Easy to Win, Hard to Master:
Optimal Strategies in Parity Games with Costs

Joint work with Martin Zimmermann

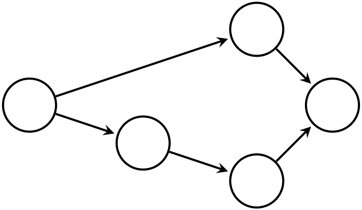
Alexander Weinert

Saarland University

August 31st, 2016

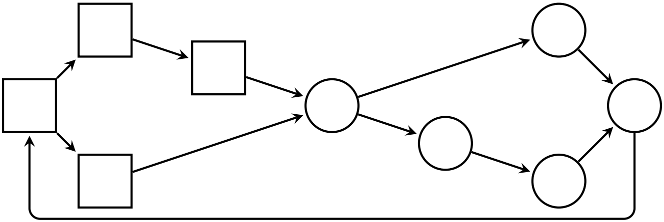
CSL 2016 - Marseille, France

Parity Games



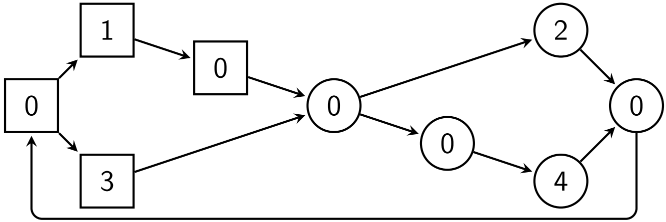
Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games



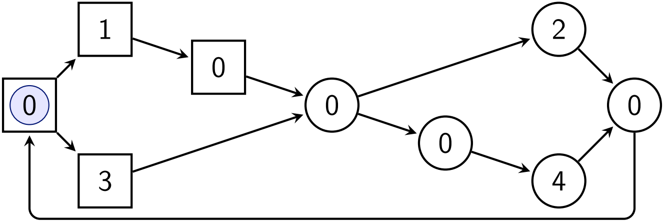
Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games



Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

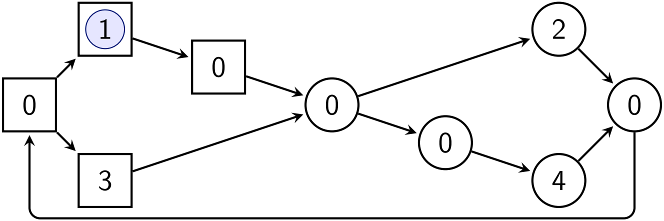
Parity Games



0

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

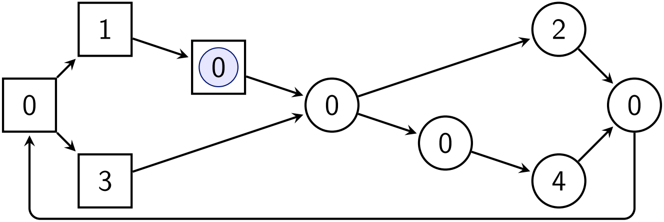
Parity Games



$0 \longrightarrow 1$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

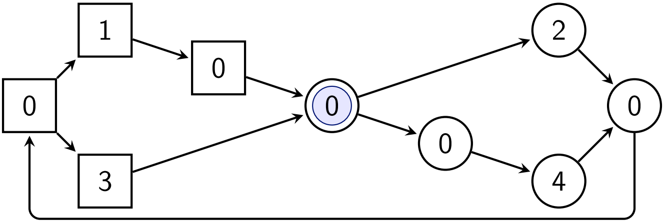
Parity Games



$0 \longrightarrow 1 \longrightarrow 0$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

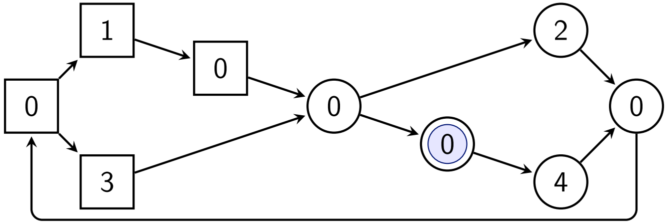
Parity Games



$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

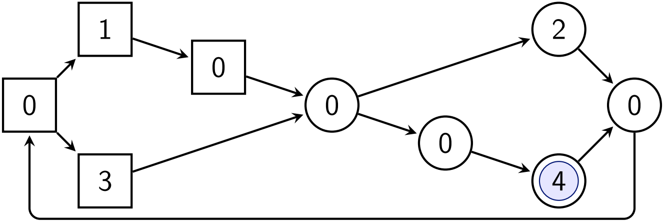
Parity Games



$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

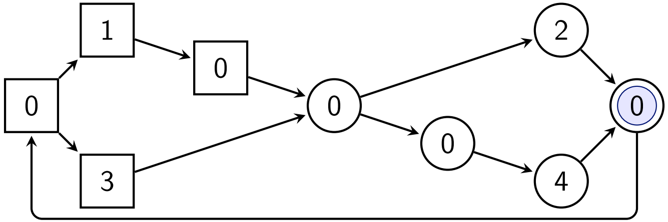
Parity Games



$$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow 4$$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

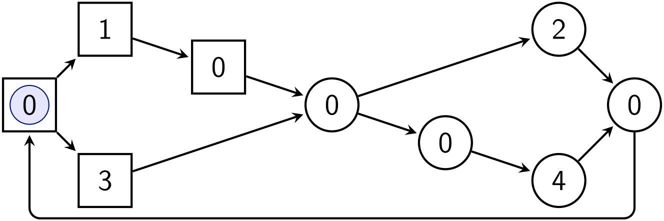
Parity Games



$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow 4 \longrightarrow 0$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games

