Parity Games with Weights

Joint work with Sven Schewe (University of Liverpool)
and Martin Zimmermann (Saarland University)

Alexander Weinert

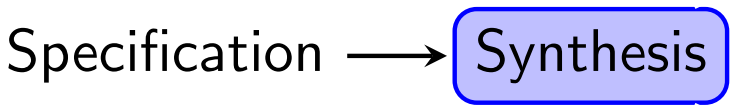
Saarland University

September 6th, 2018

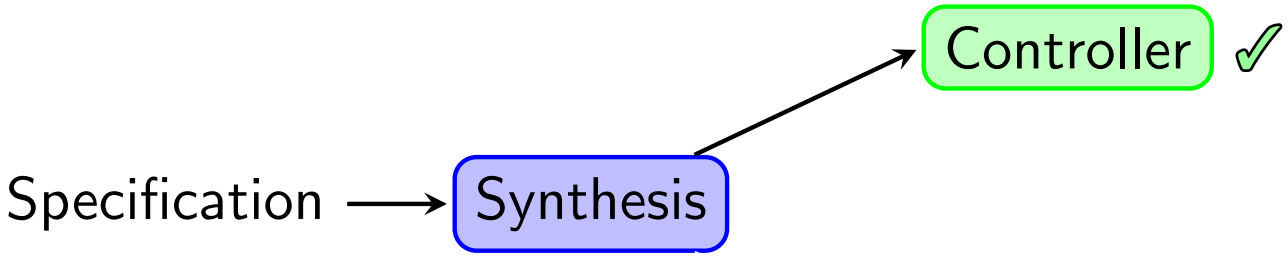
Controller Synthesis

Specification

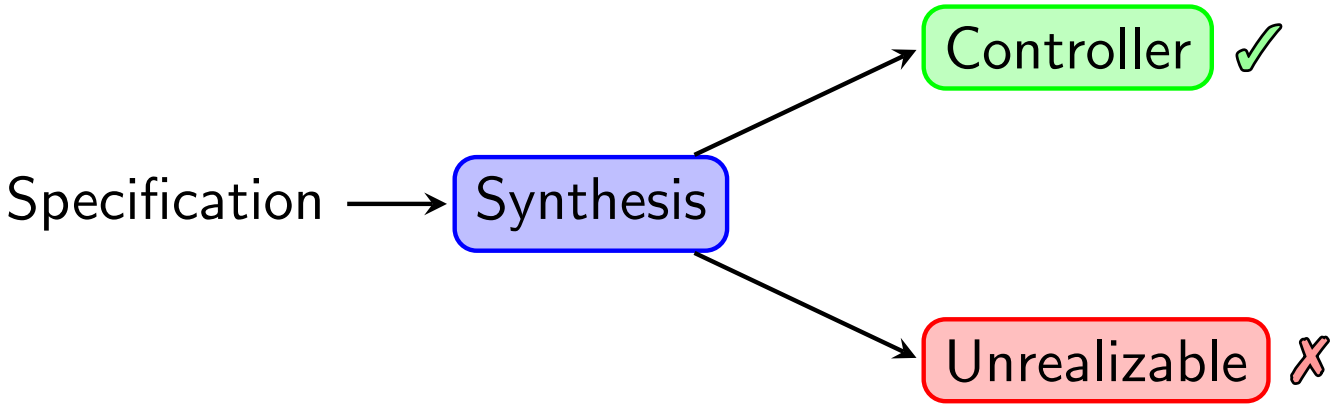
Controller Synthesis



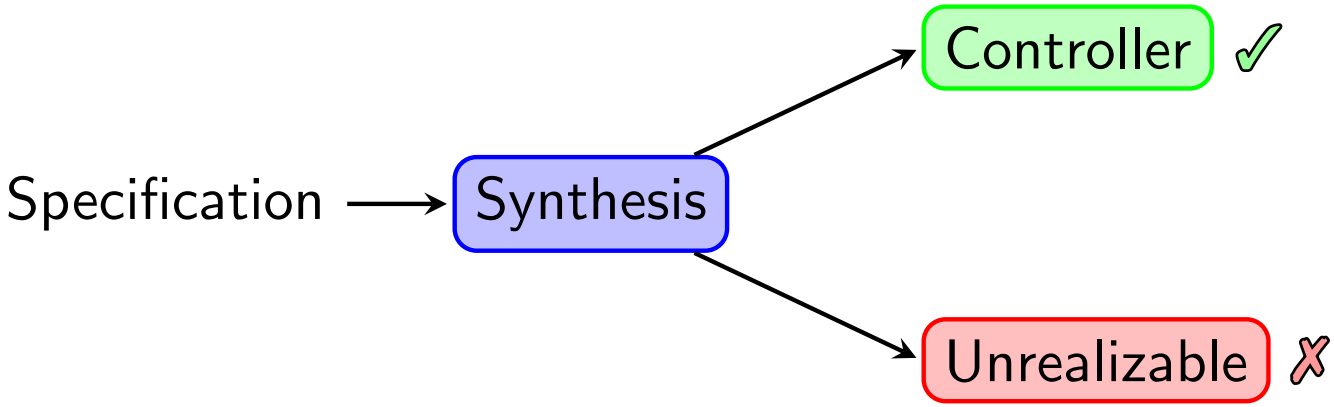
Controller Synthesis



Controller Synthesis



Controller Synthesis

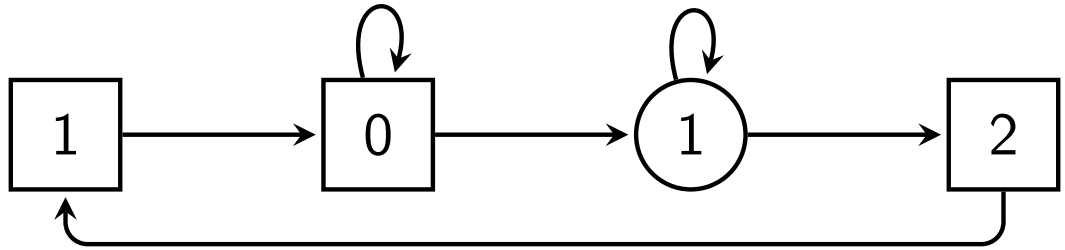


In most cases boils down to solving parity game

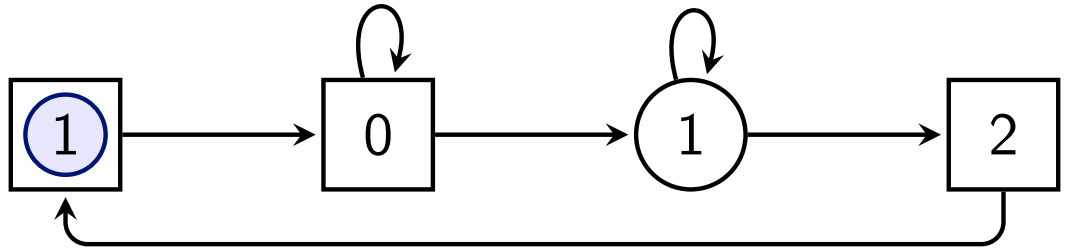
Parity Games

Parity Games

Parity Games

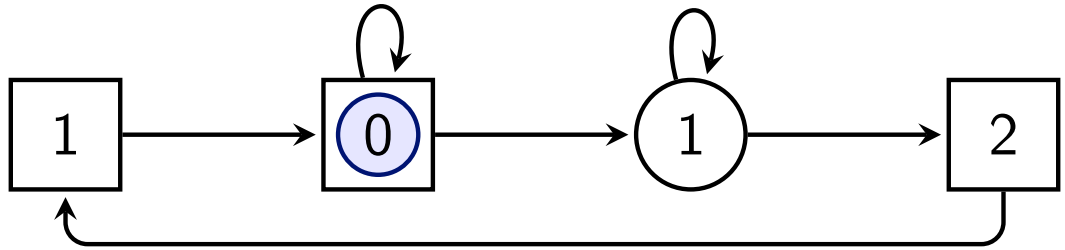


Parity Games



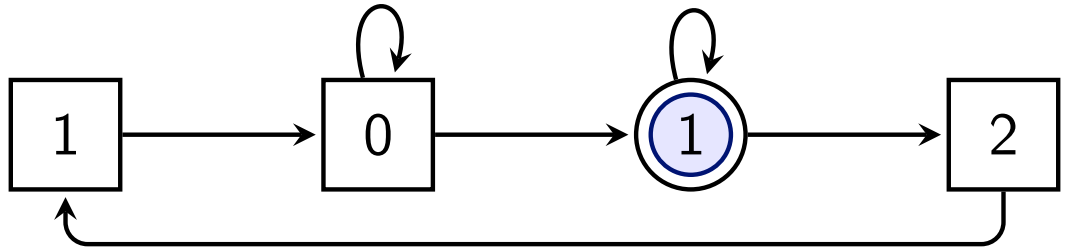
1

Parity Games



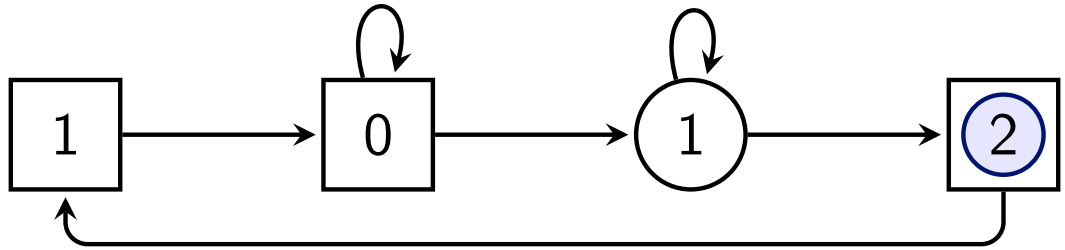
$1 \longrightarrow 0$

Parity Games



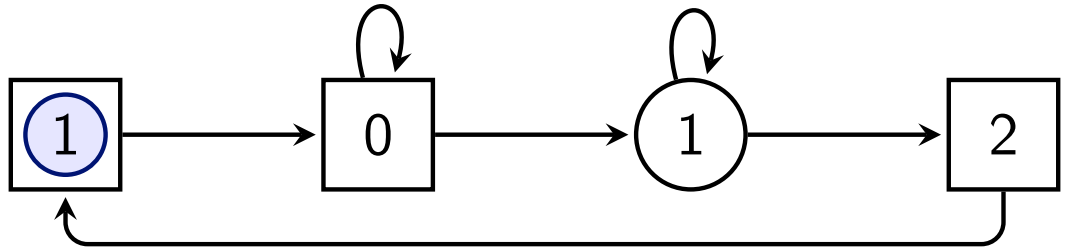
$1 \longrightarrow 0 \longrightarrow 1$

Parity Games



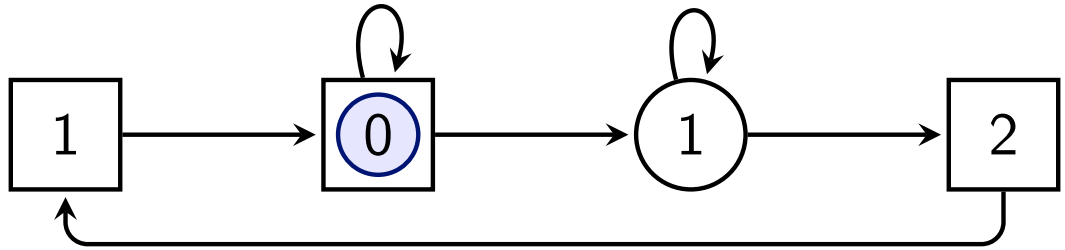
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2$

