Parity Games and Propositional Proofs

Neil Thapen

Institute of Mathematics
Academy of Sciences of the Czech Republic

Joint work with Arnold Beckmann and Pavel Pudlák

Proof complexity

How hard is it to prove a given true sentence?

Proof complexity

How hard is it to prove a given true sentence?

How hard is it to search for a proof (if we know one exists)?

Proof complexity

How hard is it to prove a given true sentence?

How hard is it to search for a proof (if we know one exists)?

How hard is it to tell whether a proof of a given length exists?

Resolution

Resolution is a propositional proof system, used for refuting propositional formulas in conjunctive normal form (CNFs).

A *resolution refutation* of a CNF F is a sequence $C_1, \ldots C_t$ of disjunctions, where

C_t is the empty disjunction

and for each $i = 1, \ldots, t$ either

C_i is a disjunction in F, or

C_i is derived from from two earlier disjunctions C_j and C_k by the *resolution rule*

$$\frac{C \vee x \qquad\qquad D \vee \neg x}{C \vee D}$$

Resolution refutations

Resolution is sound and complete.
But which CNFs have *short* refutations?

Resolution refutations

Resolution is sound and complete.
But which CNFs have *short* refutations?

The ordering principle

A CNF in propositional variables $\{R_{ij} : i, j \in [n]\}$.
It asserts that R is an ordering of $[n]$ with no least element.
It has a resolution refutation of size $O(n^3)$.
The refutation imitates a proof by induction on n.

The pigeonhole principle

A CNF in propositional variables $\{R_{ij} : i \in [n+1], j \in [n]\}$.
It asserts that R is the graph of an injection from $[n+1]$ to $[n]$.
It has no resolution refutation of size $\leq 2^{O(n)}$.
Any refutation has, essentially, to exhaustively check cases.

Resolution refutations

SAT solvers

Many modern SAT solvers, given as input a CNF F, either

- output a satisfying assignment of F, or
- generate what is essentially a resolution refutation of F, witnessing that F is unsatisfiable, or
- run out of time.

They do not run well on CNFs which are unsatisfiable but do not have short resolution refutations.

Automatizability

Resolution is a useful, well-understood system.

Automatizability

Resolution is a useful, well-understood system.

How difficult is it to search for resolution refutations?

Automatizability

Resolution is a useful, well-understood system.

How difficult is it to search for resolution refutations?

Automatizability of resolution

Input: An unsatisfiable CNF F

Task: Output a resolution refutation of F.

We say that resolution is *automatizable* if there is an algorithm which carries out this task in time polynomial in the length of the shortest resolution refutation of F.

Automatizability

Resolution is a useful, well-understood system.

How difficult is it to search for resolution refutations?

Automatizability of resolution

Input: An unsatisfiable CNF F
Task: Output a resolution refutation of F.

We say that resolution is *automatizable* if there is an algorithm which carries out this task in time polynomial in the length of the shortest resolution refutation of F.

Resolution is probably not automatizable [Alekhnovich and Razborov 2008].

Weak automatizability

Is it possible to tell at least whether F has a refutation of a given size?

Weak automatizability

Is it possible to tell at least whether F has a refutation of a given size?

Weak automatizability of resolution

Input: A CNF F and a string 1^m
Task: Output 0 if F is satisfiable.
Output 1 if F has a resolution refutation of size $\leq m$.
In any other case, output anything.

We say that resolution is *weakly automatizable* if there is a polynomial time algorithm carrying out this task.

Results about weak automatizability

Theorem [Krajíček and Pudlák 1994]

If RSA is secure, then the *extended Frege* proof system is not weakly automatizable.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

Results about weak automatizability

Theorem [Krajíček and Pudlák 1994]

If RSA is secure, then the *extended Frege* proof system is not weakly automatizable.

Theorem [Bonet, Pitassi and Raz 1997]

If factoring is hard, then the *Frege* proof system is not weakly automatizable.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

Results about weak automatizability

Theorem [Krajíček and Pudlák 1994]

If RSA is secure, then the *extended Frege* proof system is not weakly automatizable.

Theorem [Bonet, Pitassi and Raz 1997]

If factoring is hard, then the *Frege* proof system is not weakly automatizable.

Theorem [Bonet, Domingo, Gavalda, Maciel and Pitassi 2004]

If factoring is hard, then the *depth*-5 *Frege* proof system is not weakly automatizable.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

Results about weak automatizability

Theorem [Atserias and Maneva 2011]

If the *depth-1 Frege* proof system is weakly automatizable, then the decision problem for *mean payoff games* is in P.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

$$\text{parity games} \leqslant \text{mean payoff games} \leqslant \text{simple stochastic games}$$

Results about weak automatizability

Theorem [Atserias and Maneva 2011]

If the *depth-1 Frege* proof system is weakly automatizable, then the decision problem for *mean payoff games* is in P.

Theorem [Huang and Pitassi 2011]

If depth-1 Frege is weakly automatizable, then the decision problem for *simple stochastic games* is in P.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

$$\text{parity games} \leqslant \text{mean payoff games} \leqslant \text{simple stochastic games}$$

Results about weak automatizability

Theorem [Atserias and Maneva 2011]

If the *depth-1 Frege* proof system is weakly automatizable, then the decision problem for *mean payoff games* is in P.

Theorem [Huang and Pitassi 2011]

If depth-1 Frege is weakly automatizable, then the decision problem for *simple stochastic games* is in P.