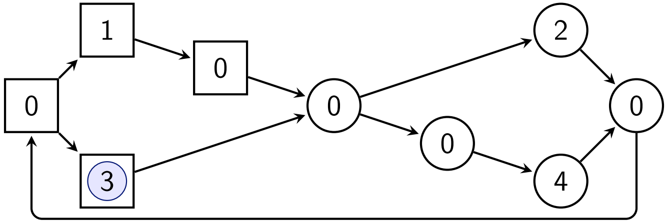


$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow 4 \longrightarrow 0$

$\longrightarrow 0$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games

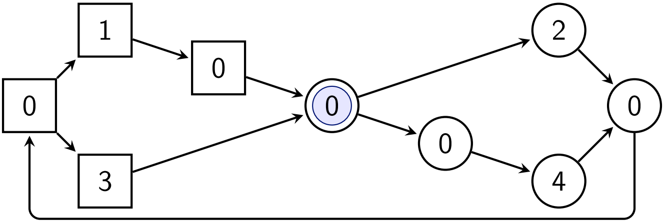


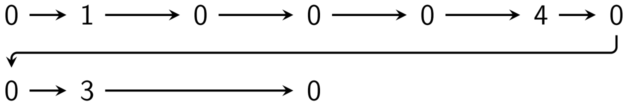
$0 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow 4 \longrightarrow 0$

$0 \longrightarrow 3$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

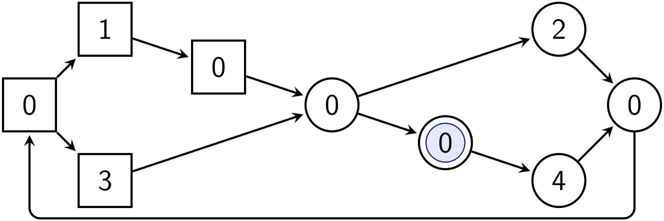
Parity Games

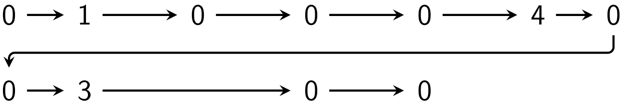




Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

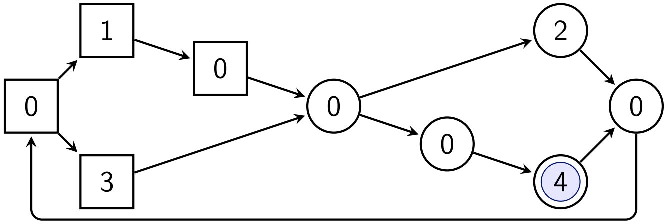
Parity Games





Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

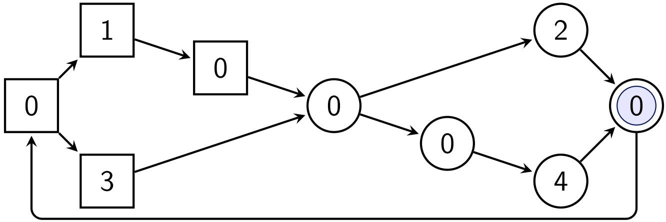
Parity Games



$$0 \to 1 \to 0 \to 0 \to 0 \to 4 \to 0$$
$$\to 0 \to 3 \to 0 \to 0 \to 4$$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games

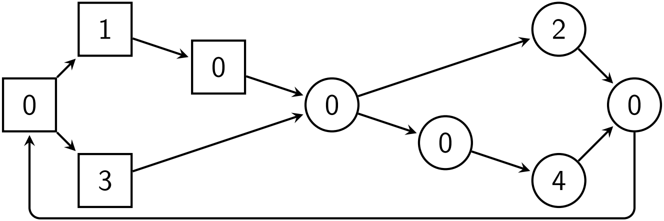


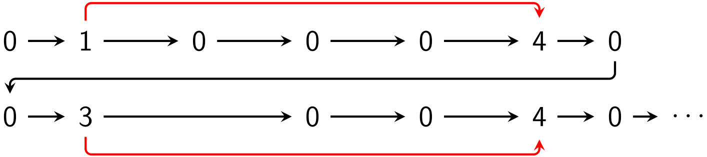
$$0 \to 1 \to 0 \to 0 \to 0 \to 4 \to 0$$

$$0 \to 3 \to 0 \to 0 \to 4 \to 0 \to \cdots$$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