Parity Games



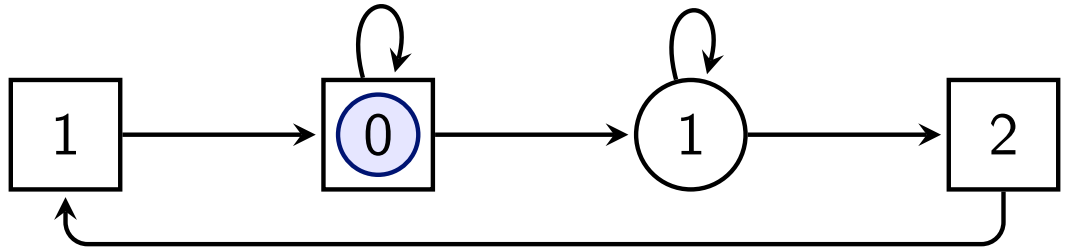
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1$

Parity Games



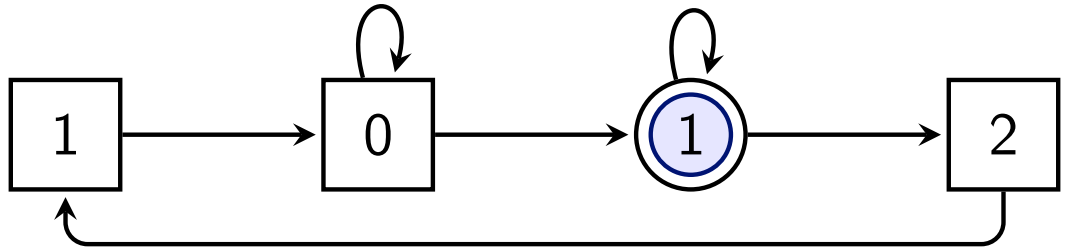
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0$

Parity Games



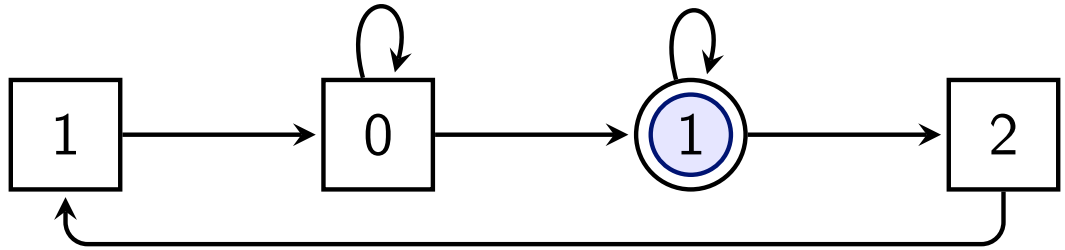
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0$

Parity Games



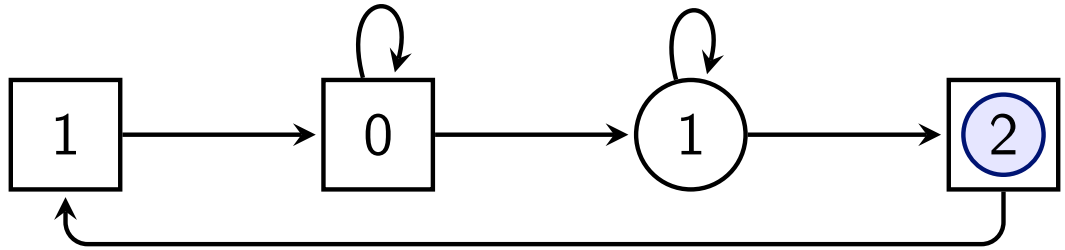
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1$

Parity Games



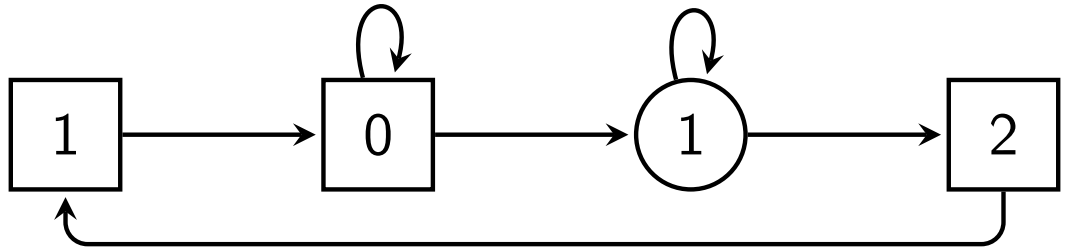
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1$

Parity Games



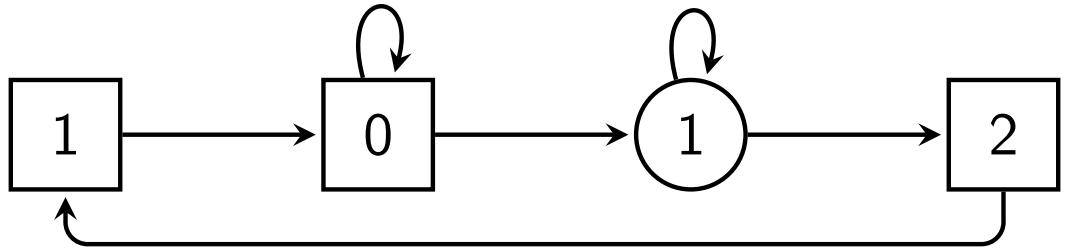
$$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

Parity Games



$$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

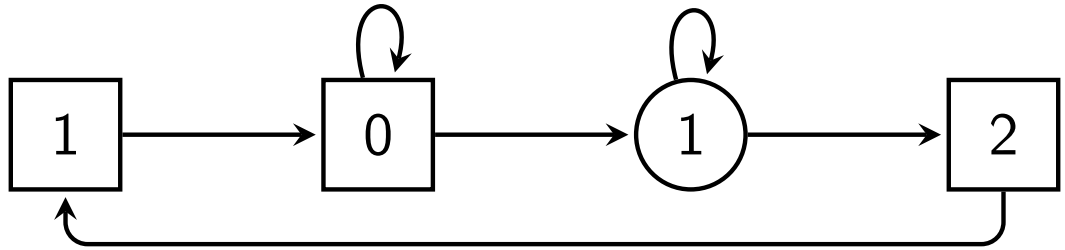
Parity Games



$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$

Max. color inftly often even

Parity Games



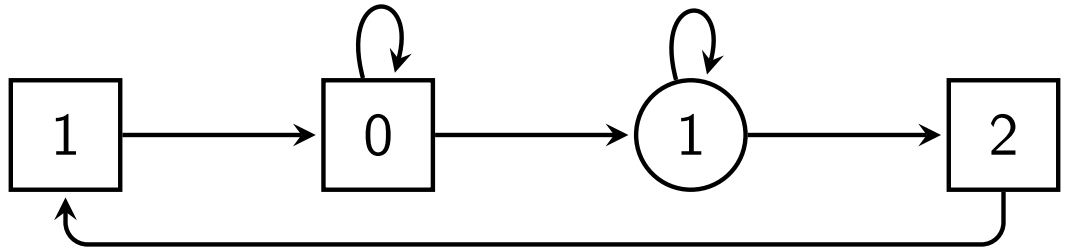
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \rightarrow \cdots$

Max. color inftly often even

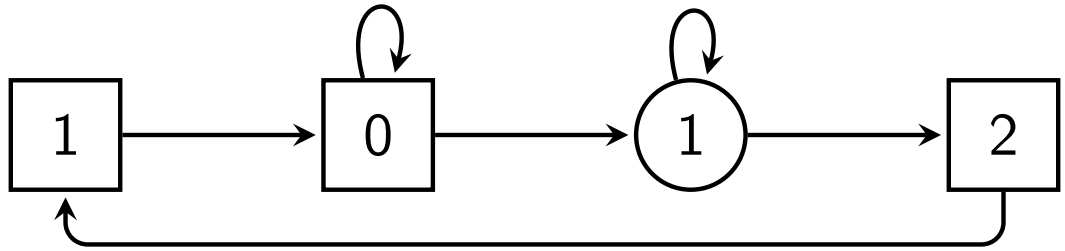
Proposition (Jurdziński 1998, Calude et al. 2017)

Solving parity games is in $\mathrm{UP} \cap \mathrm{co\text{-}UP}$ *and they can be solved in quasi-polynomial time. Both players have memoryless winning strategies.*

Finitary Parity Games

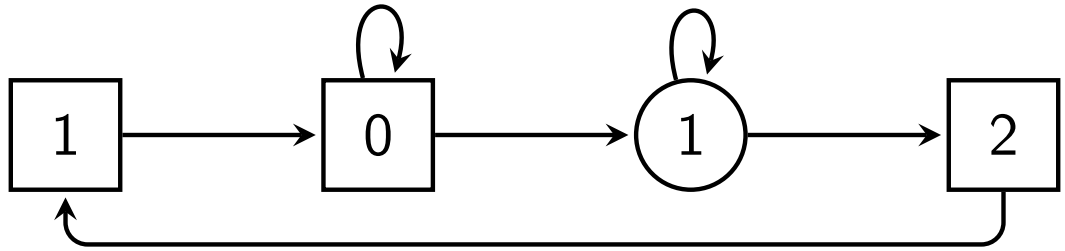


Finitary Parity Games



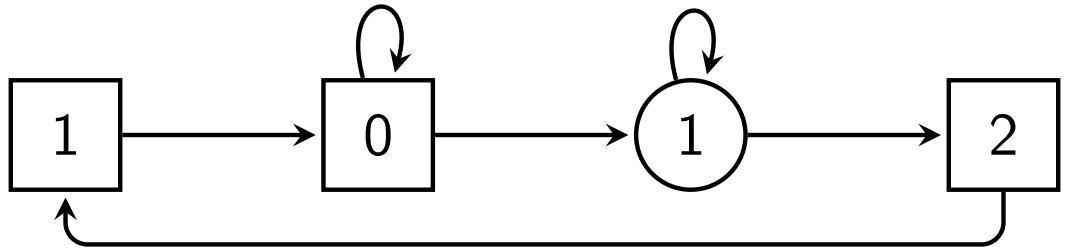
$$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