Theorem

If resolution is weakly automatizable, then the decision problem for *parity games* is in P.

$$\text{resolution} \leqslant \text{depth-1 Frege} \leqslant \ldots \leqslant \text{Frege} \leqslant \text{extended Frege}$$

$$\text{parity games} \leqslant \text{mean payoff games} \leqslant \text{simple stochastic games}$$

Automatizability

Parity Games

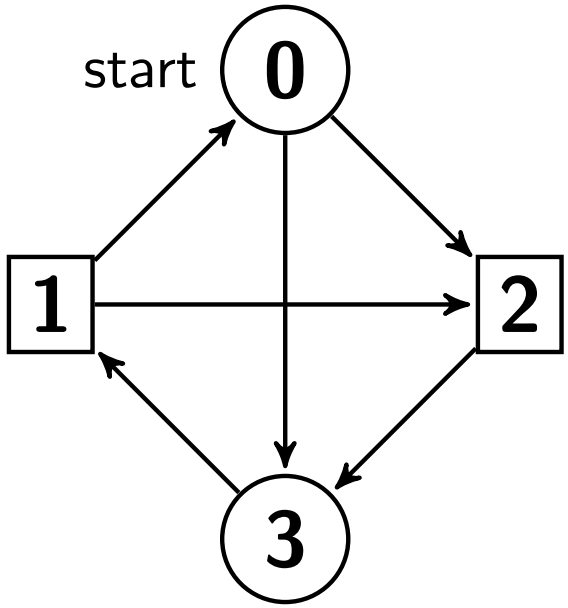
Results

Proof

Parity games

Parity games are a class of two-player, infinite games played on finite directed graphs.

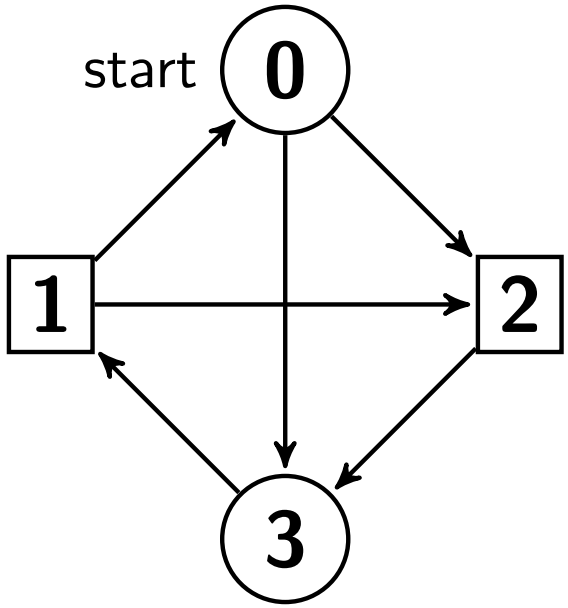
Parity games - structure



$\{0,3\}$ belong to Player 0

$\{1,2\}$ belong to Player 1

Parity games - structure



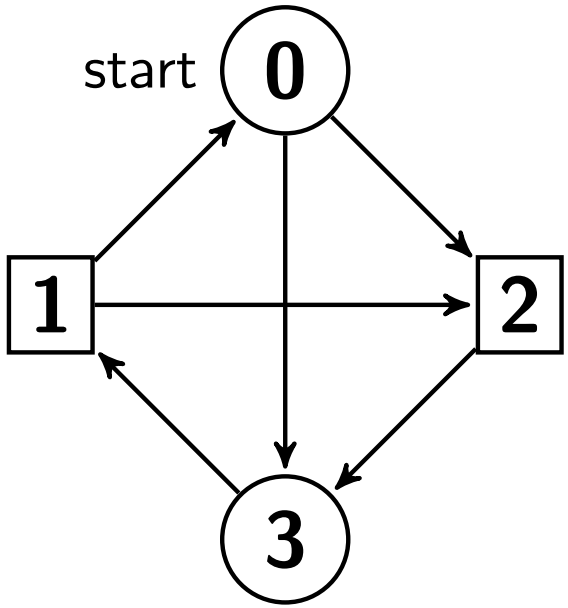
{0,3} belong to Player 0

{1,2} belong to Player 1

A *parity game* G is given by

- a set of vertices $[n] = \{0, \ldots, n-1\}$
- an edge relation E on $[n] \times [n]$, where each vertex has at least one outgoing edge
- a partitioning of the vertices $[n]$ into two sets V_0 and V_1, belonging respectively to Player 0 and Player 1
- a distinguished start vertex.

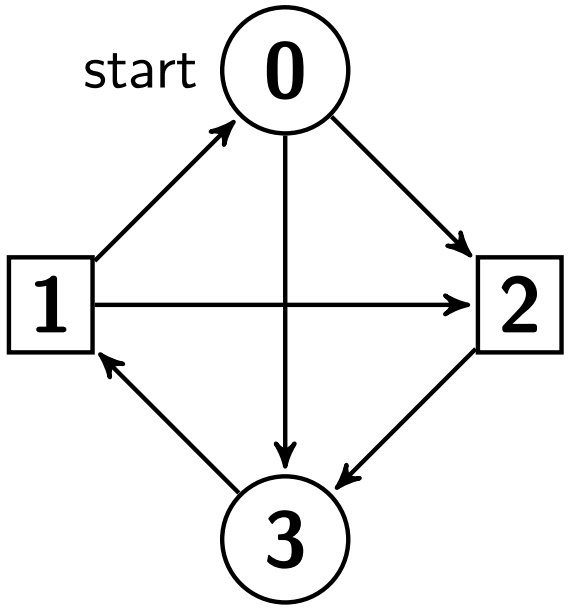
Parity games - how to play



{0,3} belong to Player 0

{1,2} belong to Player 1

Parity games - how to play

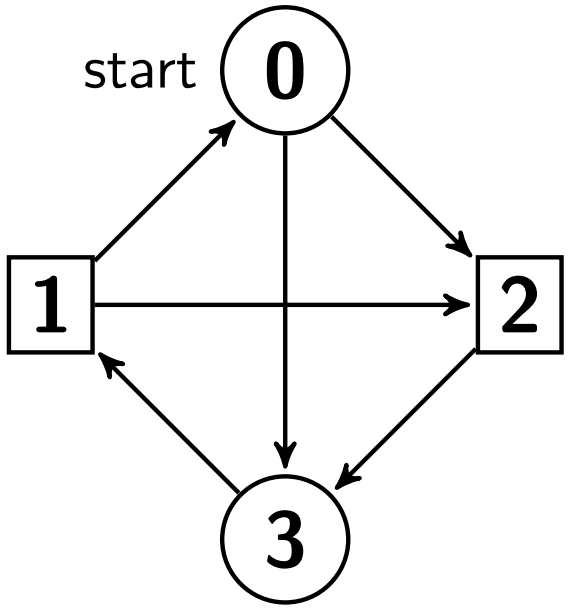


{0,3} belong to Player 0

{1,2} belong to Player 1

Play starts with a token on the start vertex.

