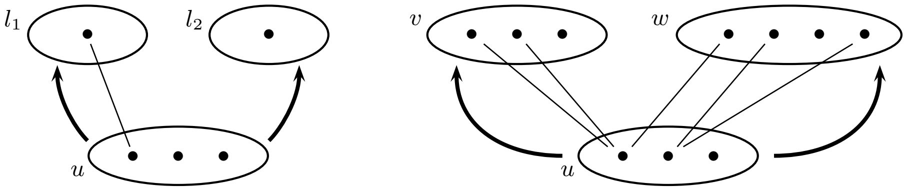
In this section we will define the *point-line game* and prove the following:

Theorem 5.1. *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 Figure 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 $\sigma : V_0 \to V$ or $\tau : V_1 \to V$ 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



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

Non-leaf vertices with points and lines

Figure 1: Components of point-line game graphs.

this case, it is not in general true that a winning strategy exists if and only 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 5.2. *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.*

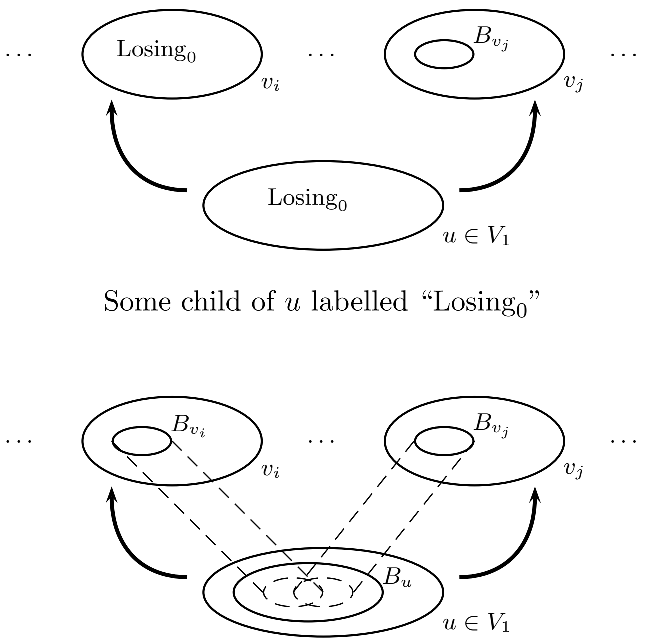
Proof We describe a polynomial time algorithm which, working backwards from the leaves, labels each vertex u with either a set $B_u \subseteq S_u$ of points or a symbol "Losing$_0$". This labelling will have the property that if u is labelled "Losing$_0$" then, regardless of the colouring of u, if the 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}). See Figure 2.

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



Some child of u labelled "Losing$_0$"

No child of u labelled "Losing$_0$"

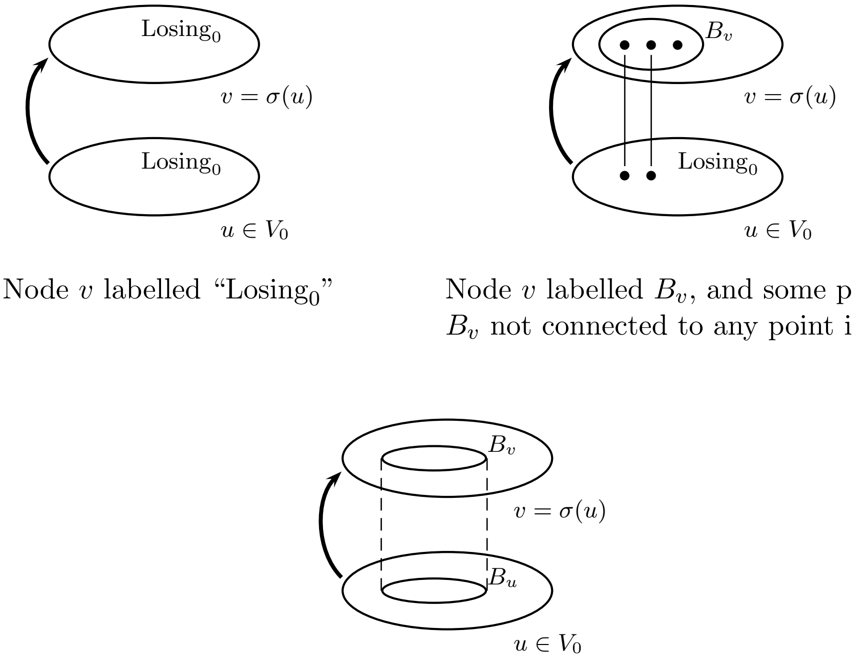
Figure 2: The algorithm constructing a labelling of a point-line game under a strategy σ: the cases when constructing a label for a vertex u in V_1.

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. See Figure 3. □

Theorem 5.3. *The canonical pair for the point-line game is reducible to the canonical pair for* $\mathrm{PK}_{0,k}$ *for some* $k \in \mathbb{N}$*, and hence to the canonical pair for resolution.*

Proof Our proof uses the same basic structure as for the games in previous sections. Expand the language L to include a tuple G of relation symbols E, V_0, V_1, F, S, N and a constant symbol n, together describing the game. Here $S(u, p)$ means that a point p is in the set S_u and $N(p, q)$ means that there is a line from point p to point q. Let $\mathrm{Game}(G)$ be a conjunction of U_2 formulas asserting that G has the properties of a game. In particular, we enforce that the underlying graph is a acyclic by only allowing an edge $E(u, v)$ if $u < v$.

For player 0 we also add symbols E^σ, R^σ, B^σ and Losing_0^σ. We let $\mathrm{Win}_0(G, E^\sigma, R^\sigma, B^\sigma, \mathrm{Losing}_0^\sigma)$ be a conjunction of U_2 formulas asserting that E^σ arises from a strategy for player 0; that $R^\sigma(u, v)$ holds if v is reachable



Node v labelled "Losing$_0$"

Node v labelled B_v, and some point in B_v not connected to any point in u

Node v labelled B_v, and all points in B_v connected to points in u

Figure 3: The algorithm constructing a labelling of a point-line game under a strategy σ: the cases when constructing a label for a vertex u in V_0.

from u in E^σ; that B^σ and Losing_0^σ give a labelling of the vertices as in the proof of Lemma 5.2; and that for every vertex v, if $R^\sigma(s, v)$ then $v \notin \text{Losing}_0^\sigma$.

For player 1 we similarly add symbols E^τ, R^τ, W^τ and Losing_1^τ and define a similar formula $\text{Win}_1(G, E^\tau, R^\tau, W^\tau, \text{Losing}_1^\tau)$. Here W^τ corresponds to B^σ and represents points that must be coloured white for player 1 to win using strategy τ.

It is easy to see that if player 0 has a winning strategy then we can satisfy $\text{Win}_0(G, E^\sigma, R^\sigma, B^\sigma, \text{Losing}_0^\sigma)$ and that a similar thing is true for player 1. Hence for the theorem it is enough to give a U$_2$-IND proof that Game(G), $\text{Win}_0(G, E^\sigma, R^\sigma, B^\sigma, \text{Losing}_0^\sigma)$ and $\text{Win}_1(G, E^\tau, R^\tau, W^\tau, \text{Losing}_1^\tau)$ cannot be satisfied simultaneously.