Finitary Parity Games



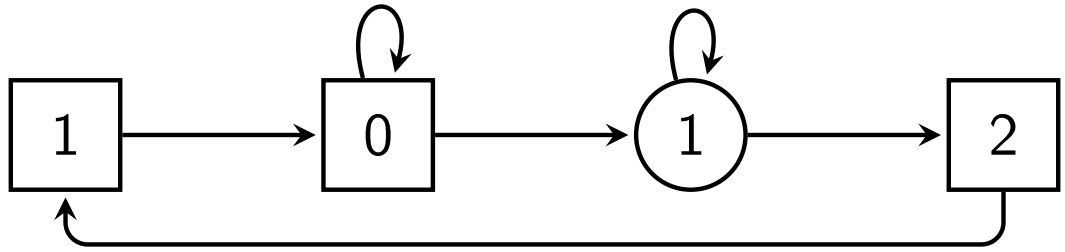
$$\underbrace{1 \to 0 \to 1 \to 2}_{3 \text{ steps}} \to 1 \to 0 \to 0 \to 1 \to 1 \to 2 \to \cdots$$

Finitary Parity Games



$$\underbrace{1 \to 0 \to 1 \to 2}_{3 \text{ steps}} \to \underbrace{1 \to 0 \to 0 \to 1 \to 1 \to 2}_{5 \text{ steps}} \to \cdots$$

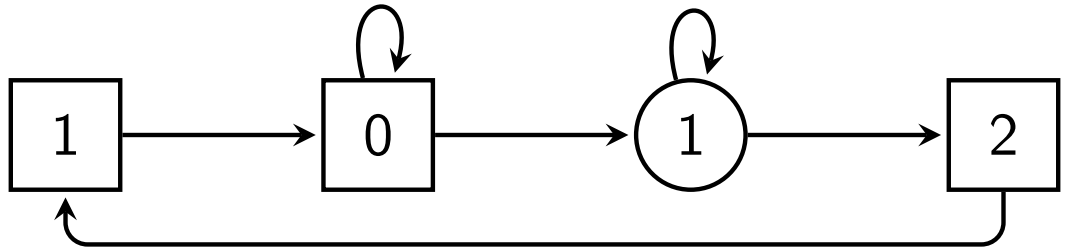
Finitary Parity Games



$$\underbrace{1 \to 0 \to 1 \to 2}_{3 \text{ steps}} \to \underbrace{1 \to 0 \to 0 \to 1 \to 1 \to 2}_{5 \text{ steps}} \to \cdots$$

Aim for Player 0:
Eventually bound steps between requests and answers

Finitary Parity Games



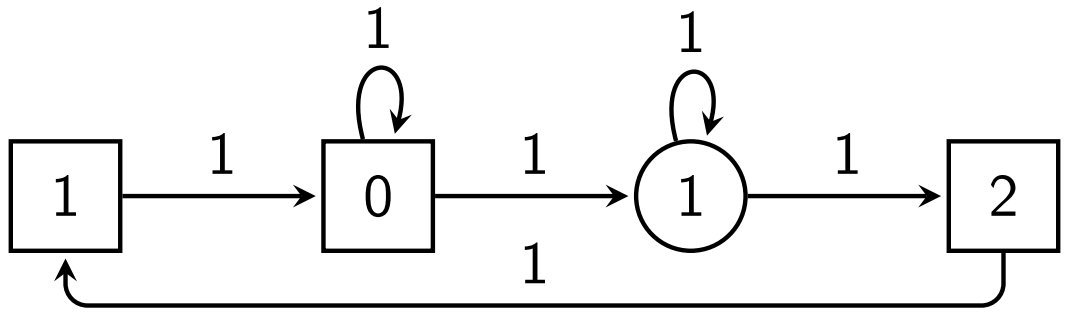
$$\underbrace{1 \to 0 \to 1 \to 2}_{3 \text{ steps}} \to \underbrace{1 \to 0 \to 0 \to 1 \to 1 \to 2}_{5 \text{ steps}} \to \cdots$$

Aim for Player 0:
Eventually bound steps between requests and answers

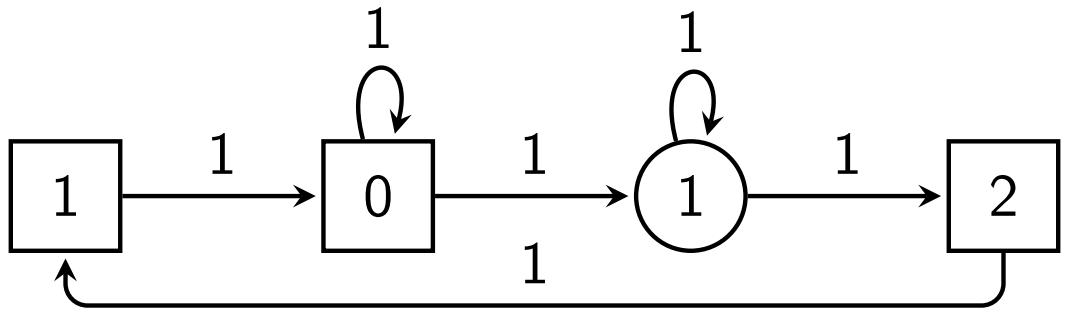
Proposition (Chatterjee, Henzinger, and Horn, 2009)

Solving finitary parity games is in PTIME. *Player* 0 *has memoryless winning strategies.*

Parity Games with Costs

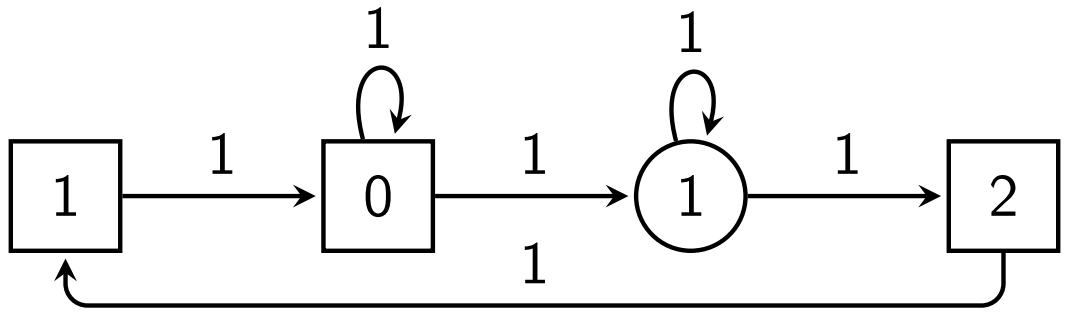


Parity Games with Costs



$$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

Parity Games with Costs

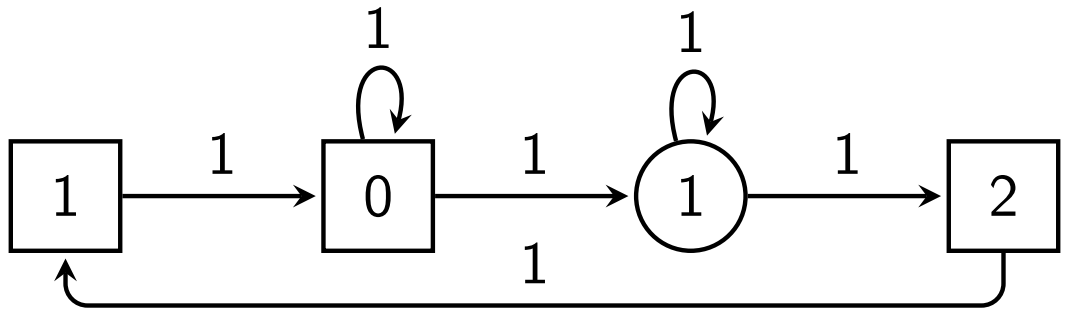


$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \rightarrow \cdots$

Aim for Player 0:

Eventually bound steps between request and answer

Parity Games with Costs



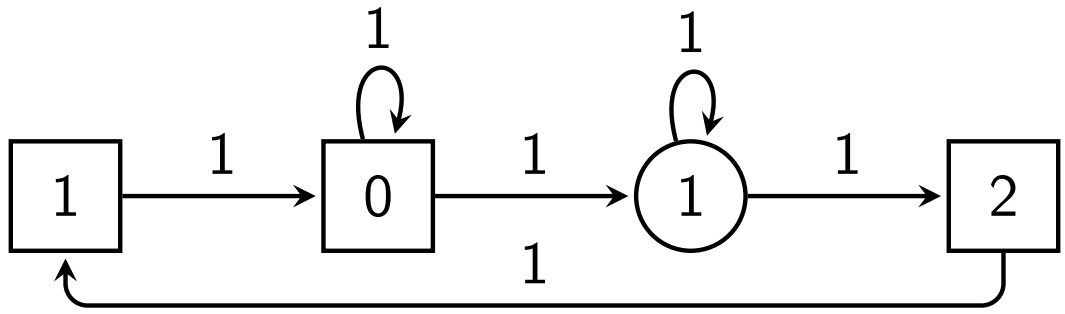
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$

Weight 3

Aim for Player 0:

Eventually bound steps between request and answer

Parity Games with Costs



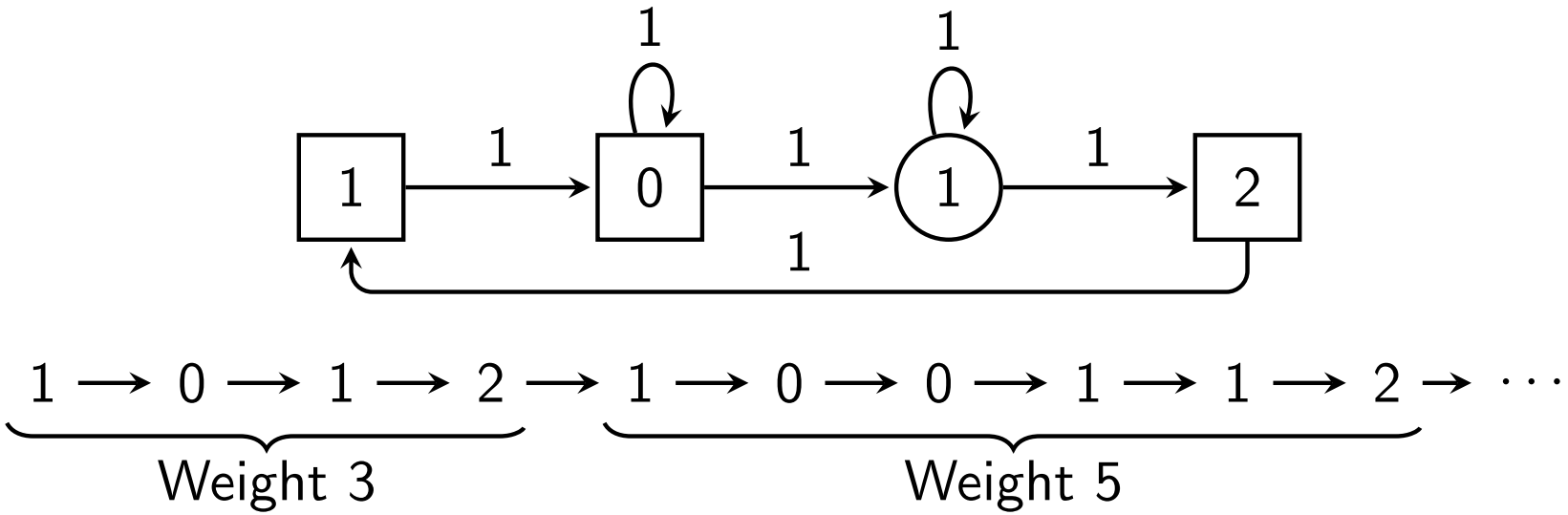
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$