Parity games - how to play



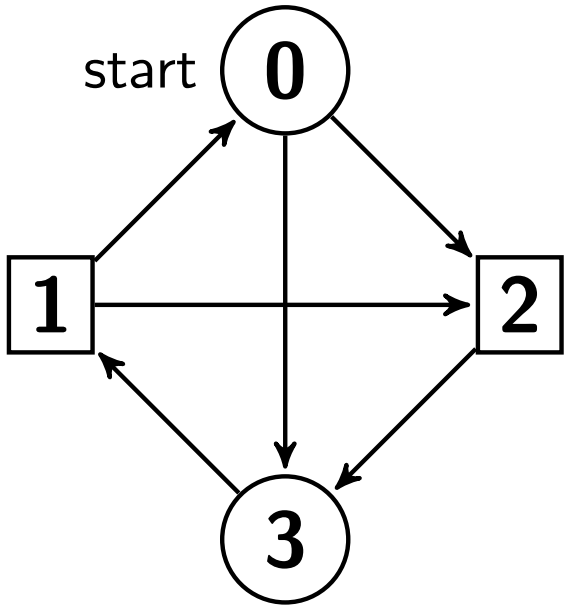
{0,3} belong to Player 0

{1,2} belong to Player 1

Play starts with a token on the start vertex.

The owner of the current vertex chooses which outgoing edge to move along.

Parity games - how to play



{0,3} belong to Player 0

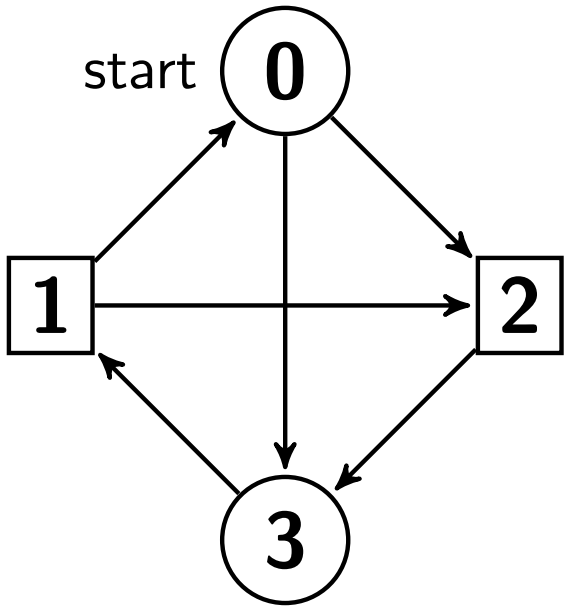
{1,2} belong to Player 1

Play starts with a token on the start vertex.

The owner of the current vertex chooses which outgoing edge to move along.

Play continues forever.

Parity games - how to win

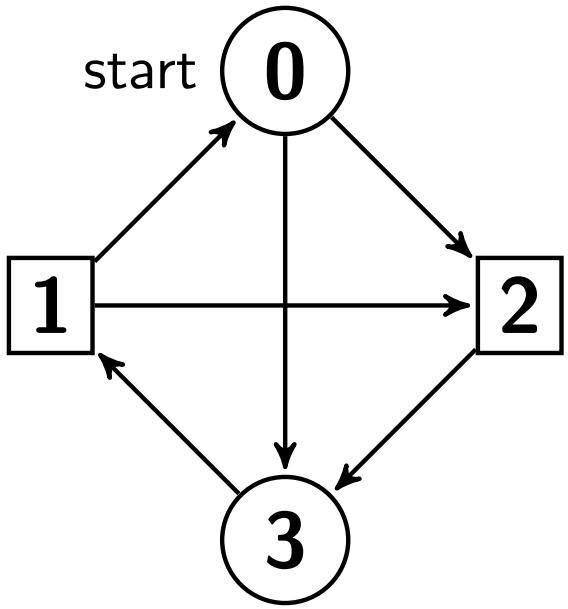


{0,3} belong to Player 0

{1,2} belong to Player 1

Let $v_0, v_1, v_2, \ldots$ be the sequence of vertices visited during the play.

Parity games - how to win

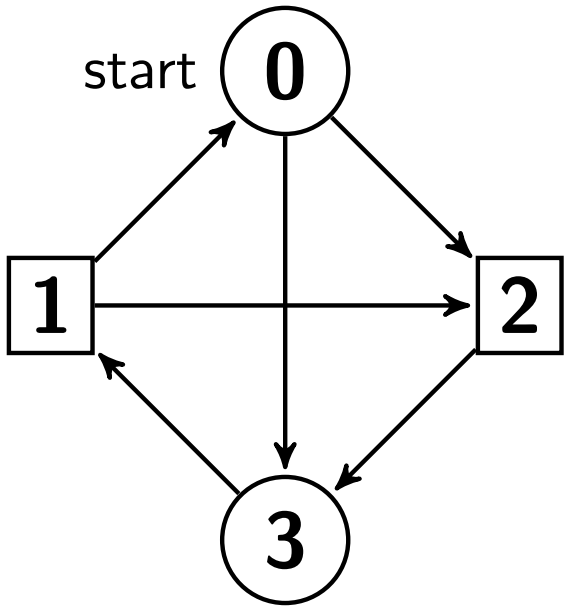


{0,3} belong to Player 0

{1,2} belong to Player 1

Let $v_0, v_1, v_2, \ldots$ be the sequence of vertices visited during the play. Let x be the least-numbered vertex which occurs infinitely often in this sequence.