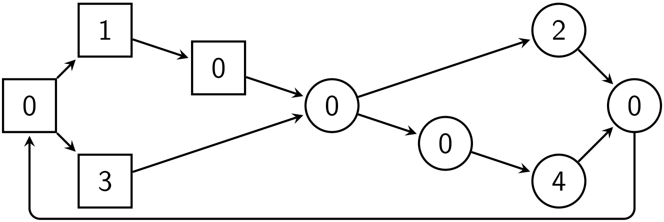
Parity Games

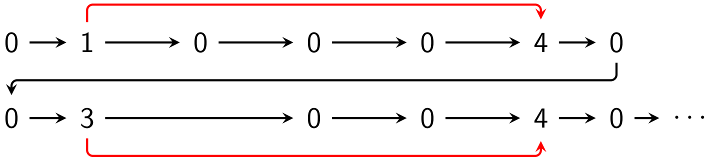




Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Parity Games



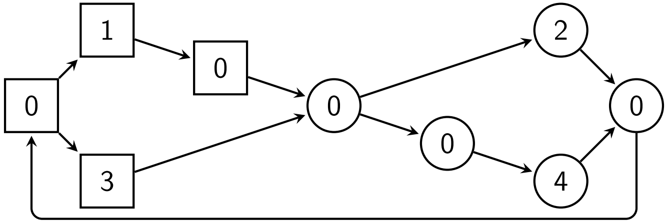


Deciding winner in $\mathrm{UP} \cap \mathrm{co\text{-}UP}$

Positional Strategies

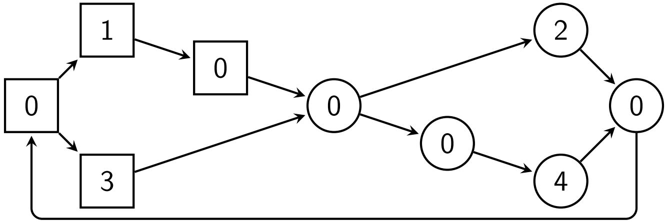
Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Finitary Parity Games



Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Finitary Parity Games

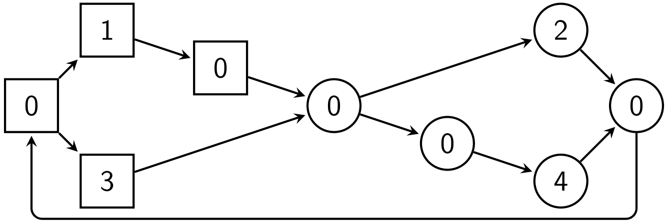


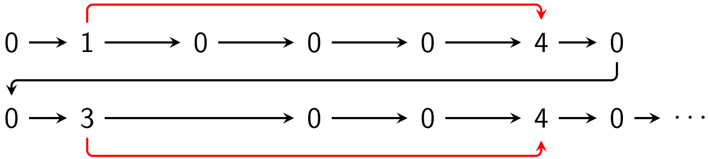
$$0 \to 1 \to 0 \to 0 \to 0 \to 4 \to 0$$

$$0 \to 3 \to 0 \to 0 \to 4 \to 0 \to \cdots$$

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

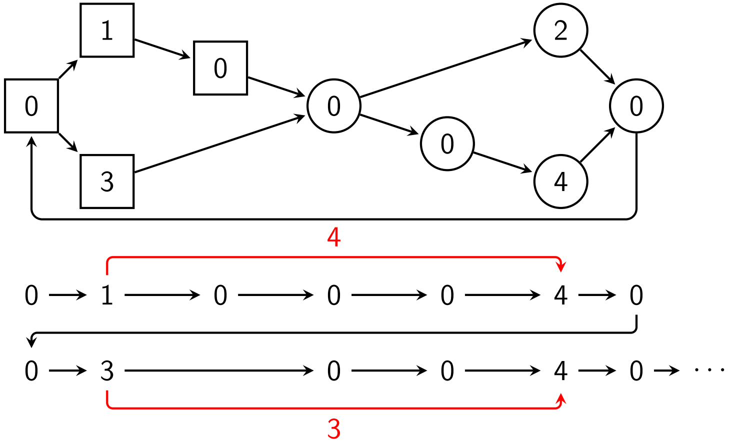
Finitary Parity Games





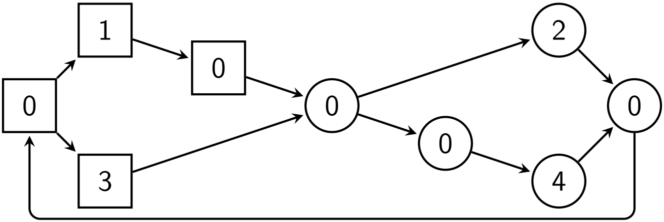
Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

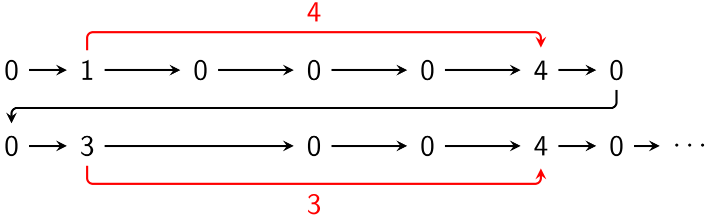
Finitary Parity Games



Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Finitary Parity Games





Goal for Player 0: Bound response times

Example due to (Fijalkow and Chatterjee, Infinite-state games, 2013)

Decision Problem

Theorem (Chatterjee et al., Finitary Winning, 2009)

The following decision problem is in $\textsc{PTime}$*:*

Input: *Finitary parity game* $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$

Question: *Does there exist a strategy* σ *with* $\mathrm{Cst}(\sigma) < \infty$*?*

Decision Problem

Theorem (Chatterjee et al., Finitary Winning, 2009)

The following decision problem is in $\textsc{PTime}$*:*

Input: *Finitary parity game* $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$

Question: *Does there exist a strategy* σ *with* $\mathrm{Cst}(\sigma) < \infty$*?*

Theorem

The following decision problem is $\textsc{PSpace}$*-complete:*

Input: *Finitary parity game* $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$*,*
bound $b \in \mathbb{N}$

Question: *Does there exist a strategy* σ *with* $\mathrm{Cst}(\sigma) \leq b$*?*

Introduction

Introduction ✓

Introduction ✓

↓

Complexity

in PSpace

Introduction ✓

↓

Complexity

in PSPACE

PSPACE-hard

Introduction ✓

Complexity

in PSPACE

PSPACE-hard

Exponential Memory

Sufficient

Introduction ✓

Complexity

in PSpace

PSpace-hard

Exponential Memory

Sufficient

Introduction ✓

Complexity

in PSpace

PSpace-hard

Exponential Memory

Sufficient

Necessary

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS{\scriptstyle PACE}}$.