Weight 3 Weight 5

Aim for Player 0:

Eventually bound steps between request and answer

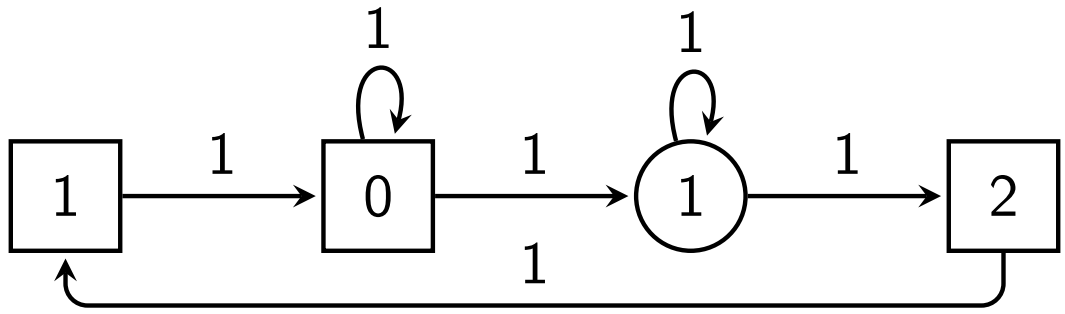
Parity Games with Costs



Aim for Player 0:

Eventually bound ~~steps~~ weight between request and answer

Parity Games with Costs



$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$

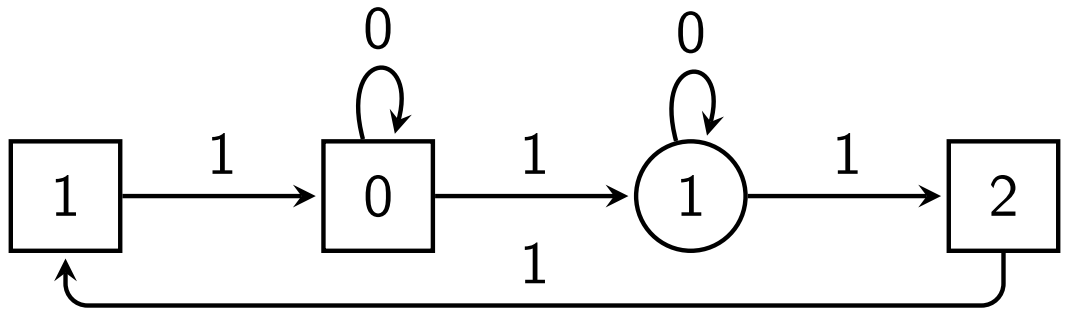
Weight 3 Weight 5

Aim for Player 0:
Eventually bound ~~steps~~ weight between request and answer

Proposition (Fijalkow and Zimmermann 2014 / Mogavero, Murano, and Sorrentino 2015)

Solving parity games with costs is in $\mathrm{UP} \cap \text{co-}\mathrm{UP}$. *Player* 0 *has memoryless winning strategies.*

Parity Games with Costs



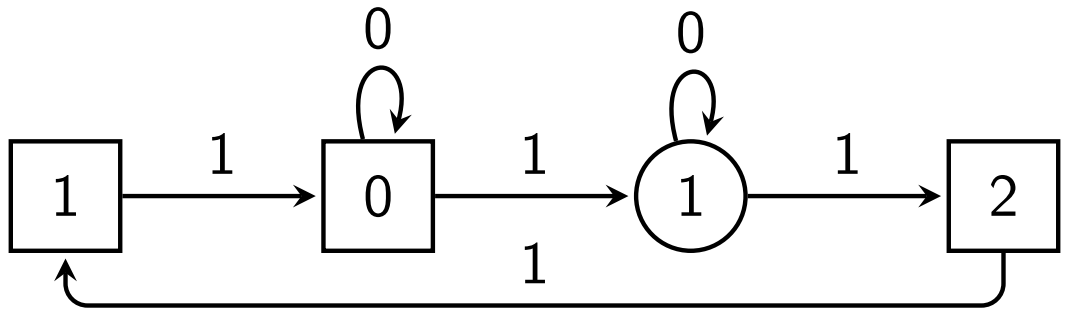
$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \rightarrow \cdots$

Aim for Player 0:
Eventually bound ~~steps~~ weight between request and answer

Proposition (Fijalkow and Zimmermann 2014 / Mogavero, Murano, and Sorrentino 2015)

Solving parity games with costs is in $\mathrm{UP} \cap \text{co-}\mathrm{UP}$. *Player* 0 *has memoryless winning strategies.*

Parity Games with Costs



$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$

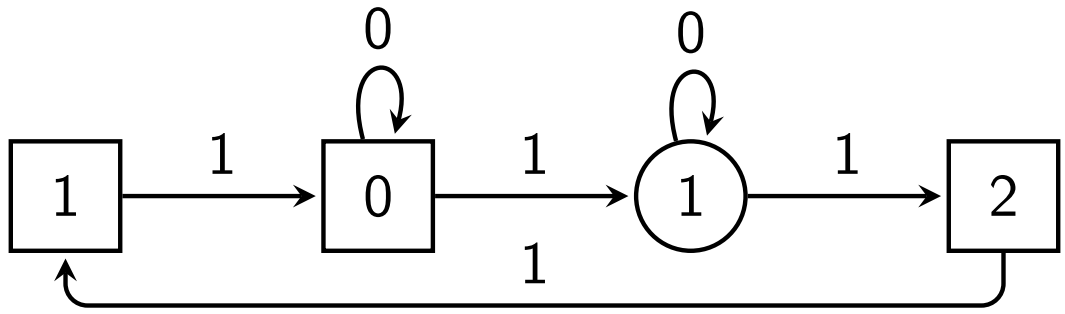
Weight 3

Aim for Player 0:
Eventually bound ~~steps~~ weight between request and answer

Proposition (Fijalkow and Zimmermann 2014 / Mogavero, Murano, and Sorrentino 2015)

Solving parity games with costs is in $\mathrm{UP} \cap \text{co-}\mathrm{UP}$. *Player* 0 *has memoryless winning strategies.*

Parity Games with Costs

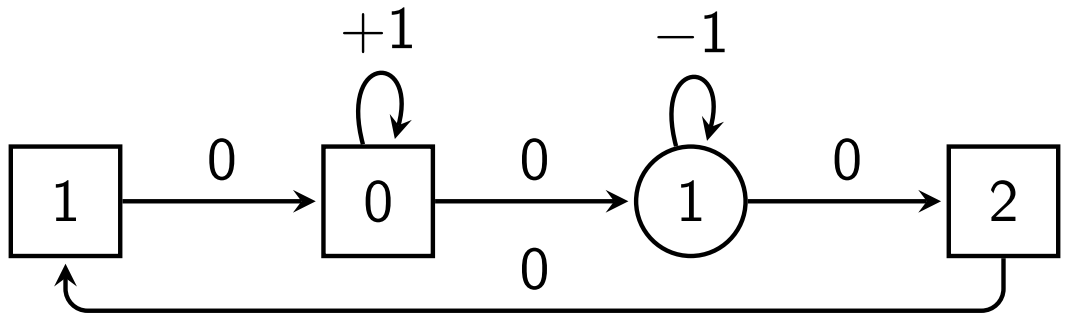


Aim for Player 0:
Eventually bound ~~steps~~ weight between request and answer

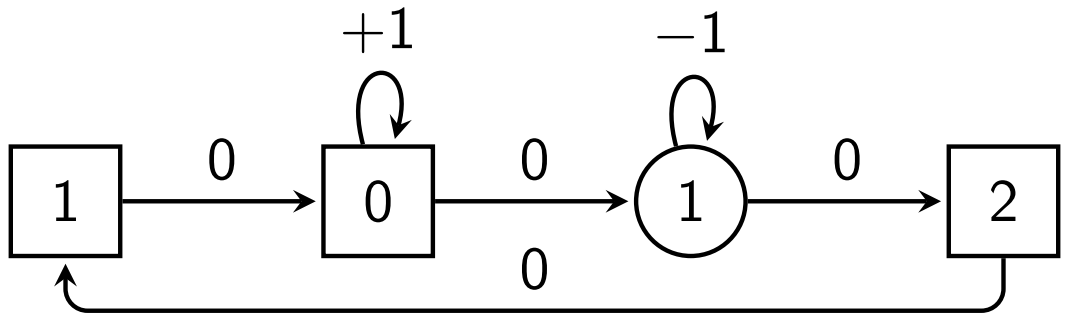
Proposition (Fijalkow and Zimmermann 2014 / Mogavero, Murano, and Sorrentino 2015)

Solving parity games with costs is in $\mathrm{UP} \cap \mathrm{co\text{-}UP}$. *Player* 0 *has memoryless winning strategies.*

Parity Games with Weights

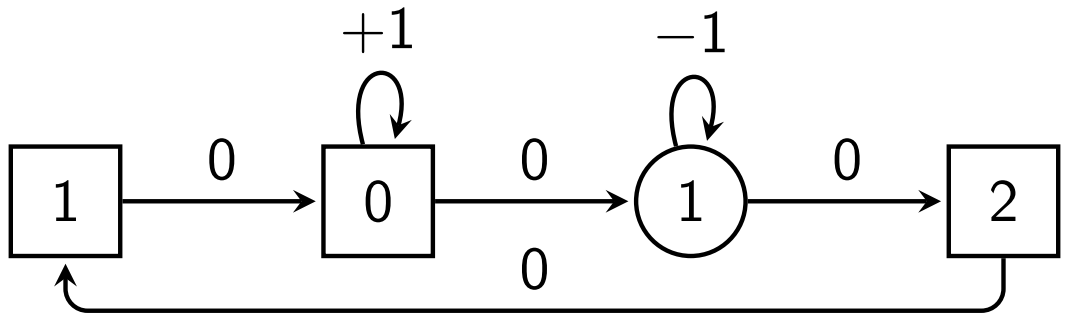


Parity Games with Weights



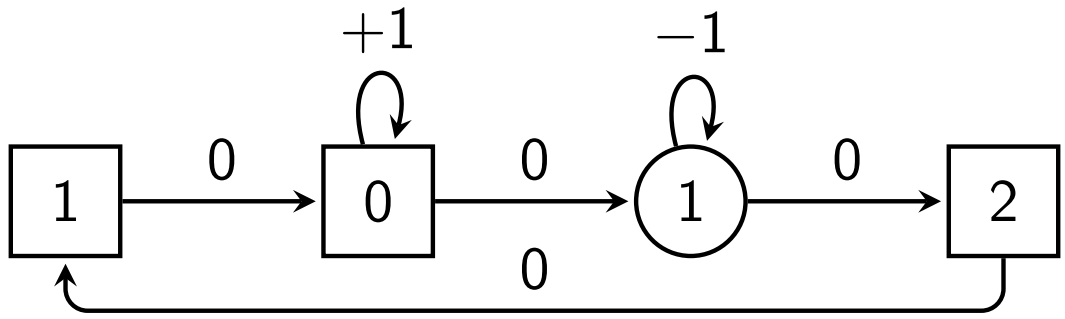
$$1 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 1 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