Suppose otherwise. We will show by U$_2$-IND that for $u = n-1, \ldots, 0$,

$$\forall v > u,\ R^\sigma(s, v) \wedge R^\tau(s, v) \rightarrow \exists p \in B_v^\sigma \cap W_v^\tau.$$

This holds trivially at $n-1$. Suppose that it holds at u, and that $R^\sigma(s, u)$ and $R^\tau(s, u)$. For the induction, it is enough to show that $B_u^\sigma \cap W_u^\tau$ is nonempty. If u is a leaf, then by rule 1 of our labelling algorithm the unique point p in u must be in both B^σ_u and W^τ_u. So we may assume that u is an internal vertex.

Without loss of generality assume $u \in V_0$. Taking $v = \sigma(u)$, from our assumptions we must have $R^\sigma(s,v)$, $R^\tau(s,v)$ and $v > u$, so by the inductive hypothesis there is some point q in $B^\sigma_v \cap W^\tau_v$. Furthermore, neither u nor v is in either Losing^σ_0 or Losing^τ_1.

Since $u \notin \text{Losing}^\sigma_0$ and $q \in B^\sigma_v$, there must be a line in the game connecting q with some point p in u, by rule 3 of our labelling algorithm for σ. Hence p is in B^σ_u by rule 3 of the algorithm for σ, and in W^τ_u by rule 2 of the algorithm for τ. This completes the induction.

It follows that there is some point $p \in B^\sigma_s \cap W^\tau_s$ for the start vertex s, which is impossible since s contains no points. □

We will prove the other direction of Theorem 5.1 by showing that the interpolation pair for PK_1, which is known to be equivalent to the canonical pair for resolution, is reducible to the canonical pair for the game. In fact we will not use the system PK_1 directly, but will use a similar system PK'_1 defined below, which is easily shown to be p-equivalent to PK_1 and hence to have an equivalent interpolation pair.

A PK'_1 refutation is a sequence of DNFs, each written as a list of conjunctions separated by commas. However unlike in (our definition of) PK_1, DNFs in PK'_1 behave like *sequences* of their disjuncts, not like sets. This means that repetition and ordering of disjuncts now matter, and we include explicit structural rules to manipulate them. We still treat conjunctions as sets of their conjuncts.

The rules of PK'_1 are as follows, for literals z, conjunctions α, β and sequences of conjunctions Γ, Δ:

$$\text{weakening } \frac{\Gamma}{\Gamma, \Delta} \qquad \text{contraction } \frac{\Gamma, \alpha, \alpha}{\Gamma, \alpha} \qquad \text{exchange } \frac{\Gamma, \alpha, \beta, \Delta}{\Gamma, \beta, \alpha, \Delta}$$

$$\text{cut } \frac{\Gamma, \alpha \wedge z \qquad \Gamma, \neg z}{\Gamma} \qquad \wedge\text{-introduction } \frac{\Gamma, \alpha \qquad \Gamma, z}{\Gamma, \alpha \wedge z}$$

We also allow introduction of logical axioms $\overline{z, \neg z}$.

Theorem 5.4. *The interpolation pair of* PK'_1 *is reducible to the canonical pair for the point-line game.*

Proof We are given two sets of clauses Φ and Ψ in disjoint sets of variables X and Y. We are also given a PK'_1 refutation π of $\Phi \cup \Psi$. We may assume without loss of generality that Φ already contains all axioms $x, \neg x$ for variables from X, and similarly for Ψ and axioms from Y, and that there are no other introductions of axioms in π. We will 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.

$$\Gamma \;=\; \ldots, z, \ldots, z', \ldots$$

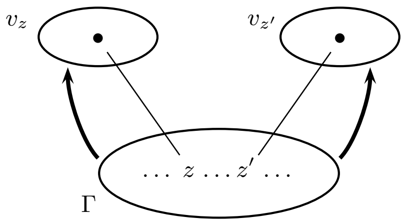


Figure 4: Constructing point-line games from PK_1' derivations: the case of an initial clause Γ.

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 vertex for each literal z arising from a variable in $X \cup Y$, and each such vertex contains a single point.

The vertices corresponding to literals are the leaf vertices. For each vertex u corresponding to an initial clause of the proof (that is, a clause from $\Phi \cup \Psi$), and each literal z occurring in the clause, there is an edge connecting u to the leaf vertex v_z for the literal z, and a line connecting the point in u corresponding to z to the single point in v_z. See Figure 4.

The vertices corresponding to non-initial DNFs are connected by edges to the vertices corresponding to the premises from which they are derived. We define the lines connecting the points contained in these vertices as follows, using the notation we used in the definitions of the rules of PK_1'. See Figure 5.

For every rule, we connect every conjunction in a sequence Γ in either premise of a rule to its descendant in Γ in the conclusion of the rule. For the exchange rule, we also do this for Δ.

For the cut and weakening rules, there are no other lines. For the $\wedge$-introduction rule, we connect α in the left-hand premise with $\alpha \wedge z$ in the conclusion. For the contraction rule, we connect both occurrences of α in the premise with the α in the conclusion. For the exchange rule, we connect α in the premise with α in the conclusion, and similarly for β.

We finally describe how the non-leaf vertices are assigned to the two players. If a vertex u corresponds to an initial clause in Φ then $u \in V_0$, and if u corresponds to an initial clause in Ψ then $u \in V_1$. For vertices u corresponding to non-initial DNFs, if u was derived by weakening, contraction or exchange, it does not matter how we assign it. If it was defined by cut or $\wedge$-introduction, the assignment depends on the literal z appearing in the rule: if z comes from an X variable, we put $u \in V_0$, and if z comes from a Y variable, we put $u \in V_1$. This completes the definition of the game G.

Now let A be a truth-assignment to the variables X which satisfies Φ. We will use A to define a positional winning strategy for player 0 (the case