Parity games - how to win

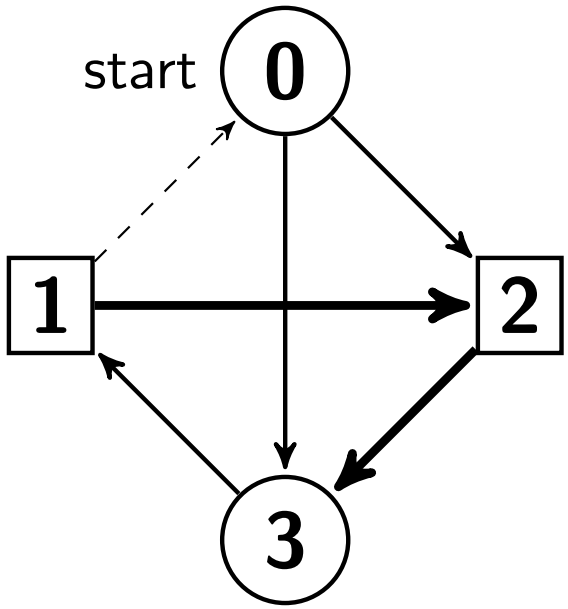


{0,3} belong to Player 0

{1,2} belong to Player 1

Let $v_0, v_1, v_2, \ldots$ be the sequence of vertices visited during the play.
Let x be the least-numbered vertex which occurs infinitely often in this sequence.
The winner of the play is the player who owns vertex x.

Parity games - how to win



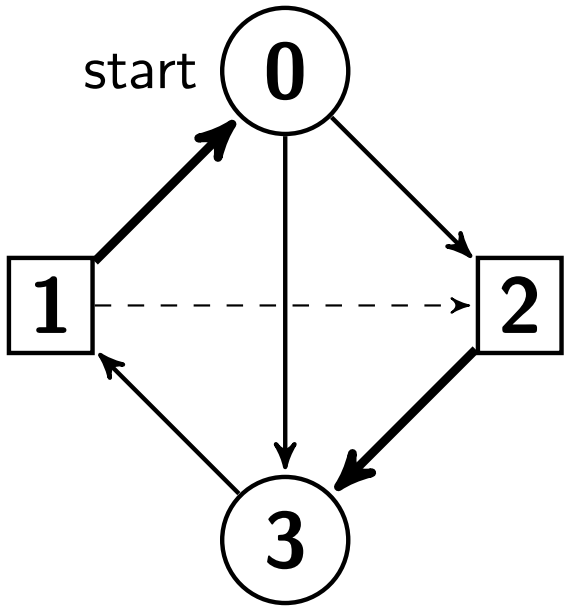
{0,3} belong to Player 0

{1,2} belong to Player 1

Let $v_0, v_1, v_2, \ldots$ be the sequence of vertices visited during the play.
Let x be the least-numbered vertex which occurs infinitely often in this sequence.
The winner of the play is the player who owns vertex x.

In the example, Player 1 **wins** in any play in which he uses the positional strategy $1 \mapsto 2$, $2 \mapsto 3$.

Parity games - how to win



{0,3} belong to Player 0

{1,2} belong to Player 1

Let $v_0, v_1, v_2, \ldots$ be the sequence of vertices visited during the play.
Let x be the least-numbered vertex which occurs infinitely often in this sequence.
The winner of the play is the player who owns vertex x.

In the example, Player 1 **loses** in any play in which he uses the positional strategy $1 \mapsto 0$, $2 \mapsto 3$.

Decision problem

Parity games are determined.

Decision problem

Parity games are determined.

The decision problem for parity games

Input: Parity game G
Question: Which player has a winning strategy in G?

Decision problem

Parity games are determined.

The decision problem for parity games

Input: Parity game G
Question: Which player has a winning strategy in G?

A *positional strategy* is one in which a player chooses, at the beginning of the game, one fixed edge for each of his vertices and uses it throughout the game.

Theorem [Emerson 1985]

A player in a parity game has a winning strategy if and only if he has a positional winning strategy.

Decision problem

Parity games are determined.

The decision problem for parity games (equivalent version)

Input: Parity game G
Question: Which player has a positional winning strategy in G?

A *positional strategy* is one in which a player chooses, at the beginning of the game, one fixed edge for each of his vertices and uses it throughout the game.

Theorem [Emerson 1985]

A player in a parity game has a winning strategy if and only if he has a positional winning strategy.

Complexity of the decision problem - 1

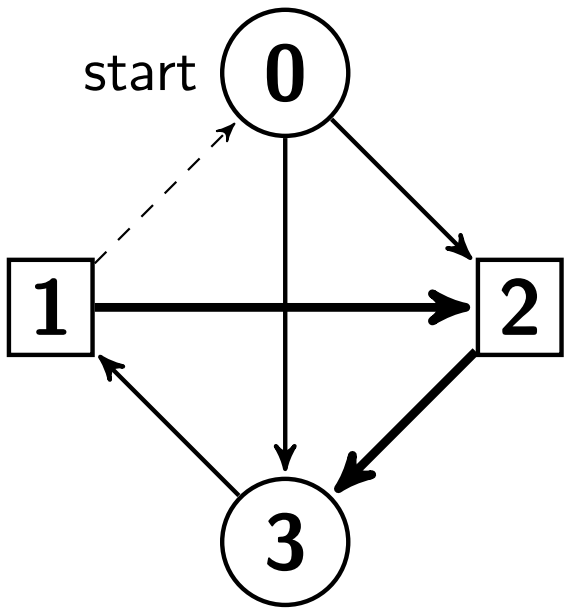
Winning positional strategies are recognizable in polynomial time.

Complexity of the decision problem - 1

Winning positional strategies are recognizable in polynomial time.

Algorithm

Suppose we are given a positional strategy τ for Player 1.
Let E^τ be the edge relation E restricted by τ.
For each vertex x reachable in E^τ from the start vertex, check that, if x is on a loop in E^τ, then the least vertex on the loop belongs to Player 1.