Parity Games with Weights



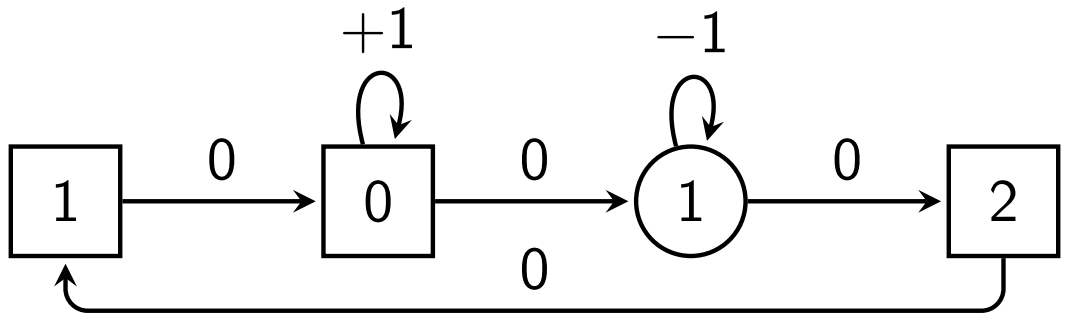
$$\underbrace{1 \to 0 \to 1 \to 2}_{\text{Weight } 0} \to 1 \to 0 \to 0 \to 1 \to 1 \to 2 \to \cdots$$

Parity Games with Weights



$$\underbrace{1 \to 0 \to 1 \to 2}_{\text{Weight } 0} \to \underbrace{1 \to 0 \to 0 \to 1 \to 1 \to 2}_{\text{Weight } 0} \to \cdots$$

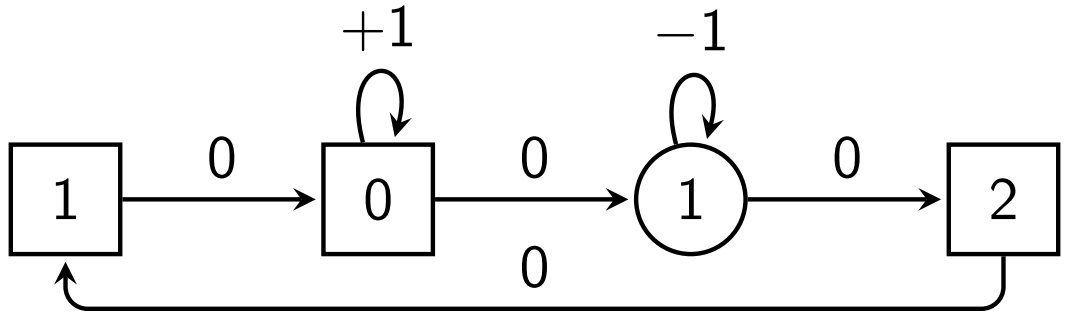
Parity Games with Weights



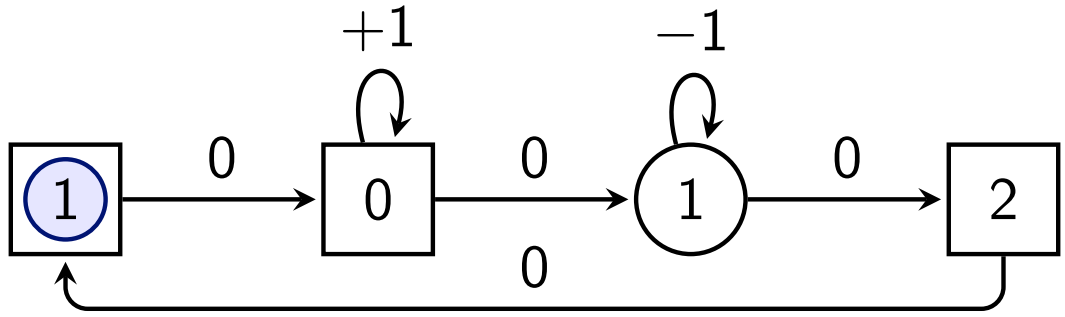
$$\underbrace{1 \to 0 \to 1 \to 2}_{\text{Weight } 0} \to \underbrace{1 \to 0 \to 0 \to 1 \to 1 \to 2}_{\text{Weight } 0} \to \cdots$$

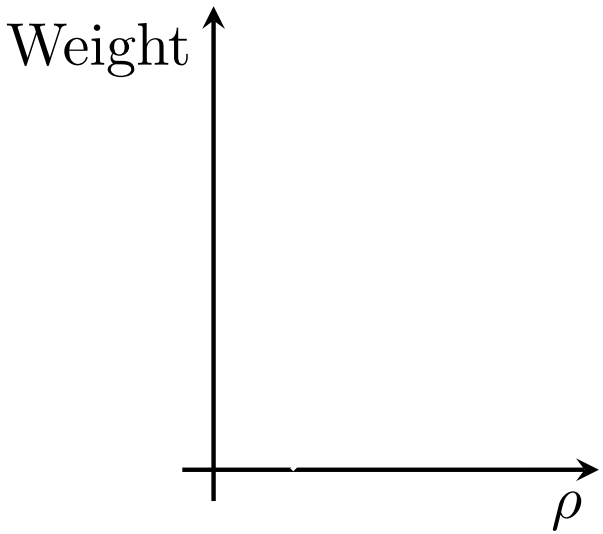
Not the complete picture...