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS{\small PACE}}$.

Idea: Simulate game, keeping track of open requests.

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS}\textsc{pace}$.

Idea: Simulate game, keeping track of open requests.

Lemma

Player 0 has such a strategy iff she "survives" $p(|\mathcal{G}|)$ steps in extended game $\mathcal{G}'$.

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in PSPACE.

Idea: Simulate game, keeping track of open requests.

Lemma

Player 0 has such a strategy iff she "survives" $p(|\mathcal{G}|)$ steps in extended game $\mathcal{G}'$.

Algorithm:

Simulate all plays in $\mathcal{G}'$ on-the-fly for $p(|\mathcal{G}|)$ steps

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS{\small PACE}}$.

Idea: Simulate game, keeping track of open requests.

Lemma

Player 0 has such a strategy iff she "survives" $p(|\mathcal{G}|)$ steps in extended game $\mathcal{G}'$.

Algorithm:

Simulate all plays in $\mathcal{G}'$ on-the-fly for $p(|\mathcal{G}|)$ steps using an alternating Turing machine.

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS{\scriptsize PACE}}$.

Idea: Simulate game, keeping track of open requests.

Lemma

Player 0 has such a strategy iff she "survives" $p(|\mathcal{G}|)$ steps in extended game $\mathcal{G}'$.

Algorithm:

Simulate all plays in $\mathcal{G}'$ on-the-fly for $p(|\mathcal{G}|)$ steps using an alternating Turing machine.

$\Rightarrow$ Problem is in $\mathrm{APT{\scriptsize IME}}$

From Finitary Parity to Parity

Given: Finitary parity game $\mathcal{G} = (\mathcal{A}, \mathrm{FinParity}(\Omega))$, bound $b \in \mathbb{N}$.

Lemma

Deciding if Player 0 has strategy σ with $\mathrm{Cst}(\sigma) \leq b$ is in $\mathrm{PS}\textsc{pace}$.

Idea: Simulate game, keeping track of open requests.

Lemma

Player 0 has such a strategy iff she "survives" $p(|\mathcal{G}|)$ steps in extended game $\mathcal{G}'$.

Algorithm:

Simulate all plays in $\mathcal{G}'$ on-the-fly for $p(|\mathcal{G}|)$ steps using an alternating Turing machine.

$\Rightarrow$ Problem is in $\mathrm{APT}\textsc{ime}$

(Chandra et al., Alternation, 1981)

$\Rightarrow$ Problem is in $\mathrm{PS}\textsc{pace}$

Introduction ✓

Complexity

in PSpace

PSpace-hard

Exponential Memory

Sufficient

Necessary

Introduction ✓

Complexity

in PSPACE ✓ | PSPACE-hard

Exponential Memory

Sufficient | Necessary

PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

PSPACE-completeness

Lemma

The given decision problem is $\textsc{PSpace}$*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$

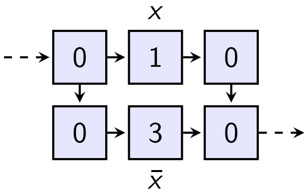
PSPACE-completeness

Lemma

The given decision problem is $\textsc{PSpace}$*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\boxed{\forall x}\ \exists y\ .\ (\ x\ \vee\ \neg y\ ) \wedge (\ \neg x\ \vee\ y\ )$$



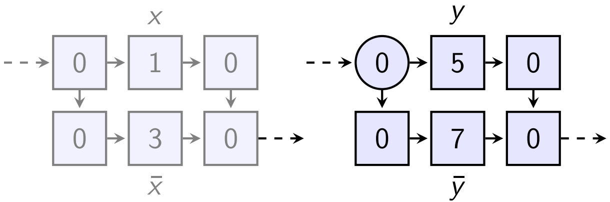
PSPACE-completeness

Lemma

The given decision problem is $\textsc{PSpace}$*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x\ \exists y.\ (x \vee \neg y) \wedge (\neg x \vee y)$$



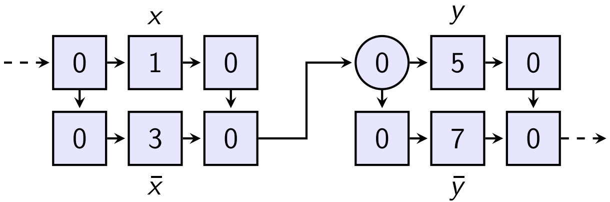
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$



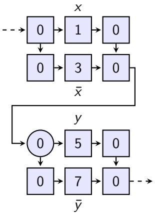
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$



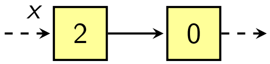
PSPACE-completeness

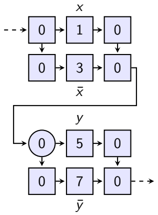
Lemma

The given decision problem is PSpace*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x\ \exists y\ .\ (\ x \vee \neg y\ ) \wedge (\ \neg x \vee y\ )$$





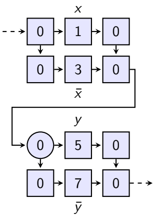
PSPACE-completeness

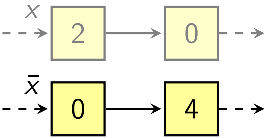
Lemma

The given decision problem is $\mathrm{PS{\small PACE}}$*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x\ \exists y\ .\ (\ x\ \vee\ \neg y\ ) \wedge (\ \neg x\ \vee\ y\ )$$