$\{0,3\}$ belong to Player 0

$\{1,2\}$ belong to Player 1

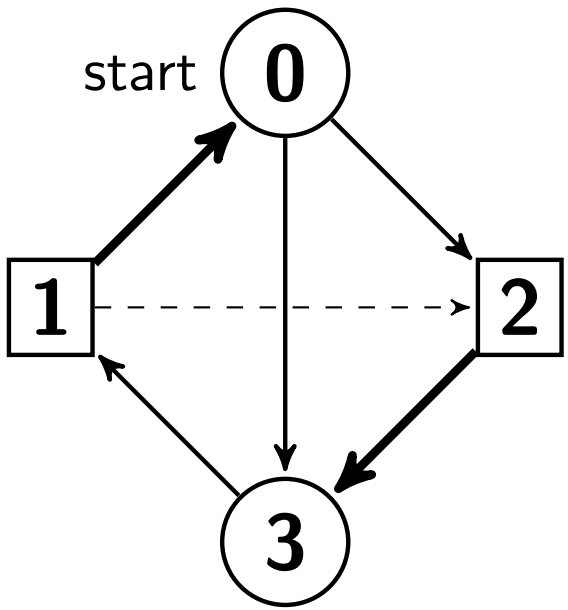
τ is winning for Player 1

Complexity of the decision problem - 1

Winning positional strategies are recognizable in polynomial time.

Algorithm

Suppose we are given a positional strategy τ for Player 1.
Let E^τ be the edge relation E restricted by τ.
For each vertex x reachable in E^τ from the start vertex, check that, if x is on a loop in E^τ, then the least vertex on the loop belongs to Player 1.



$\{0,3\}$ belong to Player 0

$\{1,2\}$ belong to Player 1

τ is losing for Player 1

Complexity of the decision problem - 1

Winning positional strategies are recognizable in polynomial time.

Algorithm

Suppose we are given a positional strategy τ for Player 1.
Let E^τ be the edge relation E restricted by τ.
For each vertex x reachable in E^τ from the start vertex, check that, if x is on a loop in E^τ, then the least vertex on the loop belongs to Player 1.

Corollary

The decision problem is in $NP \cap coNP$.

Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

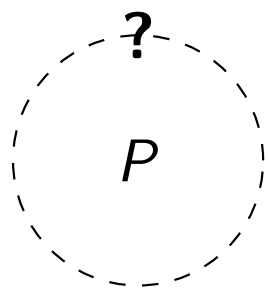
Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

- Not known to be in P



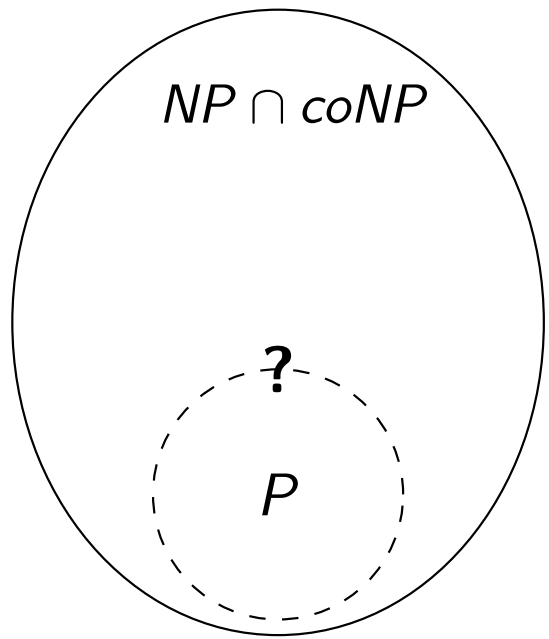
Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

- Not known to be in P
- In $NP \cap coNP$



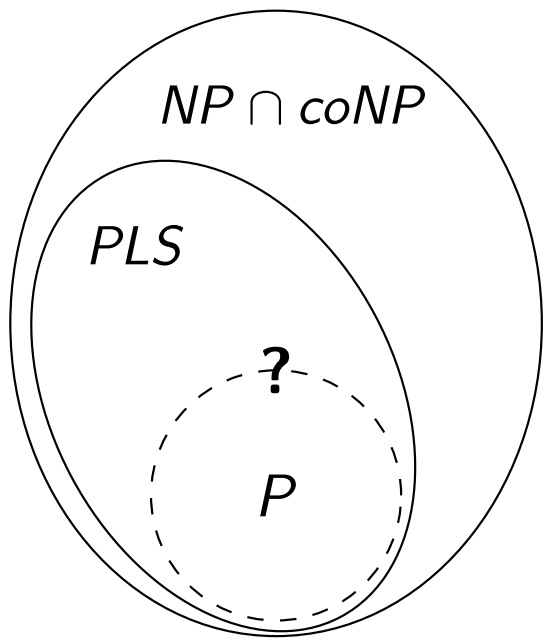
Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

- Not known to be in P
- In $NP \cap coNP$
- Reducible to a problem in PLS



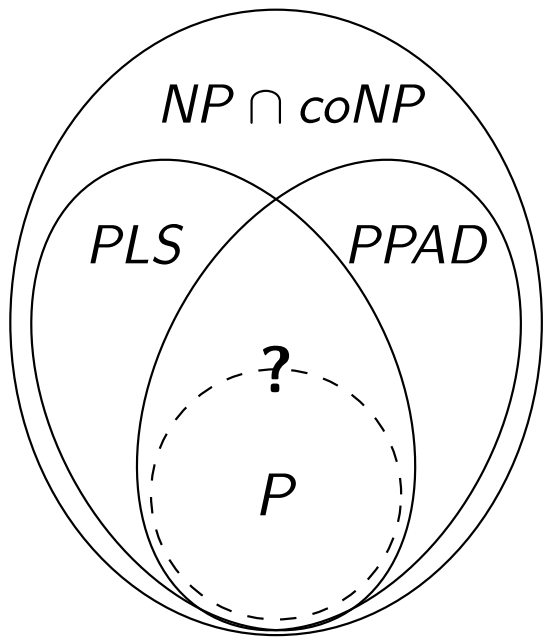
Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

- Not known to be in P
- In $NP \cap coNP$
- Reducible to a problem in PLS
- Reducible to a problem in $PPAD$