Cost Measure

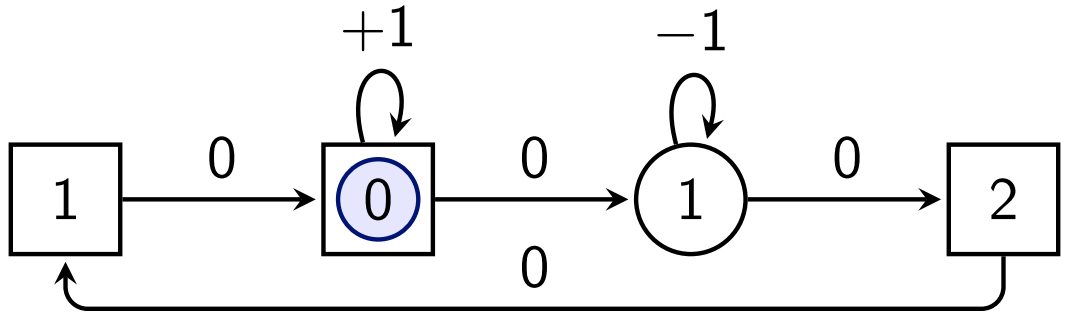


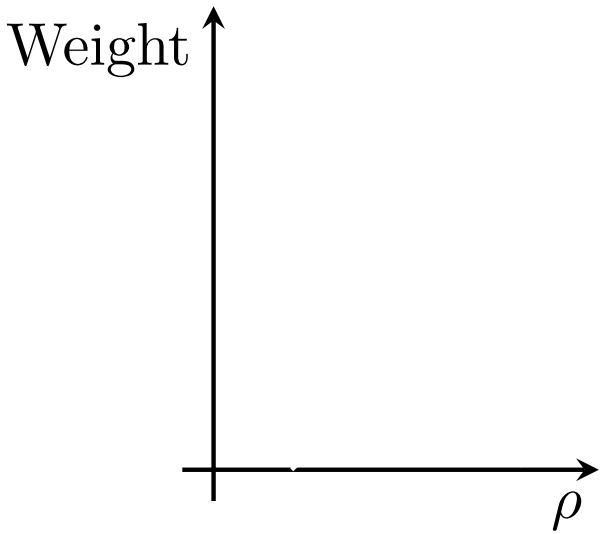
Cost Measure



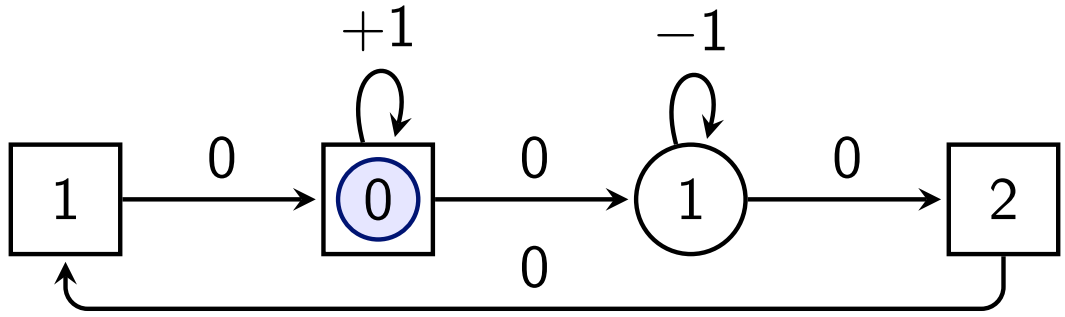


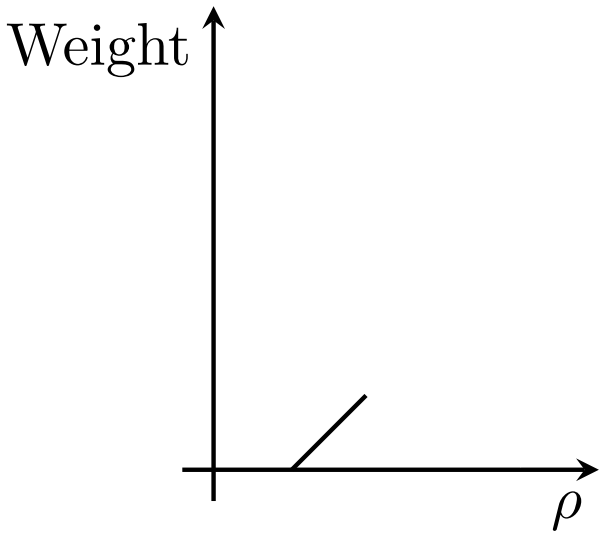
Cost Measure



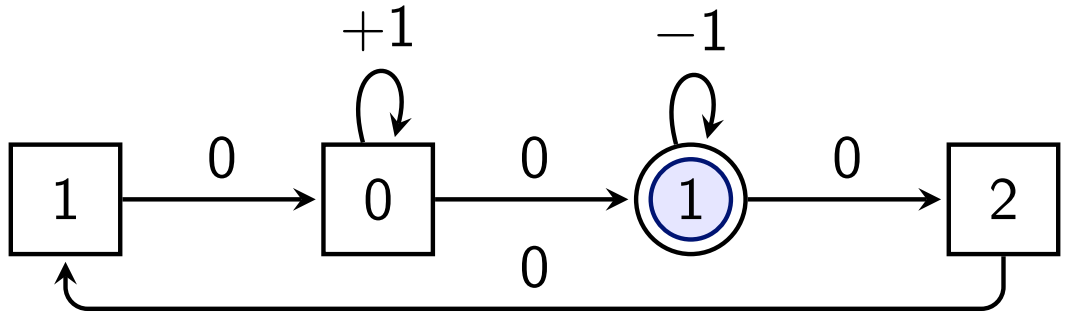


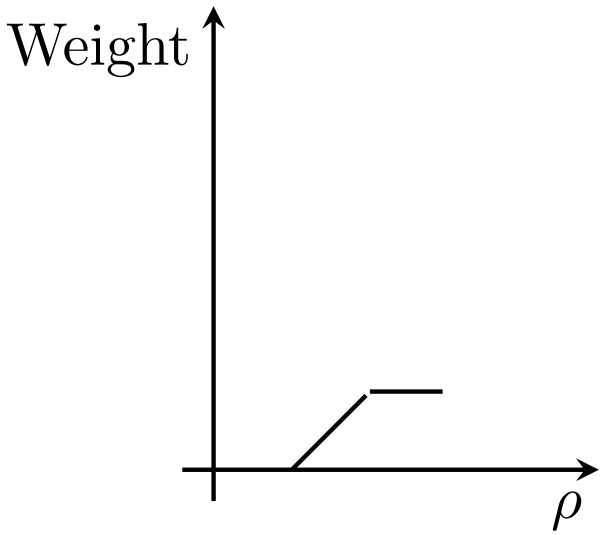
Cost Measure



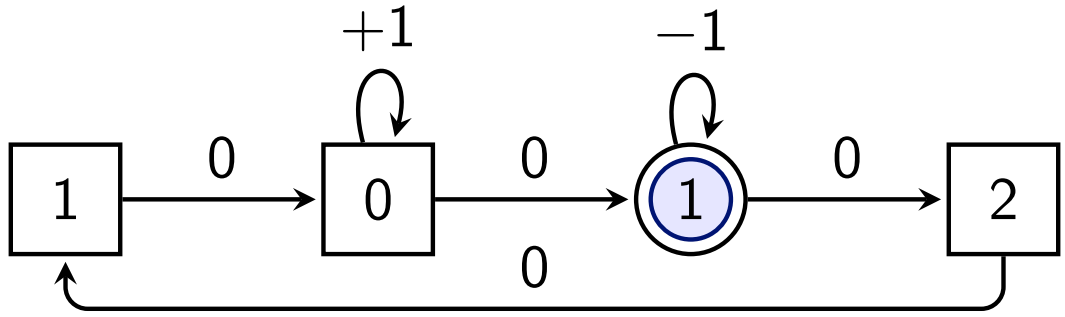


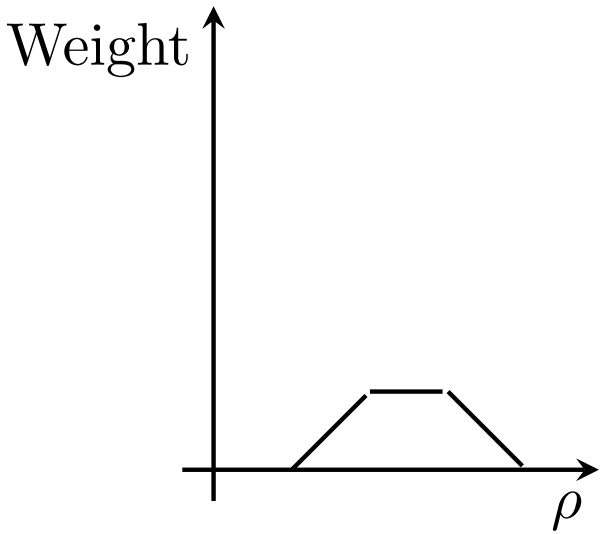
Cost Measure



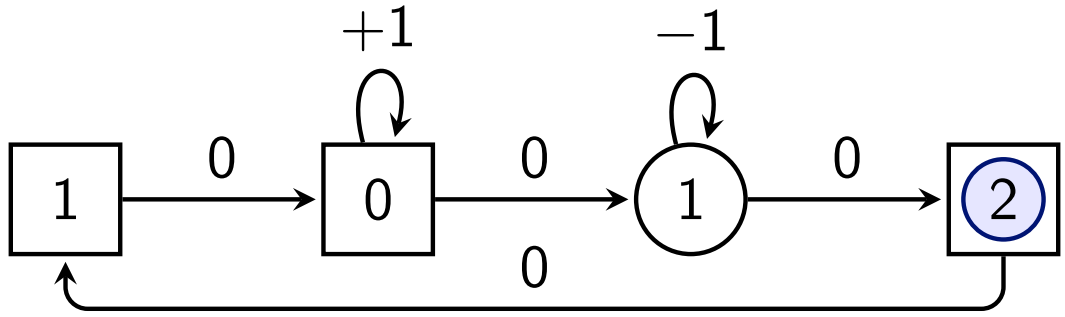


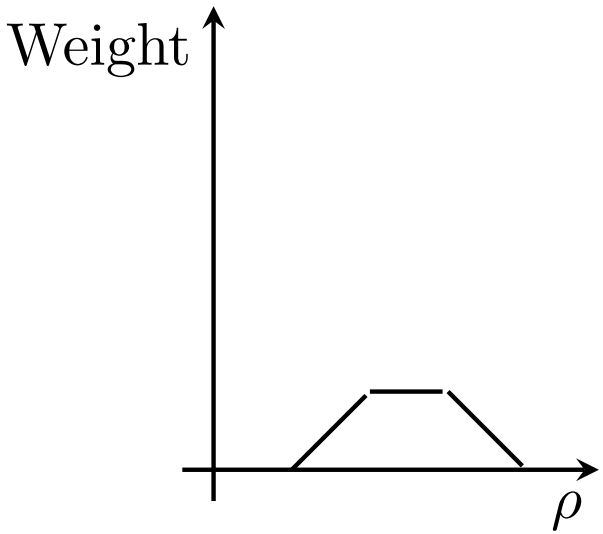
Cost Measure