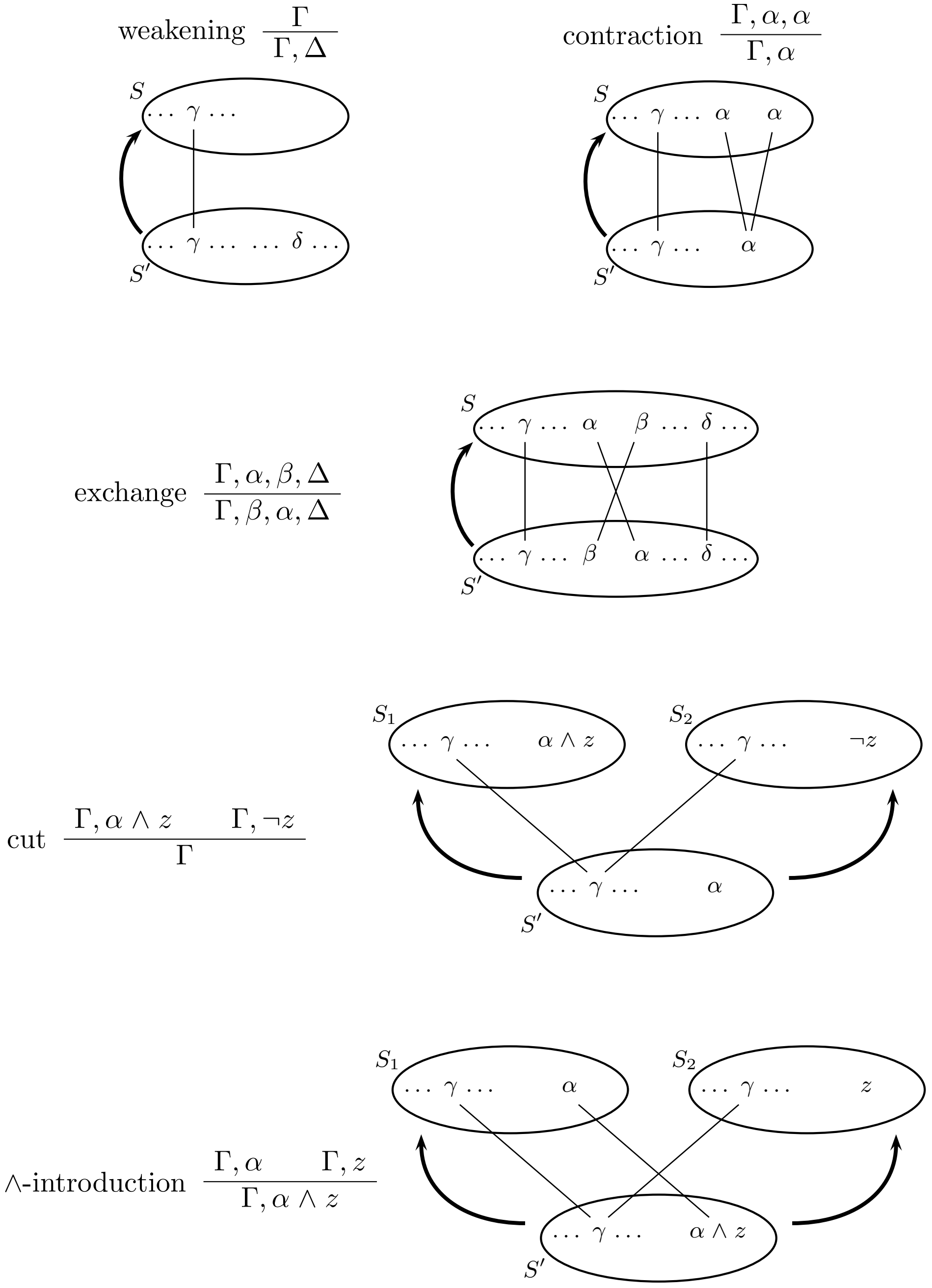

weakening $\frac{\Gamma}{\Gamma, \Delta}$
S
$\ldots\ \gamma \ldots$
$\ldots\ \gamma \ldots\ \ldots\ \delta \ldots$
S'
contraction $\frac{\Gamma, \alpha, \alpha}{\Gamma, \alpha}$
S
$\ldots\ \gamma \ldots\ \alpha\ \ \alpha$
$\ldots\ \gamma \ldots\ \ \alpha$
S'
exchange $\frac{\Gamma, \alpha, \beta, \Delta}{\Gamma, \beta, \alpha, \Delta}$
S
$\ldots\ \gamma \ldots\ \alpha\ \ \beta \ldots\ \delta \ldots$
$\ldots\ \gamma \ldots\ \beta\ \ \alpha \ldots\ \delta \ldots$
S'
cut $\frac{\Gamma, \alpha \wedge z \qquad \Gamma, \neg z}{\Gamma}$
S_1
$\ldots\ \gamma \ldots\ \ \alpha \wedge z$
S_2
$\ldots\ \gamma \ldots\ \ \neg z$
$\ldots\ \gamma \ldots\ \ \alpha$
S'
$\wedge$-introduction $\frac{\Gamma, \alpha \qquad \Gamma, z}{\Gamma, \alpha \wedge z}$
S_1
$\ldots\ \gamma \ldots\ \ \alpha$
S_2
$\ldots\ \gamma \ldots\ \ z$
$\ldots\ \gamma \ldots\ \ \alpha \wedge z$
S'

Figure 5: Constructing point-line games from PK_1' derivations: applying rules. By S' we always denote the sequence of formulas in the conclusion of a rule; by S those in the premise, if the rules have exactly one premise; and by S_1 and S_2 those in respectively the left and right premise, for rules with two premises.

for player 1 is symmetrical). Let $u \in V_0$.

1. Suppose u is derived using the cut rule. If z is false in A, choose the edge from u going to the left, that is, to the vertex containing $\alpha \wedge z$. Otherwise go right, that is, to the vertex containing $\neg z$.

2. Suppose u is derived using the $\wedge$-introduction rule. If z is false, go right; otherwise go left.

3. Suppose u is an initial clause from Φ. Pick the first satisfied literal z in the clause and go to the leaf vertex corresponding to z.

For a conjunction γ, define the *X-part of γ* to be γ with all literals that use variables from Y deleted. In particular, if γ consists solely of Y literals, the X-part of γ is empty and we will treat it as the constant for truth. To prove that the strategy described above is a winning strategy, we will show that the following invariant is preserved during any game played according to it:

> *If the pebble is on a non-leaf vertex u, and p is a point in u corresponding to a conjunction whose X-part is satisfied by A, then p is coloured black.*

This property immediately implies that player 0 wins. This is because the game must eventually reach a vertex u that is an initial clause of the refutation. If $u \in \Phi$, then some literal in u is satisfied by A. Hence by the invariant, this literal must be coloured black, since Φ only contains variables from X. Hence player 0 can move to the corresponding leaf vertex and colour its point black (and notice that this move can be chosen depending only on A, and not on the colouring of the points in u). On the other hand if $u \in \Psi$, then the X-parts of all literals in u are empty, hence true and coloured black. Thus, whatever leaf player 1 picks, its point will be coloured black.

It remains to show that the invariant is preserved. It holds at the start vertex, since that has no points. Suppose it holds at a non-leaf vertex u which does not correspond to an initial clause. If u was derived by weakening, contraction or exchange then the invariant is preserved trivially. Otherwise, let z be the literal appearing in the rule by which u was derived and let v be the vertex that the game moves to after u. In all cases, the property is preserved trivially on points corresponding to conjunctions in Γ.

Suppose that u was derived using the cut rule. If $u \in V_1$, then whether player 1 chooses to go left or right, the new point (corresponding respectively to $\alpha \wedge z$ or $\neg z$) gets coloured black, so the invariant is preserved. So suppose $u \in V_0$, meaning that z comes from an X-variable. If z is false, then by the definition of the strategy player 0 chooses to go left. But from our assumption the X-part of $\alpha \wedge z$ is false, so it does not matter how it is coloured and the invariant is preserved. If z is true, then player 0 chooses to go right. But similarly the X-part of $\neg z$ is false and the invariant is preserved.

Suppose that u was derived using the $\wedge$-introduction rule. Suppose first that $u \in V_1$, so z comes from a Y-variable. If player 1 goes left, then α gets coloured the same colour as $\alpha \wedge z$, since there is a line connecting them; but both points have the same X-part so the invariant is preserved. If player 1 goes right, then z is automatically coloured black. Suppose now that $u \in V_0$, so z comes from an X-variable. If z is false, then player 0 goes right, and it does not matter how the point z is coloured. If z is true, then player 0 goes left and α gets the same colour as $\alpha \wedge z$; but in this case, the X-part of α is satisfied if and only if the X-part of $\alpha \wedge z$ is. □

Corollary 5.5. *The canonical pair of parity games is polynomially reducible to the canonical pair of point-line games.*

Proof The canonical pair of parity games is reducible to the canonical pair of resolution which in turn is reducible to the canonical pair of point-line games. □

It might be interesting to construct a direct reduction from parity games to point-line games.

A Translating first-order into propositional proofs

Let L and L^+ be theories as in Section 1.2, so that in particular all terms are polynomially bounded. For simplicity of presentation we will assume that $L^+ = L \cup \{R\}$ for exactly one binary relation symbol R. This is easily extended to arbitrary tuples of relation symbols of arbitrary arity.