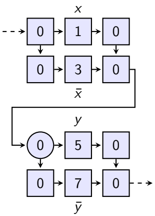
PSPACE-completeness

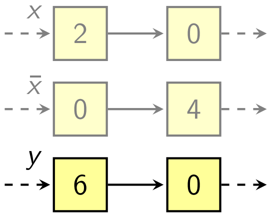
Lemma

The given decision problem is $\mathrm{PS{\scriptsize PACE}}$*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x\ \exists y\ .\ (\ x\ \vee\ \neg y\ ) \wedge (\ \neg x\ \vee\ y\ )$$



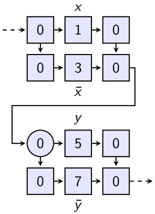


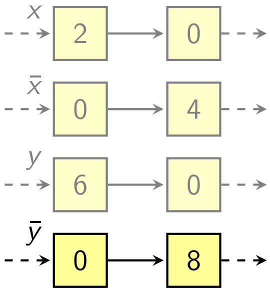
Lemma

The given decision problem is PSPACE-hard.

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (x \lor \neg y) \land (\neg x \lor y)$$



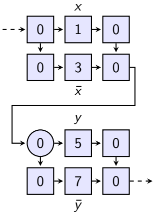


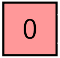
Lemma

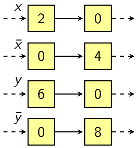
The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \;\; \exists y \; . \; (\; x \; \vee \; \neg y \;) \wedge (\; \neg x \; \vee \; y \;)$$







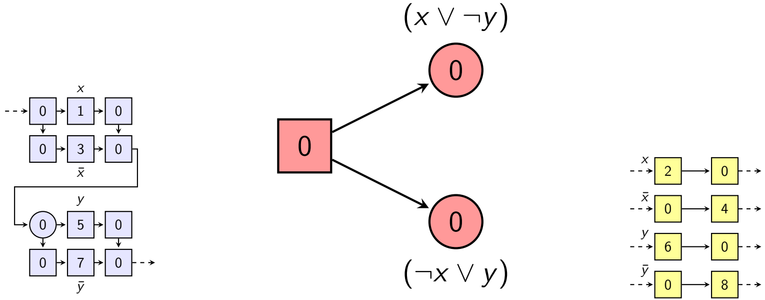
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$



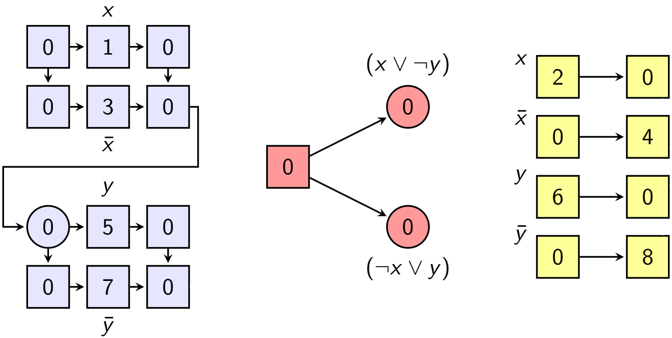
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$



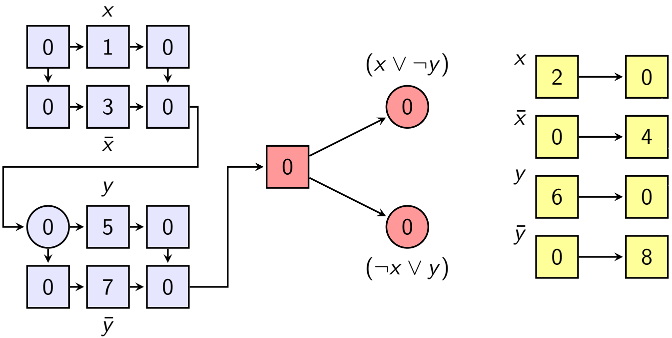
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y\ ) \wedge (\ \neg x \ \vee \ y\ )$$



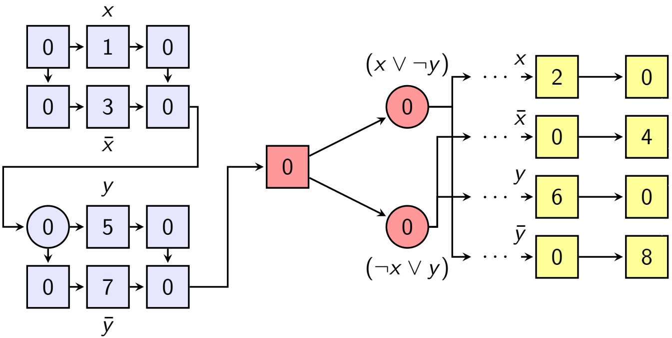
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y\ ) \wedge (\ \neg x \ \vee \ y\ )$$



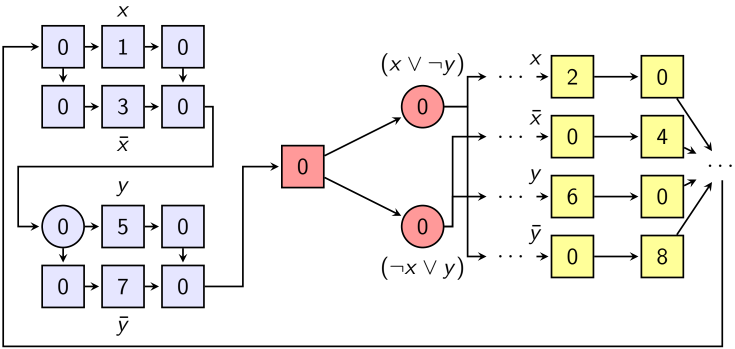
PSPACE-completeness

Lemma

The given decision problem is PSPACE*-hard.*

Idea: Reduction from Quantified Boolean Formulas, e.g.:

$$\forall x \ \exists y \ . \ (\ x \ \vee \ \neg y \ ) \wedge (\ \neg x \ \vee \ y \ )$$