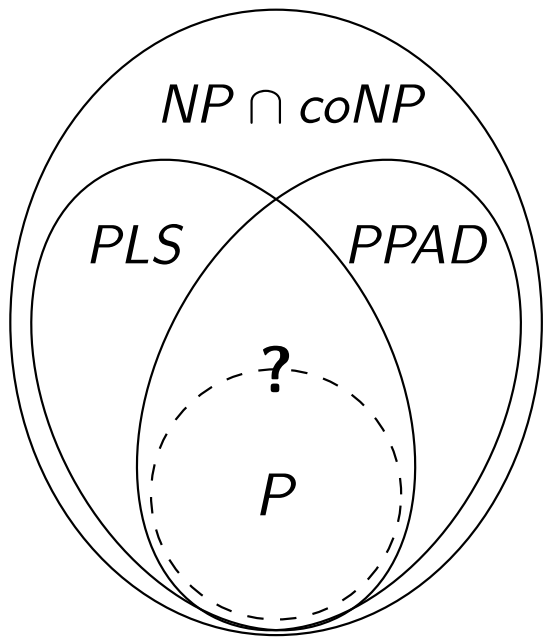
Complexity of the decision problem - 2

The decision problem for parity games

Input: Parity game G

Question: Which player has a positional winning strategy in G?

- Not known to be in P
- In $NP \cap coNP$
- Reducible to a problem in PLS
- Reducible to a problem in $PPAD$
- Has many applications in automata theory and model checking, and there is ongoing research into good algorithms for it.



Automatizability

Parity Games

Results

Proof

Main theorem

Theorem

If resolution is weakly automatizable, then the decision problem for parity games is in P.

Main theorem

Theorem

If resolution is weakly automatizable, then the decision problem for parity games is in P.

This is additional evidence that resolution is not weakly automatizable, or, at least, that weak automatizability of resolution is interesting.

Main theorem

Theorem

If resolution is weakly automatizable, then the decision problem for parity games is in P.

This is additional evidence that resolution is not weakly automatizable, or, at least, that weak automatizability of resolution is interesting.

On the other hand, SAT solvers seem to work well in practice.

Converse?

Does the existence of a fast algorithm deciding parity games imply that resolution is weakly automatizable?

Converse?

Does the existence of a fast algorithm deciding parity games imply that resolution is weakly automatizable?

We expect that the answer is no.

A game with a converse

In the paper we define a new kind of two-player game on a finite graph, the *point-line game*.

We show that resolution is weakly automatizable if and only if you can separate, in polynomial time, the games in which the first player has a positional winning strategy from those in which the second player has a positional winning strategy.

Automatizability

Parity Games

Results

Proof

Reminder

Decision problem for parity games

Input: A parity game G

Task: Output 0 if Player 0 has a positional winning strategy.
Output 1 if Player 1 has a positional winning strategy.

Reminder

Decision problem for parity games

Input: A parity game G

Task: Output 0 if Player 0 has a positional winning strategy.
Output 1 if Player 1 has a positional winning strategy.

(From now on we will omit the word "positional")

Reminder

Decision problem for parity games

Input: A parity game G

Task: Output 0 if Player 0 has a positional winning strategy.
Output 1 if Player 1 has a positional winning strategy.

(From now on we will omit the word "positional")

Weak automatizability of resolution

Input: A CNF F and a string 1^m

Task: Output 0 if F is satisfiable.
Output 1 if F has a resolution refutation of size $\leq m$.
In any other case, output anything.

Reminder

Decision problem for parity games

Input: A parity game G

Task: Output 0 if Player 0 has a positional winning strategy.
Output 1 if Player 1 has a positional winning strategy.

(From now on we will omit the word "positional")

Weak automatizability of resolution

Input: A CNF F and a string 1^m

Task: Output 0 if F is satisfiable.
Output 1 if F has a resolution refutation of size $\leq m$.
In any other case, output anything.

Suppose we have a polynomial time algorithm $M(F, 1^m)$ that solves the weak automatizability of resolution.

Proof idea

We are given a parity game G.

Find CNF formulas Win_0 and Win_1 with disjoint variables such that

Proof idea

We are given a parity game G.

Find CNF formulas Win_0 and Win_1 with disjoint variables such that

- Win_i is satisfiable if and only if Player i has a winning strategy in G

Proof idea

We are given a parity game G.

Find CNF formulas Win_0 and Win_1 with disjoint variables such that

- Win_i is satisfiable if and only if Player i has a winning strategy in G
- There exists a resolution refutation Π of $\mathrm{Win}_0 \wedge \mathrm{Win}_1$ with size $|\Pi|$ polynomial in $|G|$.

Proof idea

We are given a parity game G.

Find CNF formulas Win_0 and Win_1 with disjoint variables such that

- Win_i is satisfiable if and only if Player i has a winning strategy in G
- There exists a resolution refutation Π of $\mathrm{Win}_0 \wedge \mathrm{Win}_1$ with size $|\Pi|$ polynomial in $|G|$.

If Player 0 has a winning strategy then $M(\mathrm{Win}_0, 1^{|\Pi|}) = 0$.

Proof idea

If Player 0 has a winning strategy then $M(\mathrm{Win}_0, 1^{|\Pi|}) = 0$.

Proof idea

If Player 0 has a winning strategy then $M(\mathrm{Win}_0, 1^{|\Pi|}) = 0$.

Suppose Player 1 has a winning strategy.

Proof idea

If Player 0 has a winning strategy then $M(\mathrm{Win}_0, 1^{|\Pi|}) = 0$.

Suppose Player 1 has a winning strategy.

Then there is a satisfying assignment τ for Win_1.

Proof idea

If Player 0 has a winning strategy then $M(\mathsf{Win}_0, 1^{|\Pi|}) = 0$.

Suppose Player 1 has a winning strategy.

Then there is a satisfying assignment τ for Win_1.

Π is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1$. Substitute in τ.
Then $\Pi[\tau]$ is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1[\tau]$.

Proof idea

If Player 0 has a winning strategy then $M(\mathsf{Win}_0, 1^{|\Pi|}) = 0$.

Suppose Player 1 has a winning strategy.

Then there is a satisfying assignment τ for Win_1.

Π is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1$. Substitute in τ.
Then $\Pi[\tau]$ is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1[\tau]$.