Definition A.1. A *first-order k-conjunction* is a conjunction of the form $\sigma \wedge \phi_1 \wedge \cdots \wedge \phi_m$ where σ is any L formula, $m \leq k$ and each of $\phi_1, \ldots, \phi_m$ has the form $R(s,t)$ or $\neg R(s,t)$ for L-terms s and t. A *first-order k-disjunction* is defined dually.

For odd d, a *strict* $\mathrm{U}_{d,k}$ *formula* is a strict U_d formula whose quantifier-free part is a conjunction of first-order k-disjunctions. For even d, a strict $\mathrm{U}_{d,k}$ formula is a strict U_d formula whose quantifier-free part is a disjunction of first-order k-conjunctions. A $\mathrm{U}_{d,k}$ *formula* is a subformula of a strict $\mathrm{U}_{d,k}$ formula. The *strict* $\mathrm{E}_{d,k}$ *formulas* and $\mathrm{E}_{d,k}$ *formulas* are defined dually.

The theory $\mathrm{U}_{d,k}$-IND consists of BASE together with the usual induction scheme for all $\mathrm{U}_{d,k}$ formulas, with parameters. The theory $\mathrm{E}_{d,k}$-IND is defined similarly (and is equivalent).

We will show that $\mathrm{U}_{d+2,k}$-IND refutations can be translated into families of polynomial size $\mathrm{PK}_{d,k}$ refutations. We will first prove this for $\mathrm{U}_{2,1}$-IND and resolution, and then derive the general case.

Notice that the translation $\langle \phi \rangle^0_\alpha$ turns a $\mathrm{U}_{2,1}$ formula ϕ into a set of clauses, that is, of disjunctions of literals. Recall that we treat the symbols $\top$ and $\bot$ respectively as the empty set and the singleton set containing the empty clause.

Theorem A.2. *Suppose that $\phi_1(x), \ldots, \phi_\ell(x)$ are strict $\mathrm{U}_{2,1}$ formulas, with x the only free variable, such that $\mathrm{U}_{2,1}$-IND proves $\forall x\, \neg(\phi_1(x) \wedge \cdots \wedge \phi_\ell(x))$. Then the family of CNFs*

$$\Phi_n := \langle \phi_1(x) \rangle^0_{[x \mapsto n]} \cup \cdots \cup \langle \phi_\ell(x) \rangle^0_{[x \mapsto n]}$$

has polynomial size resolution refutations.

A *resolution derivation* of a set B of clauses from a set A of clauses is a sequence of clauses, ending with the clauses in B, such that each line in the proof is either from A, or is a logical axiom $p \vee \neg p$, or follows from earlier clauses in the sequence by a rule. We will call A the *initial clauses* and B the *final clauses*, and will call such a derivation a *derivation of $A \vdash B$*.

Definition A.3. For sets of clauses C and D, we write $C * D$ for the set of clauses $\{\phi \vee \psi : \phi \in C, \psi \in D\}$.

Notice that $\bigwedge C \vee \bigwedge D$ is logically equivalent to $\bigwedge (C * D)$.

Lemma A.4. *Let C, D and E be sets of clauses.*

1. *The operator $*$ is associative, commutative, and distributive over $\cup$.*
2. *We have $C \subseteq C * C$, $C * \bot = C$ and $C * \top = \top$.*
3. *We can derive $C \vdash C * D$ by weakening.*
4. *Given a derivation π of $C \vdash D$, there is a derivation of $C * E \vdash D * E$ of size polynomial in the sizes of π and E, obtained by multiplying each clause of π by E.* □

To analyse $\mathrm{U}_{2,1}$-IND proofs we will use the sequent calculus LKB_e for bounded arithmetic, as presented in [8]. This is a system for deriving sequents $\phi_1, \dots, \phi_\ell \longrightarrow \psi_1, \dots, \psi_m$ of bounded formulas, where the intended meaning of a sequent is that the conjunction of the formulas on the left implies the disjunction of the formulas on the right. It has weak structural rules, which allow us to treat each side of a sequent as a set of formulas. It has logical axioms $\phi \longrightarrow \phi$, equality axioms, and non-logical axioms, which in our case have the form $\longrightarrow \sigma$ for σ a formula from BASE. Its other rules are listed in the proof below. They consist of rules for introducing propositional connectives and bounded quantifiers on the left and right hand side of a sequent, the cut rule, and the induction rule. The quantifier and induction rules all involve an *eigenvariable* which is not allowed to appear in the bottom sequent.

The translation $\langle \phi \rangle^d_\alpha$ was only defined for strict $\mathrm{U}_{d+2,k}$ formulas. We extend it to non-strict $\mathrm{U}_{d+2,k}$ formulas ϕ by first padding ϕ using dummy quantifiers. Remember that assignments are total maps from variables to numbers in which at most finitely many variables are assigned non-zero values.

Theorem A.5. *Suppose that there is a sequent calculus derivation Π ending in the sequent*

$$\phi_1, \dots, \phi_\ell \longrightarrow \psi_1, \dots, \psi_m$$

where every formula appearing in the derivation is a $\mathrm{U}_{2,1}$ formula. The derivation may use the $\mathrm{U}_{2,1}$ induction rule and may use any universally true, quantifier-free L-formula as an axiom. Then there is a polynomial p such that for any assignment α, there is a resolution derivation π, of size $p(\alpha)$, of

$$\langle \phi_1 \rangle^0_\alpha \cup \dots \cup \langle \phi_\ell \rangle^0_\alpha \vdash \langle \psi_1 \rangle^0_\alpha * \dots * \langle \psi_m \rangle^0_\alpha.$$

Here we write $p(\alpha)$ for $p(n)$, where n is the maximum value assigned by α.

Proof The proof is by induction on the derivation Π and splits into cases depending on the rule by which the last sequent is derived. The existence of a polynomial size bound will be clear from the construction. The construction follows that of [30] and [23] but is simpler, because we are not worried about issues of uniformity.

For readability of notation, throughout this proof we will write $\langle \phi \rangle$ to mean $\langle \phi \rangle^0_\alpha$. For a first-order cedent $\Gamma = \phi_1, \ldots, \phi_\ell$ we will write Γ° for the set of clauses $\langle \phi_1 \rangle \cup \cdots \cup \langle \phi_\ell \rangle$ and Γ^* for the set of clauses $\langle \phi_1 \rangle * \cdots * \langle \phi_\ell \rangle$.

Axioms and weak structural rules Logical axioms, exchange, contraction and left weakening are trivial. Non-logical axioms (apart from equality) are also trivial, as these are universally-true L-formulas and hence always translate into $\top$, the empty set of clauses. Equality axioms not involving R are treated similarly. For equality axioms involving R, of the form

$$s_1 = s_2, t_1 = t_2, R(s_1, t_1) \longrightarrow R(s_2, t_2),$$

if $s_1 \neq s_2$ or $t_1 \neq t_2$ in α we use the fact that we can derive anything from the empty clause $\bot$. Otherwise we use the trivial derivation of $r_{i,j} \vdash r_{i,j}$, where $i = \langle s_1 \rangle_\alpha$ and $j = \langle t_1 \rangle_\alpha$. For right weakening

$$\frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \phi}$$

we use weakening to derive $\Delta^* * \langle \phi \rangle$ from Δ^*.