Introduction ✓

Complexity

in PSpace ✓

PSpace-hard

Exponential Memory

Sufficient

Necessary

Introduction ✓

Complexity

in PSPACE ✓

PSPACE-hard ✓

Exponential Memory

Sufficient

Necessary

Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of $\textsc{PSpace}$-membership

Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:

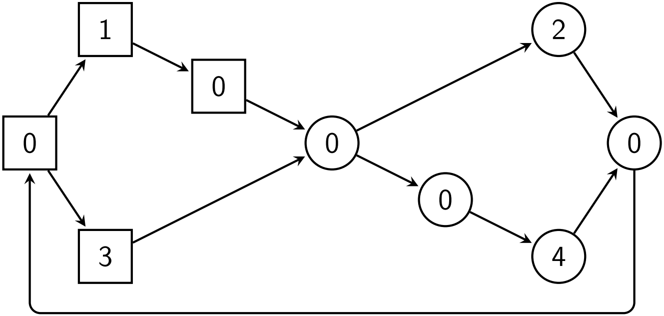
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



(Fijalkow and Chatterjee, Infinite-state games, 2013)

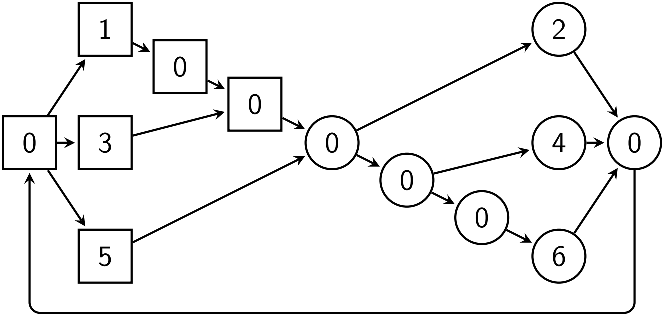
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



(Fijalkow and Chatterjee, Infinite-state games, 2013)

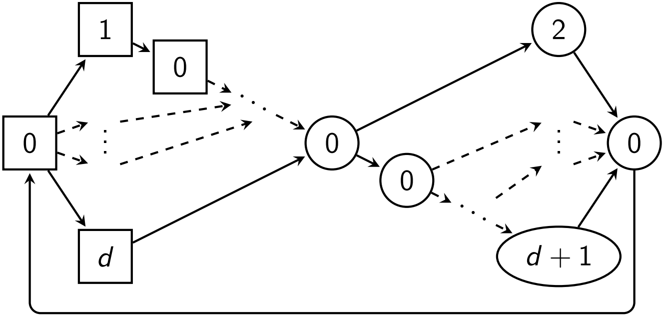
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



(Fijalkow and Chatterjee, Infinite-state games, 2013)

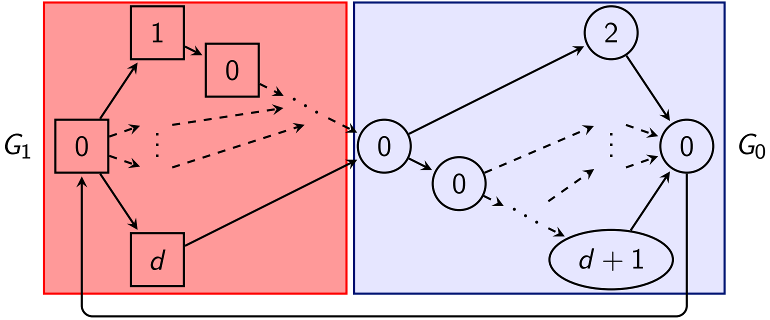
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



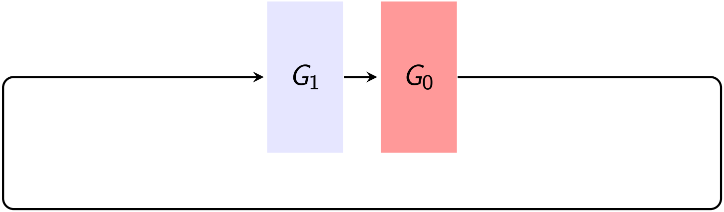
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



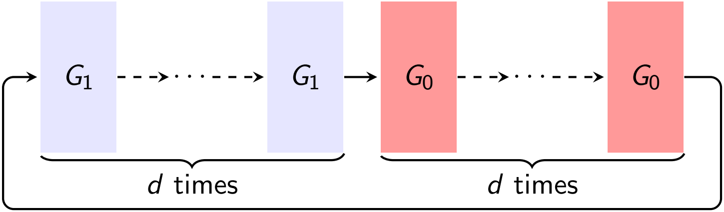
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



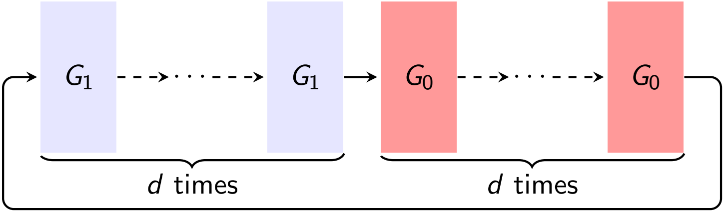
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



Player 0 needs to store d choices of d possible values each

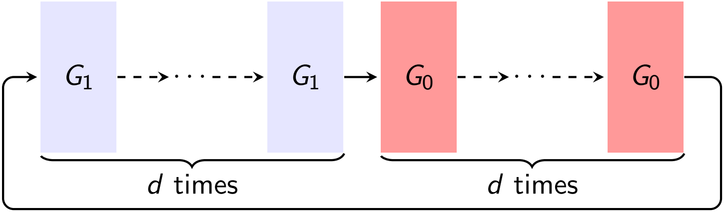
Memory Requirements (for Player 0)

Theorem

Optimal strategies for parity games require exponential memory.