$\mathsf{Win}_1[\tau]$ simplifies to $\top$.
So $\Pi[\tau]$ is a resolution refutation of Win_0.

Proof idea

If Player 0 has a winning strategy then $M(\mathsf{Win}_0, 1^{|\Pi|}) = 0$.

Suppose Player 1 has a winning strategy.

Then there is a satisfying assignment τ for Win_1.

Π is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1$. Substitute in τ.
Then $\Pi[\tau]$ is a resolution refutation of $\mathsf{Win}_0 \wedge \mathsf{Win}_1[\tau]$.

$\mathsf{Win}_1[\tau]$ simplifies to $\top$.
So $\Pi[\tau]$ is a resolution refutation of Win_0.

Hence there exists a resolution refutation of Win_0 of size $\leq |\Pi|$.
Therefore $M(\mathsf{Win}_0, 1^{|\Pi|}) = 1$.

Small correction

Small correction

I am cheating a little.

The refutation Π is actually in the proof system Res(10).
This is slightly stronger than resolution.

Small correction

I am cheating a little.

The refutation Π is actually in the proof system Res(10).
This is slightly stronger than resolution.

Res(k) is like resolution, but each line in the proof is a disjunction of size-k conjunctions, rather than a disjunction of literals.

Small correction

I am cheating a little.

The refutation Π is actually in the proof system Res(10).
This is slightly stronger than resolution.

Res(k) is like resolution, but each line in the proof is a disjunction of size-k conjunctions, rather than a disjunction of literals.

For $k \in \mathbb{N}$,

$$\text{Res}(k) \text{ is weakly automatizable}$$
$$\iff$$
$$\text{resolution is weakly automatizable.}$$

Some details

It is easier to describe strategies if we can also talk about the game. Suppose we want to solve the decision problem for games of size n.

Introduce sets of propositional variables $\widehat{G}, \widehat{\sigma}, \widehat{\tau}$.

Game($\widehat{G}$) is a CNF expressing that $\widehat{G}$ describes a parity game.

Some details

It is easier to describe strategies if we can also talk about the game.
Suppose we want to solve the decision problem for games of size n.

Introduce sets of propositional variables $\widehat{G}, \widehat{\sigma}, \widehat{\tau}$.

$\mathrm{Game}(\widehat{G})$ is a CNF expressing that $\widehat{G}$ describes a parity game.

Find a short refutation Π of $\mathrm{Game}(\widehat{G}) \wedge \mathrm{Win}_0(\widehat{G}, \widehat{\sigma}) \wedge \mathrm{Win}_1(\widehat{G}, \widehat{\tau})$.

Given a game G:

If Player 0 has a winning strategy, then $\mathrm{Win}_0(G/\widehat{G}, \widehat{\sigma})$ is satisfiable.

Suppose Player 1 has a winning strategy τ.
Then $\mathrm{Win}_1(G/\widehat{G}, \tau/\widehat{\tau})$ and $\mathrm{Game}(G/\widehat{G})$ both simplify to $\top$.
Hence $\Pi[G/\widehat{G}, \tau/\widehat{\tau}]$ is a refutation of $\mathrm{Win}_0(G/\widehat{G}, \widehat{\sigma})$.

CNF for a game

$\widehat{G}$ is a set of propositional variables describing a parity game with vertices $[n]$. It consists of variables

$\{e_{ij} : i, j \in [n]\}$ expressing which edges are in E
$\{v_i^0 : i \in [n]\}$ expressing whether i belongs to Player 0
$\{v_i^1 : i \in [n]\}$ expressing whether i belongs to Player 1
$\{s_i : i \in [n]\}$ expressing whether i is the start vertex.

Game($\widehat{G}$) asserts that

- Every vertex has at least one outgoing edge
- Every vertex belongs to exactly one of the players
- There is exactly one start vertex.

This can be done with a CNF formula.

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

- If E^σ is the edge relation E of G restricted by σ, then for each vertex x reachable in E^σ from the start vertex, if x is on a loop in E^σ, then the least vertex on the loop belongs to Player 0.

CNF for a strategy - doesn't work

$\widehat{\sigma}$ is a set of propositional variables describing a strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ.

$\mathrm{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

- If E^σ is the edge relation E of G restricted by σ, then for each vertex x reachable in E^σ from the start vertex, if x is on a loop in E^σ, then the least vertex on the loop belongs to Player 0.

We do not know how to express this with a CNF formula.

CNF for a strategy

A *rich strategy* is a pair of

CNF for a strategy

A *rich strategy* is a pair of

1. A strategy σ

CNF for a strategy

A *rich strategy* is a pair of

1. A strategy σ
2. A relation $R^\sigma(x, y, z)$ which is true if and only if there is a non-trivial path in E^σ from x to y on which the least vertex visited is z.

CNF for a strategy

A *rich strategy* is a pair of

1. A strategy σ
2. A relation $R^\sigma(x, y, z)$ which is true if and only if there is a non-trivial path in E^σ from x to y on which the least vertex visited is z.

Whenever a strategy exists, a rich strategy also exists.

Therefore we can replace "strategy" by "rich strategy" everywhere in our argument, and the argument will still work.

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a (rich) strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ

$\{r_{ijk} : i, j, k \in [n]\}$ expressing the relation $R^{\sigma}(i, j, k)$.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a (rich) strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ

$\{r_{ijk} : i, j, k \in [n]\}$ expressing the relation $R^{\sigma}(i, j, k)$.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

- If E^{σ} is the edge relation E of G restricted by σ, then $R^{\sigma}(i, j, k)$ describes the correct reachability property

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a (rich) strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ

$\{r_{ijk} : i, j, k \in [n]\}$ expressing the relation $R^{\sigma}(i, j, k)$.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

- If E^{σ} is the edge relation E of G restricted by σ, then $R^{\sigma}(i, j, k)$ describes the correct reachability property
- For each vertex x reachable in E^{σ} from the start vertex, if x is on a loop in E^{σ}, then the least vertex on the loop belongs to Player 0.