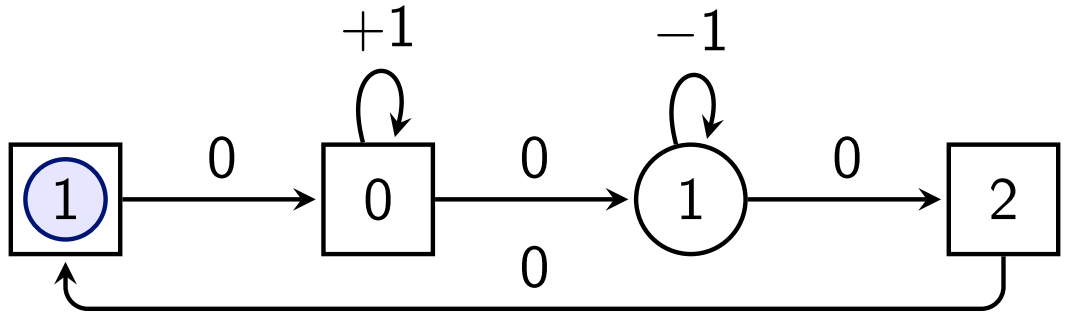


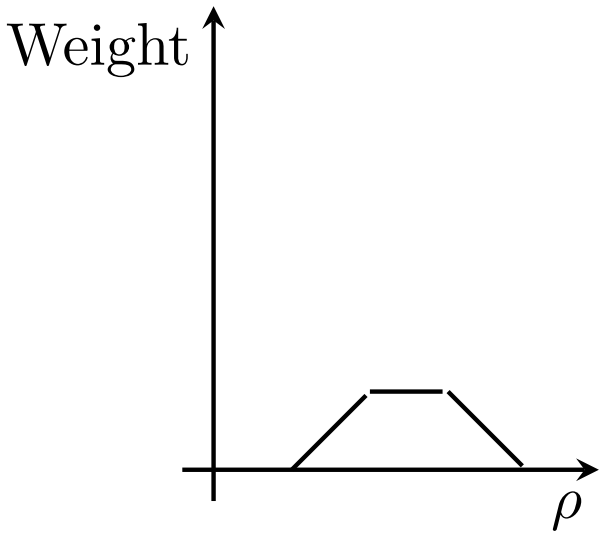
Cost Measure

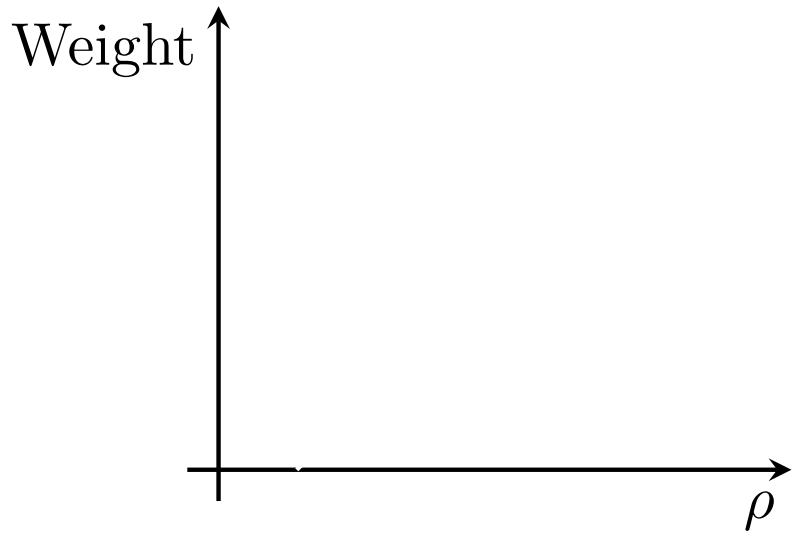




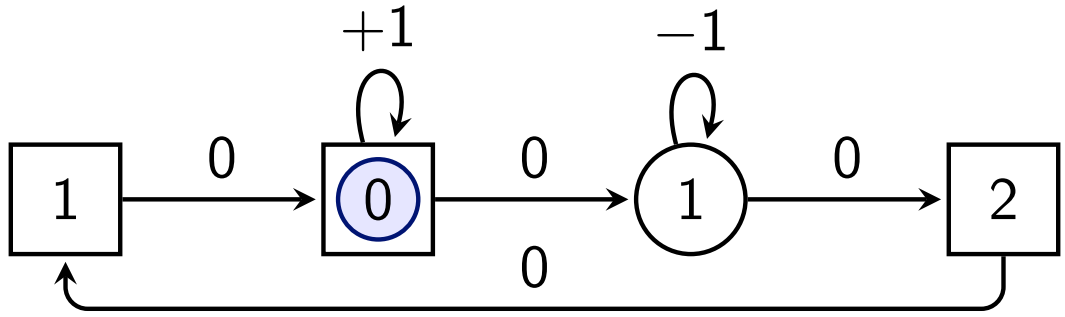
Cost Measure

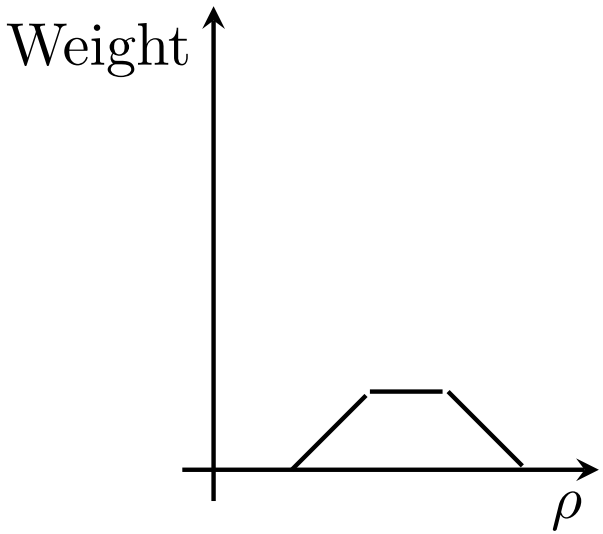


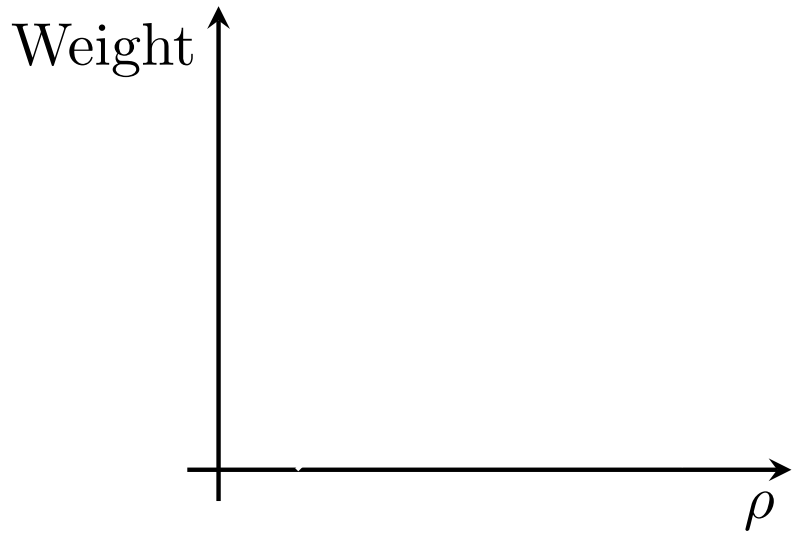




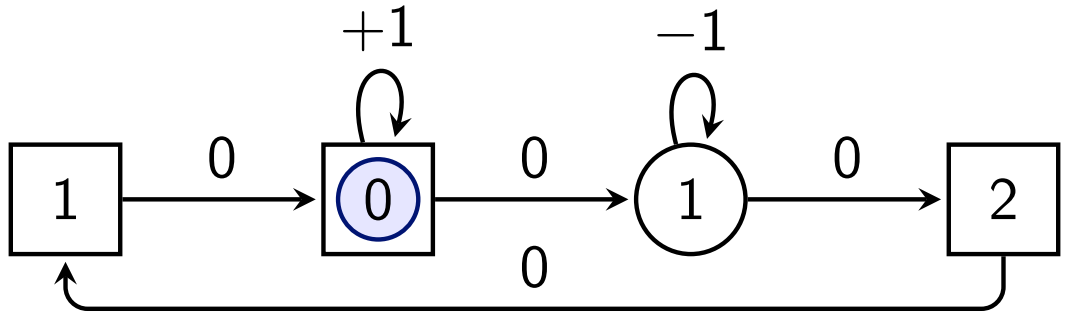
Cost Measure

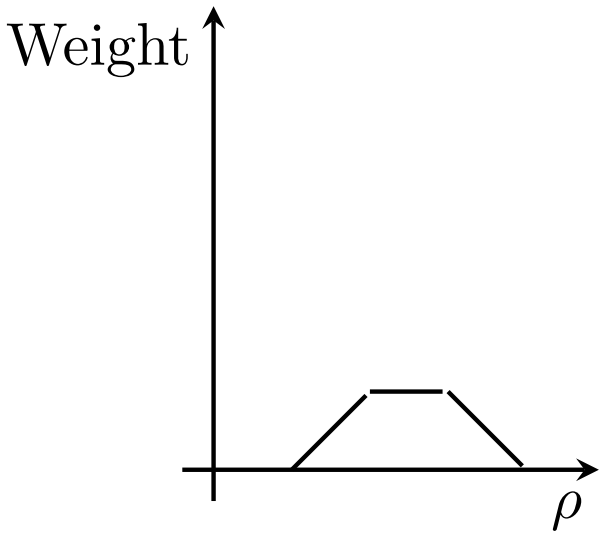


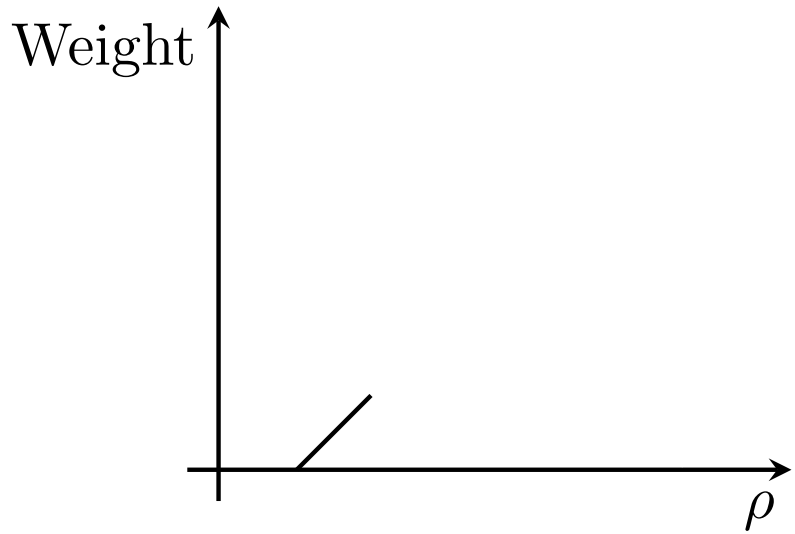




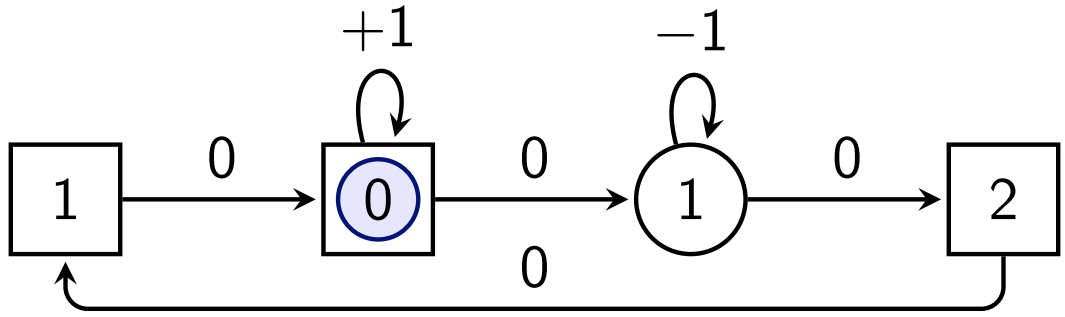
Cost Measure

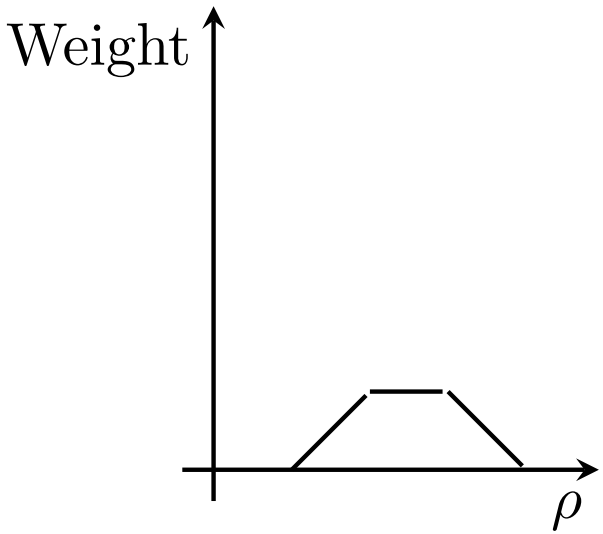