Sufficiency: Corollary of proof of PSPACE-membership

Necessity: Construct family $\mathcal{G}_d$:



Player 0 needs to store d choices of d possible values each
$\Rightarrow$ Player 0 requires $\approx 2^d$ many memory states

Introduction ✓

Complexity

in PSpace ✓

PSpace-hard ✓

Exponential Memory

Sufficient

Necessary

Introduction ✓

Complexity

in PSpace ✓

PSpace-hard ✓

Exponential Memory

Sufficient ✓

Necessary ✓

Conclusion

	Parity
Complexity	$\mathrm{UP} \cap \mathrm{co}\text{-}\mathrm{UP}$
Strategies	1

Conclusion

	Parity	Finitary Parity	
		Winning	
Complexity	$\mathrm{UP} \cap \textsc{co}\text{-}\mathrm{UP}$	$\textsc{PTime}$	
Strategies	1	1	

Conclusion

	Parity	Finitary Parity	
		Winning	Optimal
Complexity	$\mathrm{UP} \cap \mathrm{co}\text{-}\mathrm{UP}$	PTime	PSpace-comp.
Strategies	1	1	Exp.

Conclusion

	Parity	Finitary Parity	
		Winning	Optimal
Complexity	$\mathrm{UP} \cap \text{co-}\mathrm{UP}$	PTime	PSpace-comp.
Strategies	1	1	Exp.

Take-away: Forcing Player 0 to answer quickly in parity games makes it harder

- to decide whether she can satisfy the bound
- for her to play the game

Introduction ✓

Complexity

in PSpace ✓

PSpace-hard ✓

Exponential Memory

Sufficient ✓

Necessary ✓

Introduction ✓

Complexity

in PSpace ✓ | PSpace-hard ✓

Exponential Memory

Sufficient ✓ | Necessary ✓

Tradeoffs

Introduction ✓

Complexity

in PSpace ✓

PSpace-hard ✓

Exponential Memory

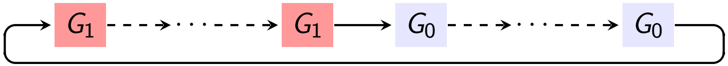
Sufficient ✓

Necessary ✓

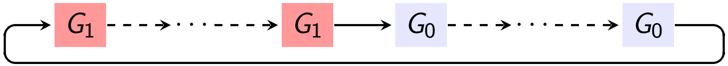
Tradeoffs

Parity Games with Costs

Tradeoffs

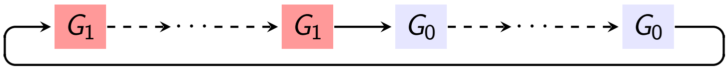


Tradeoffs



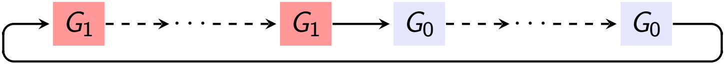
	Winning	Optimal
Size	1	2^d
Cost	$3d$	$2d$

Tradeoffs



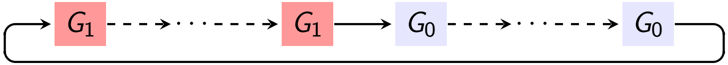
	Winning			Optimal
Size	1	d		2^d
Cost	$3d$			$2d$

Tradeoffs



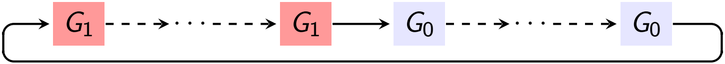
	Winning			Optimal
Size	1	d		2^d
Cost	$3d$	$3d-1$		$2d$

Tradeoffs



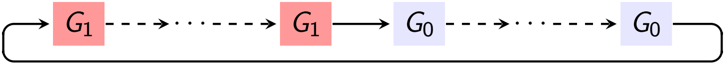
	Winning			Optimal
Size	1	d	2^{d-1}	2^d
Cost	$3d$	$3d-1$		$2d$

Tradeoffs



	Winning			Optimal
Size	1	d	2^{d-1}	2^d
Cost	$3d$	$3d-1$	$2d+1$	$2d$

Tradeoffs



	Winning				Optimal
Size	1	d	$\cdots$	2^{d-1}	2^{d}
Cost	$3d$	$3d-1$	$\cdots$	$2d+1$	$2d$