CNF for a strategy

$\widehat{\sigma}$ is a set of propositional variables describing a (rich) strategy σ for Player 0 in a game with vertices $[n]$. It consists of variables

$\{S_{ij} : i, j \in [n]\}$ expressing which edges are chosen in σ

$\{r_{ijk} : i, j, k \in [n]\}$ expressing the relation $R^\sigma(i, j, k)$.

$\mathsf{Win}_0(\widehat{G}, \widehat{\sigma})$ asserts that

- If E^σ is the edge relation E of G restricted by σ, then $R^\sigma(i, j, k)$ describes the correct reachability property
- For each vertex x reachable in E^σ from the start vertex, if x is on a loop in E^σ, then the least vertex on the loop belongs to Player 0.

This can be done with a CNF formula.

CNF for a strategy

$\mathrm{Win}_1(\widehat{G}, \widehat{\tau})$ is a similar CNF expressing that $\widehat{\tau}$ describes a (rich) winning strategy for Player 1 in G.

Remaining problem

We have to show that

$$\mathsf{Game}(\widehat{G}) \wedge \mathsf{Win}_0(\widehat{G}, \widehat{\sigma}) \wedge \mathsf{Win}_1(\widehat{G}, \widehat{\tau})$$

has a small $\mathrm{Res}(k)$ refutation.

Remaining problem

We have to show that

$$\mathsf{Game}(\widehat{G}) \wedge \mathsf{Win}_0(\widehat{G}, \widehat{\sigma}) \wedge \mathsf{Win}_1(\widehat{G}, \widehat{\tau})$$

has a small $\mathrm{Res}(k)$ refutation.

We construct the refutation indirectly, by showing that a first-order version of the formula is refutable using only bounded Π_2 induction.

Remaining problem

We have to show that

$$\mathsf{Game}(\widehat{G}) \wedge \mathsf{Win}_0(\widehat{G}, \widehat{\sigma}) \wedge \mathsf{Win}_1(\widehat{G}, \widehat{\tau})$$

has a small $\mathrm{Res}(k)$ refutation.

We construct the refutation indirectly, by showing that a first-order version of the formula is refutable using only bounded Π_2 induction.

This is made easier by our inclusion of the reachability relations in σ and τ.

How to construct propositional refutations

A U_2 *formula* is a first-order formula of the form

$$\forall \bar{u} < p(x)\, \exists \bar{v} < q(x)\, \phi(x, \bar{u}, \bar{v})$$

where p and q are polynomials and $\phi(x, \bar{u}, \bar{v})$ is quantifier free.

How to construct propositional refutations

A U_2 *formula* is a first-order formula of the form

$$\forall \bar{u} < p(x)\, \exists \bar{v} < q(x)\, \phi(x, \bar{u}, \bar{v})$$

where p and q are polynomials and $\phi(x, \bar{u}, \bar{v})$ is quantifier free.

This is analogous to a polynomial-size propositional CNF.

How to construct propositional refutations

A U_2 *formula* is a first-order formula of the form

$$\forall \bar{u} < p(x)\, \exists \bar{v} < q(x)\, \phi(x, \bar{u}, \bar{v})$$

where p and q are polynomials and $\phi(x, \bar{u}, \bar{v})$ is quantifier free.

This is analogous to a polynomial-size propositional CNF.

Theorem

Let $\Phi(x)$ be a U_2 formula. Suppose that $\forall x \neg \Phi(x)$ is provable in first-order logic using limited induction, where we only allow U_2 formulas as induction hypotheses. Then the propositional CNFs F_n expressing $\Phi(n)$ have polynomial-sized Res(k) refutations for some $k \in \mathbb{N}$.

How to construct propositional refutations

A U_2 *formula* is a first-order formula of the form

$$\forall \bar{u} < p(x)\, \exists \bar{v} < q(x)\, \phi(x, \bar{u}, \bar{v})$$

where p and q are polynomials and $\phi(x, \bar{u}, \bar{v})$ is quantifier free.

This is analogous to a polynomial-size propositional CNF.

Theorem

Let $\Phi(x)$ be a U_2 formula. Suppose that $\forall x \neg \Phi(x)$ is provable in first-order logic using limited induction, where we only allow U_2 formulas as induction hypotheses. Then the propositional CNFs F_n expressing $\Phi(n)$ have polynomial-sized Res(k) refutations for some $k \in \mathbb{N}$.

Forms of this theorem go back to [Paris and Wilkie 1985].
A version for Res(k) was shown in [Krajíček 1994, 1997, 2001].

Proof idea

We are given a parity game G and rich strategies σ, τ satisfying Win_0 and Win_1.

Proof idea

We are given a parity game G and rich strategies σ, τ satisfying Win_0 and Win_1.

Idea: In the graph $E^\sigma \cap E^\tau$, if we start from the start vertex we will eventually reach some point t on a loop.

Let v be the least vertex on this loop.

Then by Win_0, vertex v belongs to V_0. By Win_1, vertex v belongs to V_1. This is a contradiction.

Proof idea

We are given a parity game G and rich strategies σ, τ satisfying Win_0 and Win_1.

Idea: In the graph $E^\sigma \cap E^\tau$, if we start from the start vertex we will eventually reach some point t on a loop.

Let v be the least vertex on this loop.

Then by Win_0, vertex v belongs to V_0. By Win_1, vertex v belongs to V_1. This is a contradiction.

We cannot use this idea directly, because we cannot express reachability in $E^\sigma \cap E^\tau$ using first-order logic.

Proof idea

We are given a parity game G and rich strategies σ, τ satisfying Win_0 and Win_1.

Idea: In the graph $E^\sigma \cap E^\tau$, if we start from the start vertex we will eventually reach some point t on a loop.

Let v be the least vertex on this loop.

Then by Win_0, vertex v belongs to V_0. By Win_1, vertex v belongs to V_1. This is a contradiction.

We cannot use this idea directly, because we cannot express reachability in $E^\sigma \cap E^\tau$ using first-order logic.

Instead we use the reachability relation

$$\exists v,\ R^\sigma(x, y, v) \wedge R^\tau(x, y, v).$$