Propositional $\wedge$-introduction Suppose the last rule applied in Π is

$$\frac{\Gamma, \phi, \psi \longrightarrow \Delta}{\Gamma, \phi \wedge \psi \longrightarrow \Delta} \quad \text{or} \quad \frac{\Gamma \longrightarrow \Delta, \phi \qquad \Gamma \longrightarrow \Delta, \psi}{\Gamma \longrightarrow \Delta, \phi \wedge \psi}.$$

In both cases, by our assumptions about Π we may assume without loss of generality that ϕ is an L-formula. Hence either $\langle \phi \rangle = \top$ and $\langle \phi \wedge \psi \rangle = \langle \psi \rangle$, or $\langle \phi \rangle = \bot$ and $\langle \phi \wedge \psi \rangle = \bot$. Both cases are trivial.

Propositional $\vee$-introduction Suppose the last rule applied in Π is

$$\frac{\Gamma, \phi \longrightarrow \Delta \qquad \Gamma, \psi \longrightarrow \Delta}{\Gamma, \phi \vee \psi \longrightarrow \Delta} \quad \text{or} \quad \frac{\Gamma \longrightarrow \Delta, \phi, \psi}{\Gamma \longrightarrow \Delta, \phi \vee \psi}.$$

In both cases we may assume that ϕ and ψ are disjunctions of first-order 1-conjunctions. It follows that $\langle \phi \vee \psi \rangle = \langle \phi \rangle * \langle \psi \rangle$. The right-hand case is then trivial. For the left-hand case, by the inductive hypothesis we have derivations π_1 and π_2 of

$$\Gamma^\circ \cup \langle \phi \rangle \vdash \Delta^* \quad \text{and} \quad \Gamma^\circ \cup \langle \psi \rangle \vdash \Delta^*.$$

By multiplying every clause in π_1 by $\langle \psi \rangle$ we get a derivation π_1' of

$$\Gamma^\circ * \langle \psi \rangle \cup \langle \phi \rangle * \langle \psi \rangle \vdash \Delta^* * \langle \psi \rangle.$$

By multiplying every clause in π_2 by Δ^* we get a derivation π_2' of

$$\Gamma^\circ * \Delta^* \cup \langle\psi\rangle * \Delta^* \vdash \Delta^*.$$

Combining π_1' with π_2' and using weakening to derive both $\Gamma^\circ * \langle\psi\rangle$ and $\Gamma^\circ * \Delta^*$ from Γ°, we get the required derivation $\Gamma^\circ \cup \langle\phi\rangle * \langle\psi\rangle \vdash \Delta^*$.

Propositional $\neg$-introduction Suppose the last rule applied in Π is

$$\frac{\Gamma \longrightarrow \Delta, \phi}{\Gamma, \neg\phi \longrightarrow \Delta} \quad \text{or} \quad \frac{\Gamma, \phi \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg\phi}\ .$$

If ϕ is an L-formula then these are trivial. Otherwise, we may assume that ϕ is an atomic sentence $R(s,t)$ and that $\langle\phi\rangle = \{r\}$ and $\langle\neg\phi\rangle = \{\neg r\}$ for some propositional variable r (where we are abusing notation slightly and identifying the literals r and $\neg r$ with the single-element clauses $\bigvee\{r\}$ and $\bigvee\{\neg r\}$). Let π be a derivation for the upper sequent.

For the left-hand case, we resolve every final clause of π with $\neg r$.

For the right-hand case we add $\neg r$ to every clause in π, so that the initial clause r is replaced by the axiom instance $r \vee \neg r$. This gives a derivation of $\Gamma^\circ * \{\neg r\} \vdash \Delta^* * \{\neg r\}$. We derive $\Gamma^\circ * \{\neg r\}$ from Γ° by weakening.

Induction Suppose the last rule applied in Π is

$$\frac{\Gamma, \phi(x) \longrightarrow \Delta, \phi(x+1)}{\Gamma, \phi(0) \longrightarrow \Delta, \phi(t)}\ .$$

Let $n = \langle t\rangle_\alpha$. We will write $\langle\phi\rangle_i$ for $\langle\phi(x)\rangle^0_{\alpha[x\mapsto i]}$. By the definition of the translation for terms, we have $\langle\phi\rangle_{i+1} = \langle\phi(x+1)\rangle^0_{\alpha[x\mapsto i]}$.

Observe that since x does not appear as a free variable in the bottom sequent, in particular it does not appear free in Γ and Δ. It follows that Γ° and Δ^* stay the same under the two assignments α and $\alpha[x \mapsto i]$. By the inductive hypothesis there is some polynomial p such that for each $i < n$ there is a resolution derivation π_i, of size bounded by $p(\alpha)$, of

$$\Gamma^\circ \cup \langle\phi\rangle_i \vdash \Delta^* * \langle\phi\rangle_{i+1}.$$

Multiplying by Δ^*, we get a resolution derivation π_i' of

$$\Delta^* * \Gamma^\circ \cup \Delta^* * \langle\phi\rangle_i \vdash \Delta^* * \langle\phi\rangle_{i+1}.$$

Writing the derivations $\pi_0', \ldots, \pi_{n-1}'$ one after the other, and observing that we can derive $\Delta^* * \Gamma^\circ \cup \Delta^* * \langle\phi\rangle_0$ from $\Gamma^\circ \cup \langle\phi\rangle_0$ by weakening, we obtain the required derivation $\Gamma^\circ \cup \langle\phi\rangle_0 \vdash \Delta^* * \langle\phi\rangle_t$.

Cut This is done the same way as one step in the induction rule.

Bounded $\exists$-left-introduction Suppose the last rule applied in Π is

$$\frac{x < s, \theta(x), \Gamma \longrightarrow \Delta}{\exists y {<} s\, \theta(y), \Gamma \longrightarrow \Delta}\ .$$

Let $n = \langle s \rangle_\alpha$. We may assume that $\theta(x)$ is a disjunction of first-order 1-conjunctions. As above, we will write $\langle \theta \rangle_i$ for $\langle \theta(x) \rangle^0_{\alpha[x \mapsto i]}$. Note that $\langle \theta \rangle_i$ contains exactly one clause. We will write Θ_i for $\bot * \langle \theta \rangle_0 * \cdots * \langle \theta \rangle_{i-1}$. Notice that $\Theta_0 = \bot$ and $\Theta_n = \langle \exists y {<} s\, \theta(y) \rangle$.

For each $i < n$ by the inductive hypothesis we have a polynomial sized derivation π_i of

$$\Gamma^\circ \cup \langle \theta \rangle_i \vdash \Delta^*$$

where $x < s$ does not appear, as it translates to $\top$. Multiplying this derivation by $\Delta^* * \Theta_i$, we get a derivation π'_i of

$$\Delta^* * \Theta_i * \Gamma^\circ \cup \Delta^* * \Theta_i * \langle \theta \rangle_i \vdash \Delta^* * \Theta_i.$$

Now $\Theta_i * \langle \theta \rangle_i$ is just Θ_{i+1}, and $\Delta^* * \Theta_i * \Gamma^\circ$ can be obtained from Γ° by weakening. So we can construct a derivation π''_i of

$$\Gamma^\circ \cup \Delta^* * \Theta_{i+1} \vdash \Delta^* * \Theta_i.$$

Combining the derivations $\pi''_{n-1}, \ldots, \pi''_0$ and using weakening, we obtain the required derivation $\Gamma^\circ \cup \Theta_n \vdash \Delta^*$.