Introduction ✓

Complexity

in PSpace ✓ PSpace-hard ✓

Exponential Memory

Sufficient ✓ Necessary ✓

Tradeoffs

Parity Games with Costs

Introduction ✓

Complexity

in PSpace ✓

PSpace-hard ✓

Exponential Memory

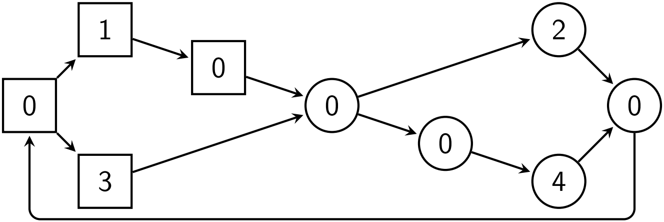
Sufficient ✓

Necessary ✓

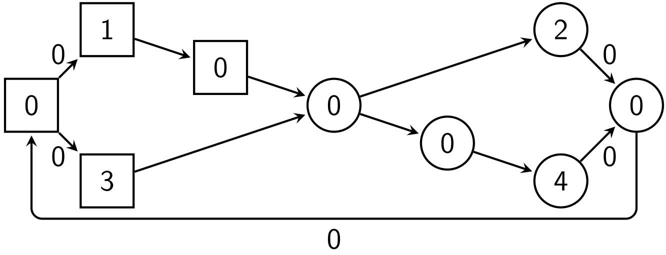
Tradeoffs ✓

Parity Games with Costs

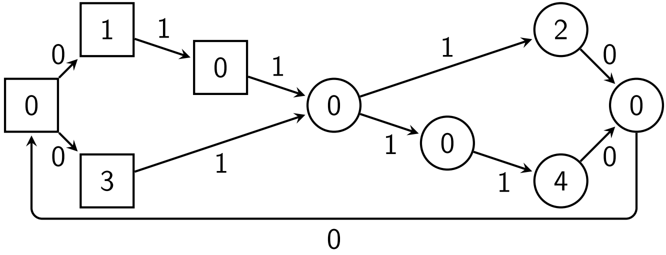
Parity Games with Cost



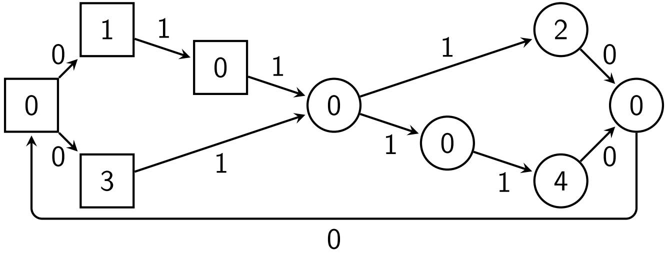
Parity Games with Cost



Parity Games with Cost

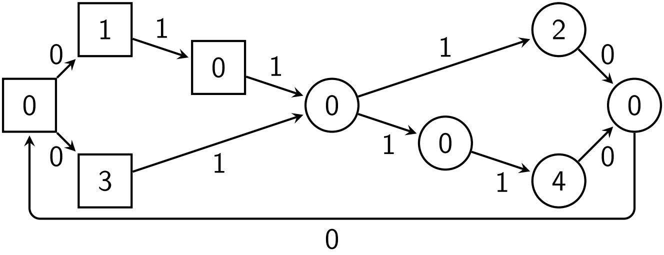


Parity Games with Cost



Finitary parity games are special case

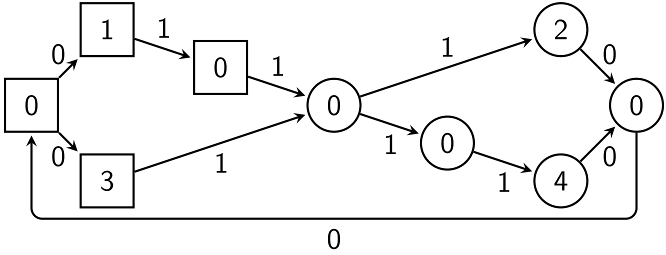
Parity Games with Cost



Finitary parity games are special case

⇒ PSPACE-hard ⇒ Exp. memory necessary

Parity Games with Cost

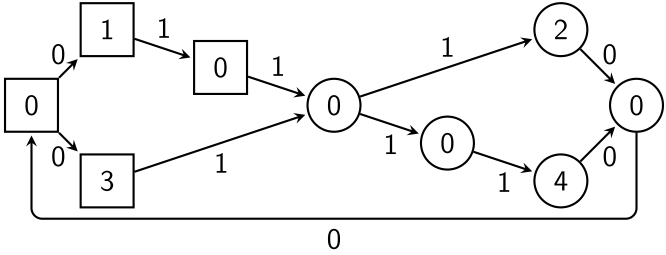


Finitary parity games are special case

$\Rightarrow$ PSPACE-hard $\Rightarrow$ Exp. memory necessary

Algorithm for solving finitary games works as well

Parity Games with Cost



Finitary parity games are special case

$\Rightarrow$ PSPACE-hard $\Rightarrow$ Exp. memory necessary

Algorithm for solving finitary games works as well

$\Rightarrow$ In PSPACE $\Rightarrow$ Exp. memory sufficient

Conclusion

	Parity	
Complexity	$\mathrm{UP} \cap \text{co-}\mathrm{UP}$	
Strategies	1	

Conclusion

	Parity	Cost-Parity
		Winning
Complexity	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$
Strategies	1	1

Conclusion

	Parity	Cost-Parity	
		Winning	Optimal
Complexity	$\mathrm{UP} \cap \mathrm{co}\text{-}\mathrm{UP}$	$\mathrm{UP} \cap \mathrm{co}\text{-}\mathrm{UP}$	$\mathrm{PS}\textsc{pace}$-comp.
Strategies	1	1	Exp.

Conclusion

	Parity	Cost-Parity	
		Winning	Optimal
Complexity	$\mathrm{UP} \cap \textsc{co}\text{-}\mathrm{UP}$	$\mathrm{UP} \cap \textsc{co}\text{-}\mathrm{UP}$	$\mathrm{P}\textsc{Space}$-comp.
Strategies	1	1	Exp.

Take-away: Forcing Player 0 to answer quickly in parity games makes it harder

- to decide whether she can satisfy the bound
- for her to play the game