Bounded $\forall$-left-introduction Suppose the last rule applied in Π is

$$\frac{\theta(r), \Gamma \longrightarrow \Delta}{r < s, \forall x {<} s\, \theta(x), \Gamma \longrightarrow \Delta}\ .$$

If $r \geq s$ under α then $r < s$ translates into $\bot$, from which we can derive anything (by weakening). If $r < s$ under α then this case is trivial, as $\langle \theta(r) \rangle$ is then formally a subset of $\langle \forall x {<} s\, \theta(x) \rangle$.

Bounded $\exists$-right-introduction Suppose the last rule applied in Π is

$$\frac{\Gamma \longrightarrow \Delta, \theta(r)}{r < s, \Gamma \longrightarrow \Delta, \exists x {<} s\, \theta(x)}\ .$$

This is similar to the previous case. Only this time $\langle \theta(r) \rangle$ is not a subset of $\langle \exists x {<} s\, \theta(x) \rangle$, but can be obtained by weakening.

Bounded $\forall$-right-introduction Suppose the last rule applied in Π is

$$\frac{x < s, \Gamma \longrightarrow \Delta, \theta(x)}{\Gamma \longrightarrow \Delta, \forall y {<} s\, \theta(y)}\ .$$

By the inductive hypothesis, if we put $n = \langle s \rangle_\alpha$ then, writing $\langle \theta \rangle_i$ for $\langle \theta(x) \rangle^0_{\alpha[x \mapsto i]}$, for each $i < n$ there is a derivation π_i of

$$\Gamma^\circ \vdash \Delta^* * \langle \theta \rangle_i.$$

Combining these gives us the required derivation. □

Proof of Theorem A.2 By our assumption and the free-cut elimination theorem (see for example [8]) there is a sequent calculus derivation satisfying the assumptions of Theorem A.5 and ending with the sequent

$$\phi_1(x), \ldots, \phi_\ell(x) \longrightarrow 0 = 1.$$

The result follows by Theorem A.5. □

We now show the general case.

Theorem A.6. *Let $d \in \mathbb{N}$ with $d \geq 0$. Suppose that $\phi_1(x), \ldots, \phi_\ell(x)$ are strict $\mathrm{U}_{d+2,k}$ formulas, with x the only free variable, such that $\mathrm{U}_{d+2,k}$-IND proves $\forall x \, \neg(\phi_1(x) \wedge \ldots \wedge \phi_\ell(x))$. Then the family*

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

has polynomial size $\mathrm{PK}_{d,k}$ refutations.

Proof First consider the case where $k = 1$. We will prove the result for all d, by induction on d. The base case $d = 0$ is Theorem A.2. So suppose that $d \geq 0$ and we can translate $\mathrm{U}_{d+2,1}$ refutations into $\mathrm{PK}_{d,1}$ refutations. We will suppose d is even—the case for odd d is similar.

Let $\theta_1, \ldots, \theta_m$ be a list of strict $\mathrm{U}_{d+3,1}$ formulas, consisting of the initial formulas $\phi_1, \ldots, \phi_l$ and every formula for which induction is used in the $\mathrm{U}_{d+3,1}$-IND proof of a contradiction from the assumption $\phi_1(x), \ldots, \phi_\ell(x)$.

Each θ_i consists of $d + 2$ alternations of quantifiers followed by a $\mathrm{E}_{1,1}$ formula of the form

$$\psi_i(\bar{z}) := \exists y \!<\! t_i, \, \gamma^i_1(\bar{z}, y) \vee \cdots \vee \gamma^i_{r_i}(\bar{z}, y)$$

where each γ^i_j is a first-order 1-conjunction. Let θ'_i be θ_i with the subformula $\psi_i(\bar{z})$ replaced by a new relation symbol $S_i(\bar{z})$. Let $A_i(x)$ be the set of *first-order extension axioms*

$$\forall \bar{z} \!<\! s_i(x) \, \exists y \!<\! t_i, \, \neg S_i(\bar{z}) \vee \gamma^i_1(\bar{z}, y) \vee \cdots \vee \gamma^i_{r_i}(\bar{z}, y)$$
$$\forall \bar{z} \!<\! s_i(x) \, \forall y \!<\! t_i, \, S_i(\bar{z}) \vee \neg \gamma^i_j(\bar{z}, y) \qquad \text{for each } j = 1, \ldots, r_i$$

expressing that $\forall \bar{z} \!<\! s_i(x), \, S_i(\bar{z}) \leftrightarrow \psi_i(\bar{z})$, where $s_i(x)$ is a bound (obtained from Parikh's theorem) on the values of $\bar{z}$ that can appear in the proof.

There is now a $\mathrm{U}_{d+2,1}$-IND proof of a contradiction from the $\mathrm{U}_{d+2,1}$ assumptions $\phi'_1(x), \dots, \phi'_l(x), A_1(x), \dots, A_m(x)$. Hence by the inductive hypothesis there are polynomial size $\mathrm{PK}_{d,1}$ refutations Π_n of the formulas

$$\langle \phi'_1(x) \rangle^d_n \cup \dots \cup \langle \phi'_l(x) \rangle^d_n \cup \langle A_1(x) \rangle^d_n \cup \dots \cup \langle A_m(x) \rangle^d_n.$$

Here we are writing $\langle \phi(x) \rangle^d_n$ for $\langle \phi(x) \rangle^d_{[x \mapsto n]}$, and abusing notation by treating $A_i(x)$ as though it were a single formula rather than several formulas.

We obtain the desired $\mathrm{PK}_{d+1,1}$ refutation of $\langle \phi_1(x) \rangle^{d+1}_n \cup \dots \cup \langle \phi_l(x) \rangle^{d+1}_n$ by substituting $\langle \psi_i(\bar{z}) \rangle_\alpha$ for $\langle S_i(\bar{z}) \rangle_\alpha$ and $\langle \neg\psi_i(\bar{z}) \rangle_\alpha$ for $\langle \neg S_i(\bar{z}) \rangle_\alpha$ into Π_n, for every α. This increases the depth of the refutation by at most 1, and after the substitution each formula in $\langle A_i(x) \rangle^d_n$ becomes a propositional tautology with a short $\mathrm{PK}_{d+1,1}$ proof (in fact essentially with a $\mathrm{PK}_{1,1}$ proof, since these represent disjunctions of conjunctions of literals whose depth has been artificially padded out by the translation).

We deal with the case $k > 1$ in a similar way, by using extension axioms to obtain a translation of $\mathrm{U}_{d+2,k}$-IND into $\mathrm{PK}_{d,k}$ from our translation of $\mathrm{U}_{d+2,1}$-IND into $\mathrm{PK}_{d,1}$. □

B Binary arithmetic

We will represent integers in *two's-complement form*. To find the m-bit two's-complement representation of an integer x in the range -2^{m-1} to $2^{m-1}-1$, if $x \geq 0$ we take the usual m-bit representation (with leading 0s) of x, and if $x < 0$ we take the usual m-bit representation of $2^m + x$.

This form has the property that, provided we ignore overflow, the operations of addition and subtraction are exactly the same as they would be for unsigned binary integers. Furthermore, comparison of two such integers can be reduced to comparison of unsigned binary integers by first flipping the leading bit. We write numbers with the most significant bit on the left.

To make the following definitions more natural, we will often treat boolean-valued formulas as though they took the numerical values 0 and 1 instead of false and true, in particular writing $\phi = \psi$ for $\phi \leftrightarrow \psi$, $\phi \leq \psi$ for $\phi \to \psi$ and $\phi < \psi$ for $\neg\phi \wedge \psi$.

For $(n{+}1)$-bit strings X and Y we define

$$\begin{aligned} X = Y &\equiv \forall i \leq n,\ X(i) = Y(i) \\ X < Y &\equiv X(n) > Y(n) \\ &\qquad \vee \exists i < n,\ X(i) < Y(i) \wedge \forall j \in (i, n]\ X(j) = Y(j) \\ X \leq Y &\equiv (X = Y) \vee (X < Y). \end{aligned}$$

We say that $X < Y$ *at* i if i witnesses the existential quantifier in the definition of ordering, or if $X(n) > Y(n)$ and $i = n$. We will use 0_n to denote the integer 0 written in $(n{+}1)$-bit two's complement, that is, a string of $n+1$ many 0s.

Observation B.1. *Over* BASE*, the relation $<$ is a strict linear order.*

Observation B.2. *For $(n{+}1)$-bit strings, over* U_1-IND*, $X(n) = 0$ implies $0_n \leq X$ and $X(n) = 1$ implies $X < 0_n$. Furthermore $X < Y$ is Δ_2, since it is equivalent to*

$$\big[X(n) > Y(n)\big] \vee \big[\forall i{<}n \, \big(X(i) > Y(i) \to \exists j{\in}(i,n) \; X(j) < Y(j)\big) \\ \wedge \exists i{<}n \, X(i) < Y(i)\big]. \qquad \square$$

Definition B.3. Let the formula $S(X)$ denote the successor of X, defined by $|S(X)| = |X|$ and

$$S(X)(i) = \begin{cases} 0 & \text{if } \forall j{<}i, \; X(j) = 1 \\ 1 & \text{if } X(i) = 0 \wedge \forall j{<}i, \; X(j) = 1 \\ X(i) & \text{otherwise.} \end{cases}$$

Note that overflow is possible, with $S(2^n - 1) = -2^n$ (working in $(n{+}1)$-bit two's complement). We will usually write $S(X)$ as $X + 1$. If we write $X + 0$, this means simply X. The formula $Y = X + 1$ is U_2.

Observation B.4. *Over* U_1-IND*, $X < Y$ and $X' = X + 1$ implies $X' \leq Y$.*

Proof First suppose that $X(n) > Y(n)$. If X contains only 1s, then $X' = 0_n \leq Y$. If X contains a 0, then nothing to the left of the 0 is changed, so $X'(n) = 1$ and hence $X' < Y$. Now suppose $X < Y$ at $i < n$. If X contains only 1s to the right of i, then $X'(i) = 1$ and $X'(j) = 0$ for all $j < i$, so $X' \leq Y$. If X contains a 0 to the right of i, then $X'(i) = 0$ so $X < Y$. $\square$

Observation B.5. *Over* U_1-IND*, if $X' = X + 1$ and $Y = X^\frown 0 + 1$ then $X'^\frown 0 = Y + 1$. (In other words, $2(X + 1) = 2X + 1 + 1$.)* $\square$

Observation B.6. *An integer X written in $(n{+}1)$-bit two's complement satisfies $-2^{n-1} \leq X \leq 2^{n-1} - 1$ if and only if $X(n) = X(n-1)$.* $\square$

Definition B.7. For $(n{+}1)$-bit strings X, Y, Z, and a string C encoding, for each i, a tuple of five i-bit strings $C_i, C_i^{(0)}, C_i^{(1)}, C_i^{(2)}, C_i^{(3)}$, the U_2 formula $\mathrm{Sum}(X, Y, Z, C)$ expresses that, for $i = n, \ldots, 0$,

1. C_{n+1} is the empty string and $C_0 = Z$
2. $C_i^{(0)} = C_{i+1}{}^\frown 0$
3. $C_i^{(k+1)} = C_i^{(k)} + 1$ for $k = 0, 1, 2$
4. $C_i = C_{i+1}{}^\frown 0 + X(i) + Y(i)$
5. $C_i^{(0)} < C_i^{(1)} < C_i^{(2)} < C_i^{(3)}$ if $i < n - 1$

6. If $Y < 0_n$ then $Z < X$ and if $Y \geq 0_n$ then $Z \geq X$

7. If $X < 0_n$ then $Z < Y$ and if $X \geq 0_n$ then $Z \geq Y$.

Here 1 and 4 contain the essential definition of summation. The other conditions are needed for the formalisation or to simplify the proof of Theorem B.9 below. The right hand side of 4 is formally written as $C_i^{(X(i)+Y(i))}$.

Lemma B.8. *Suppose X, Y, Z are in $(n+1)$-bit two's complement with $-2^{n-1} \leq X, Y \leq 2^{n-1} - 1$ and $X + Y = Z$. Then there is a string C satisfying* $\mathrm{Sum}(X, Y, Z, C)$.

Proof Put $C_n = \mathrm{XOR}(X(n), Y(n))$. Write $X{\upharpoonright}k$ for the string consisting of the k most significant bits of X. By Observation B.6, if $k \geq 2$ we have $-2^{k-2} \leq X{\upharpoonright}k,\ Y{\upharpoonright}k \leq 2^{k-2}-1$. So for $i = n-1, \ldots, 0$, we can set $k = n+1-i$ and put $C_i = X{\upharpoonright}k + Y{\upharpoonright}k$ written in k-bit two's complement, using normal integer addition. Furthermore $C_i \leq 2^{k-1} - 2$, so $C_i{}^\frown 0 \leq 2^k - 4$. It follows that if $i < n - 1$, we have $C_i^{(0)} \leq 2^{k-1} - 4$, so we can add 1 three times without overflow and satisfy condition 5. □

Theorem B.9. *Over Δ_2-IND, assume X, Y, Z, U, V, W are in $(n+1)$-bit two's complement with $X(n) = X(n-1)$, $Y(n) = Y(n-1)$, $U(n) = U(n-1)$ and $V(n) = V(n-1)$. Furthermore suppose $X \leq U$ and $Y \leq V$. Then* $\mathrm{Sum}(X, Y, Z, C)$ *and* $\mathrm{Sum}(U, V, W, D)$ *implies $Z \leq W$.*

Proof We will only do the case where $X < U$ and $Y < V$. The other cases use subsets of this argument. Let $X < U$ at k and $Y < V$ at ℓ. Because Sum is symmetrical, without loss of generality we may assume that $k \geq \ell$. By our assumptions on X and U we know that $k \neq n-1$. There are now three cases.

In the first case, $k = n$. Then $X(n) = 1$ and $U(n) = 0$. Therefore $X < 0_n \leq U$ by Observation B.2. Hence $Z < Y$ and $V \leq W$, by conditions 6 and 7, giving $Z < W$.

In the second case, $k < n - 1$ and $k = \ell$. We have $X(i) = U(i)$ and $Y(i) = V(i)$ for all $i > k$, and can thus use U_1-IND to show that $C_{k+1} = D_{k+1}$. Then $X(k) = Y(k) = 0$ and $U(k) = V(k) = 1$, so $C_k = C_{k+1}{}^\frown 0$ and $D_k = C_{k+1}{}^\frown 0 + 1 + 1$. Therefore $C_k + 1 < D_k$ by condition 5.

We have now established that $C_i + 1 < D_i$ for $i = k$. Formally this is written $C_i^{(X(i)+Y(i)+1)} < D_i$ so is Δ_2. We will use Δ_2-IND to prove it for $i = k-1, \ldots, 0$. Suppose it is true for $i + 1$, that is, $C_{i+1} + 1 < D_{i+1}$. Then it follows from the definition of $<$ that $(C_{i+1} + 1){}^\frown 1 < D_{i+1}{}^\frown 0$. But, using Observation B.5, $(C_{i+1} + 1){}^\frown 1 = C_{i+1}{}^\frown 0 + 1 + 1 + 1$. Therefore by condition 5 we have

$$C_i + 1 = C_{i+1}{}^\frown 0 + X(i) + Y(i) + 1 < D_{i+1}{}^\frown 0 \leq D_i.$$

In the third case, $\ell < k < n-1$. As in case 2, $C_{k+1} = D_{k+1}$. Now $X(k) = 0$, $U(k) = 1$ and $Y(k) = V(k)$. Hence by condition 5,

$$C_k = C_{k+1}{}^\frown 0 + Y(k) < C_{k+1}{}^\frown 0 + 1 + Y(k) = D_k.$$

We now use Δ_2-IND to show $C_i < D_i$ for $i = k-1, \ldots, \ell+1$. Assuming $C_{i+1} < D_{i+1}$, as before we have that $C_{i+1}{}^\frown 1 < D_{i+1}{}^\frown 0$, hence $C_{i+1}{}^\frown 0+1 < D_{i+1}{}^\frown 0$. Since $i > \ell$, $Y(i) = V(i)$. If $Y(i) = 0$, by condition 5

$$C_i = C_{i+1}{}^\frown 0 + X(i) < D_{i+1}{}^\frown 0 \leq D_i.$$

If $Y(i) = 1$, we use Observation B.4 to get $C_{i+1}{}^\frown 0 + 1 + 1 \leq D_{i+1}{}^\frown 0$. Then by condition 5

$$C_i = C_{i+1}{}^\frown 0 + X(i) + 1 \leq D_{i+1}{}^\frown 0 < D_{i+1}{}^\frown 0 + 1 \leq D_i$$

giving the induction step.

A similar argument shows $C_\ell + 1 < D_\ell$. We finally prove that $C_i + 1 < D_i$ for $i = \ell - 1, \ldots, 0$ exactly as in the second case. □

References

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

[2] A. Atserias and M. L. Bonet, *On the automatizability of resolution and related propositional proof systems*, Inform. and Comput., 189 (2004), pp. 182–201.

[3] A. Atserias and E. Maneva, *Mean-payoff games and propositional proofs*, Inform. and Comput., 209 (2011), pp. 664–691.

[4] E. Ben-Sasson and A. Wigderson, *Short proofs are narrow—resolution made simple*, J. ACM, 48 (2001), pp. 149–169.

[5] M. L. Bonet, C. Domingo, R. Gavaldà, A. Maciel, and T. Pitassi, *Non-automatizability of bounded-depth Frege proofs*, Comput. Complexity, 13 (2004), pp. 47–68.

[6] J. Buresh-Oppenheim and T. Morioka, *Relativized NP search problems and propositional proof systems*, in Proceedings of the 19th IEEE Annual Conference on Computational Complexity, IEEE, 2004, pp. 54–67.

[7] S. R. Buss, *Bounded arithmetic*, vol. 3 of Studies in Proof Theory. Lecture Notes, Bibliopolis, Naples, 1986.

[8] ——, *First-order proof theory of arithmetic*, in Handbook of proof theory, vol. 137 of Stud. Logic Found. Math., North-Holland, Amsterdam, 1998, pp. 79–147.

[9] S. R. Buss, L. Kołodziejczyk, and N. Thapen, *Fragments of approximate counting*, 2012. Manuscript, available at http://www.math.cas.cz/~thapen/.

[10] A. Condon, *The complexity of stochastic games*, Inform. and Comput., 96 (1992), pp. 203–224.

[11] ——, *On algorithms for simple stochastic games*, in Advances in computational complexity theory (New Brunswick, NJ, 1990), vol. 13 of DIMACS Ser. Discrete Math. Theoret. Comput. Sci., Amer. Math. Soc., Providence, RI, 1993, pp. 51–71.

[12] A. Ehrenfeucht and J. Mycielski, *Positional strategies for mean payoff games*, Internat. J. Game Theory, 8 (1979), pp. 109–113.

[13] E. A. Emerson, *Automata, tableaux, and temporal logics*, in Logics of programs (Brooklyn, N.Y., 1985), vol. 193 of Lecture Notes in Comput. Sci., Springer, Berlin, 1985, pp. 79–88.

[14] E. A. Emerson and C. S. Jutla, *Tree automata, mu-calculus and determinacy*, in Proceedings of the 32nd annual symposium on Foundations of computer science, SFCS '91, Washington, DC, USA, 1991, IEEE Computer Society, pp. 368–377.

[15] O. Friedmann, *An exponential lower bound for the latest deterministic strategy iteration algorithms*, Log. Methods Comput. Sci., 7 (2011), pp. 3:19, 42.

[16] ——, *Recursive algorithm for parity games requires exponential time*, RAIRO Theor. Inform. Appl., 45 (2011), pp. 449–457.

[17] E. Grädel, W. Thomas, and T. Wilke, eds., *Automata, logics, and infinite games*, vol. 2500 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2002.

[18] L. Huang and T. Pitassi, *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).

[19] B. Juba, *On the hardness of simple stochastic games*, Master's thesis, Carnegie Mellon University, 2005.

[20] J. Krajíček, *Lower bounds to the size of constant-depth Frege proofs*, J. Symbolic Logic, 59 (1994), pp. 73–86.

[21] ——, *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*, J. Symbolic Logic, 62 (1997), pp. 457–486.

[22] ——, *On the weak pigeonhole principle*, Fund. Math., 170 (2001), pp. 123–140.

[23] J. Krajíček, A. Skelley, and N. Thapen, *NP search problems in low fragments of bounded arithmetic*, J. Symbolic Logic, 72 (2007), pp. 649–672.

[24] O. Kullmann, *Upper and lower bounds on the complexity of generalised resolution and generalised constraint satisfaction problems*, Ann. Math. Artif. Intell., 40 (2004), pp. 303–352.

[25] J. Paris and A. Wilkie, *Counting problems in bounded arithmetic*, in Methods in mathematical logic (Caracas, 1983), vol. 1130 of Lecture Notes in Math., Springer, Berlin, 1985, pp. 317–340.

[26] P. Pudlák, *On reducibility and symmetry of disjoint NP pairs*, Theoret. Comput. Sci., 295 (2003), pp. 323–339.

[27] A. Puri, *Theory of hybrid systems and discrete event structures*, PhD thesis, University of California, Berkeley, 1995.

[28] A. A. Razborov, *On provably disjoint NP-pairs*, Tech. Rep. RS-94-36, Basic Research in Computer Science Center, Aarhus, Denmark, November 1994.

[29] L. S. Shapley, *Stochastic games*, Proc. Nat. Acad. Sci. U. S. A., 39 (1953), pp. 1095–1100.

[30] A. Skelley and N. Thapen, *The provably total search problems of bounded arithmetic*, Proc. Lond. Math. Soc. (3), 103 (2011), pp. 106–138.

[31] C. Stirling, *Modal and Temporal Properties of Processes*, Texts in Computer Science, Springer, 2001.

[32] G. Wilmers, *Bounded existential induction*, J. Symbolic Logic, 50 (1985), pp. 72–90.

[33] U. Zwick and M. Paterson, *The complexity of mean payoff games on graphs*, Theoret. Comput. Sci., 158 (1996), pp. 343–359.