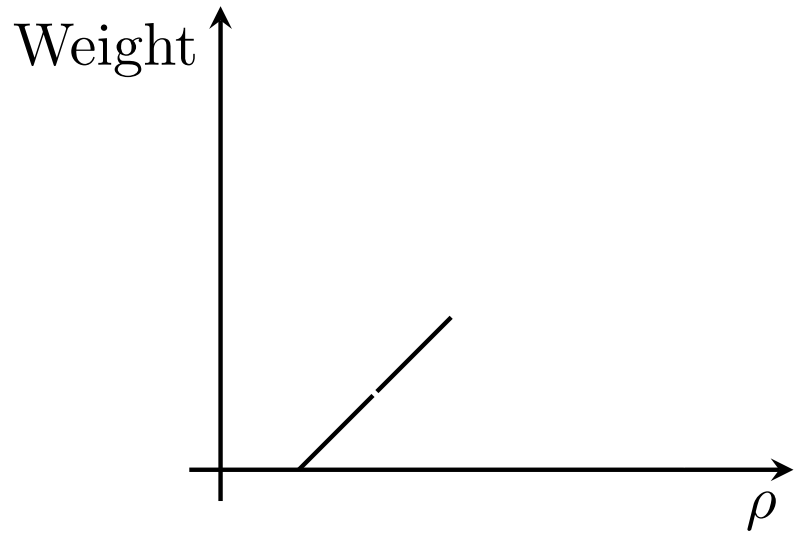




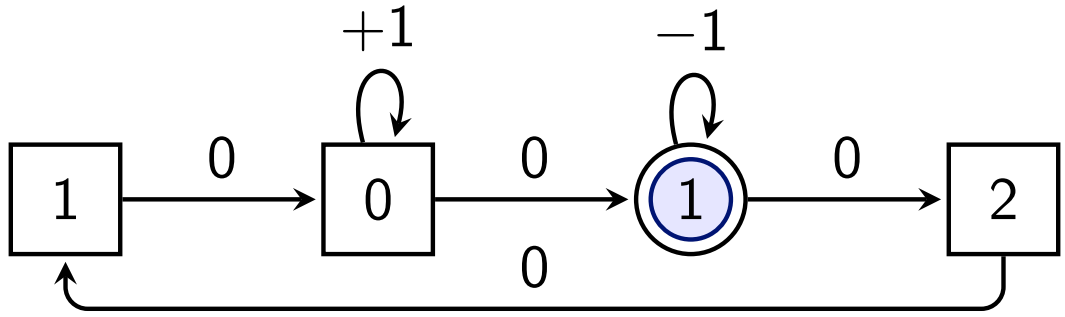
Cost Measure

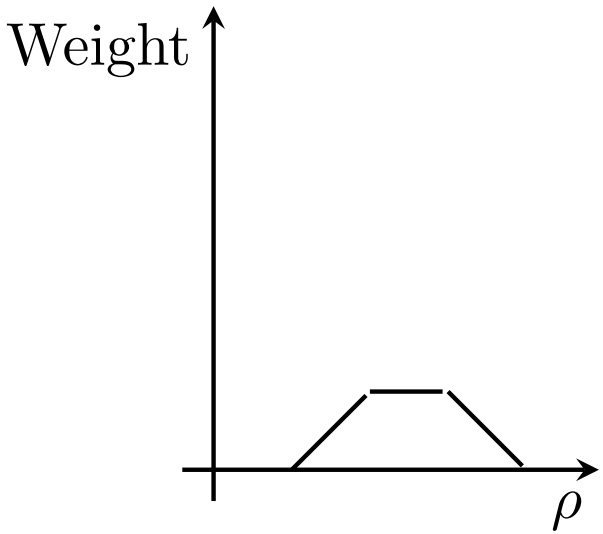


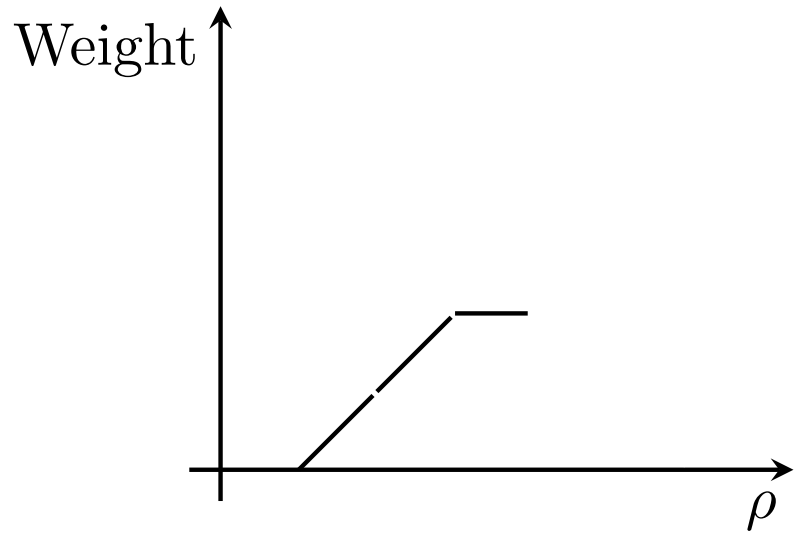




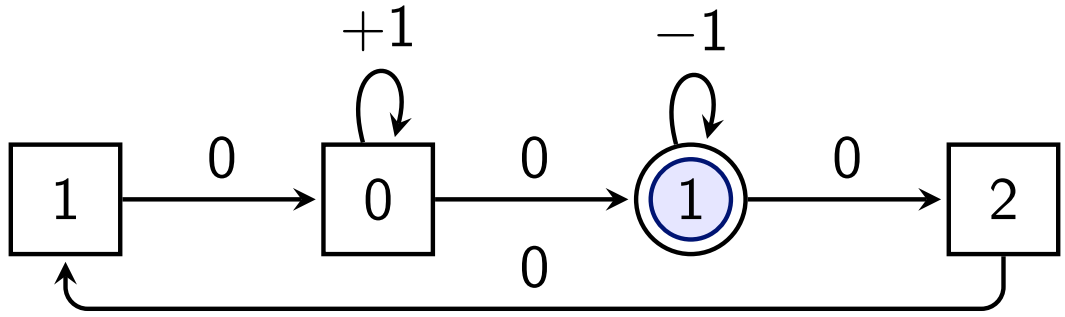
Cost Measure

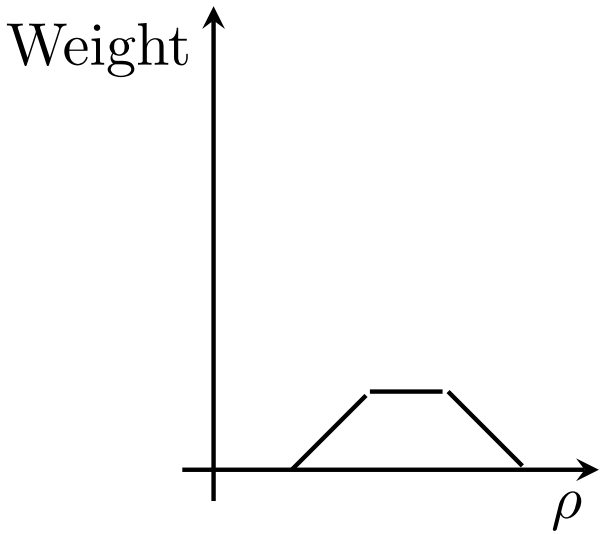


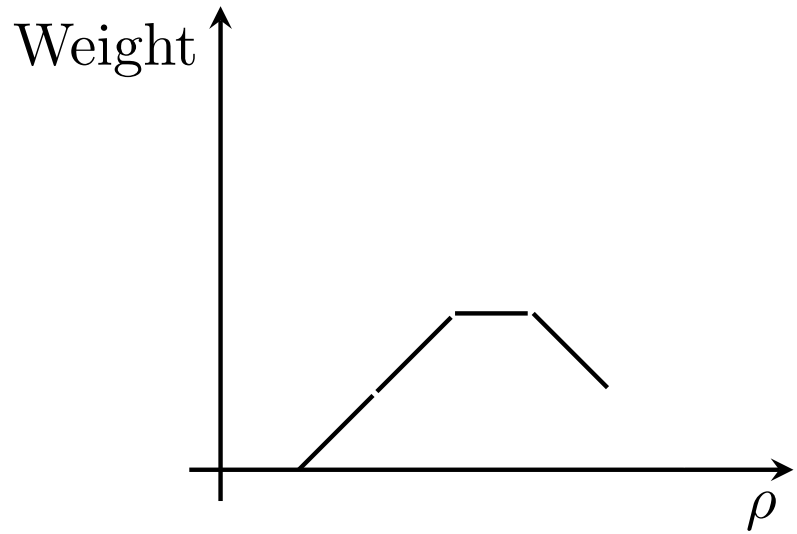




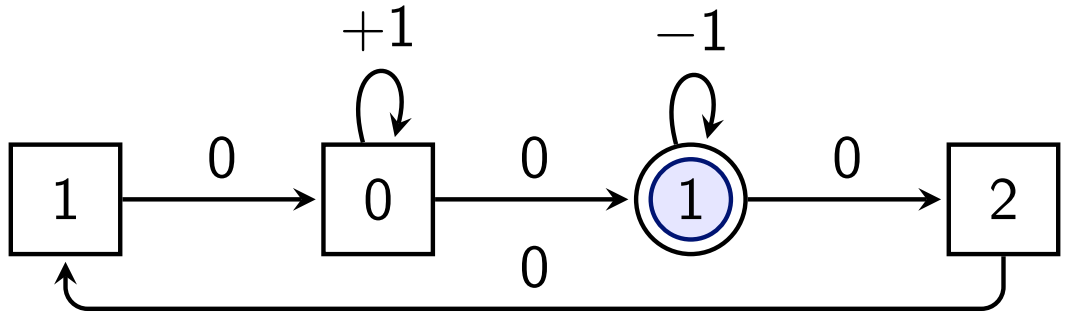
Cost Measure

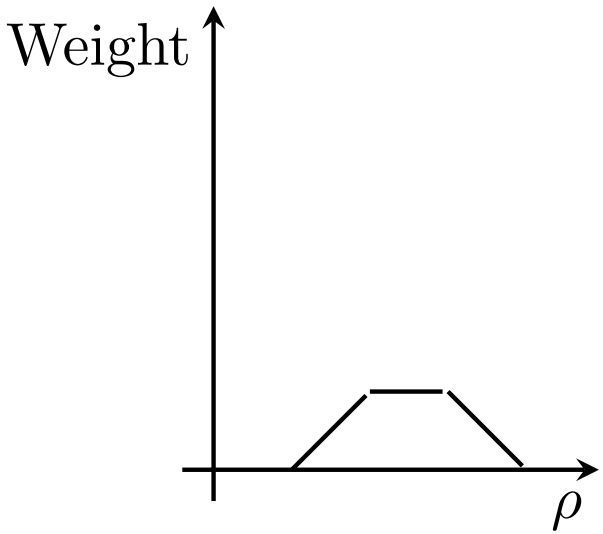


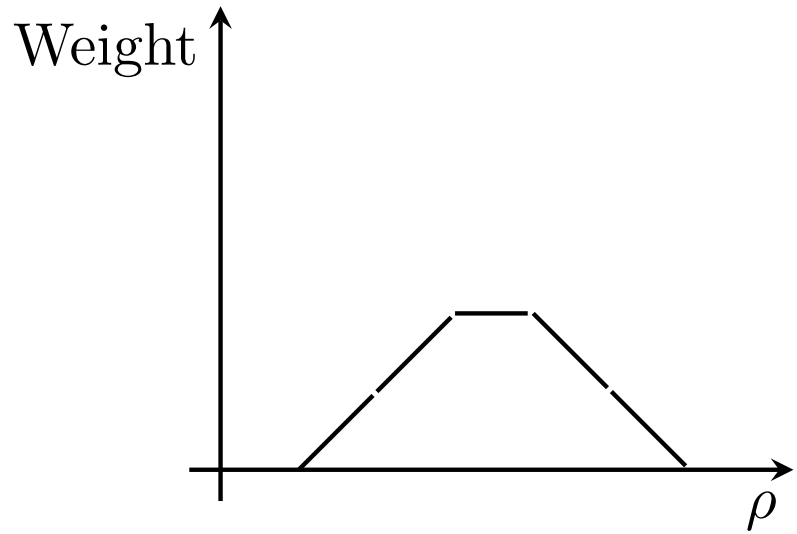




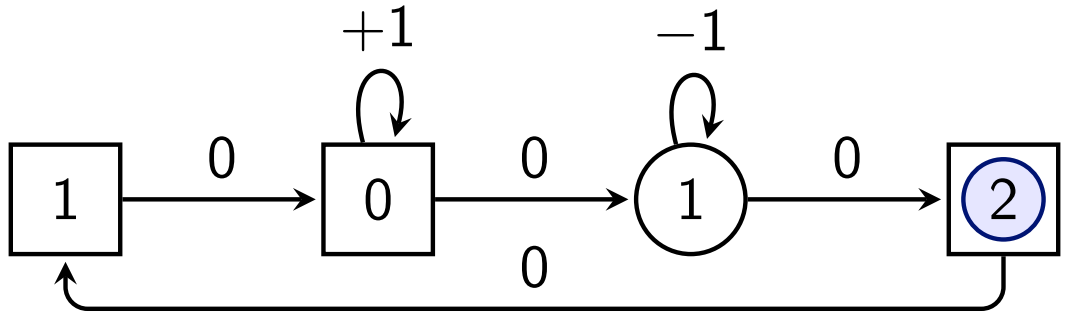
Cost Measure

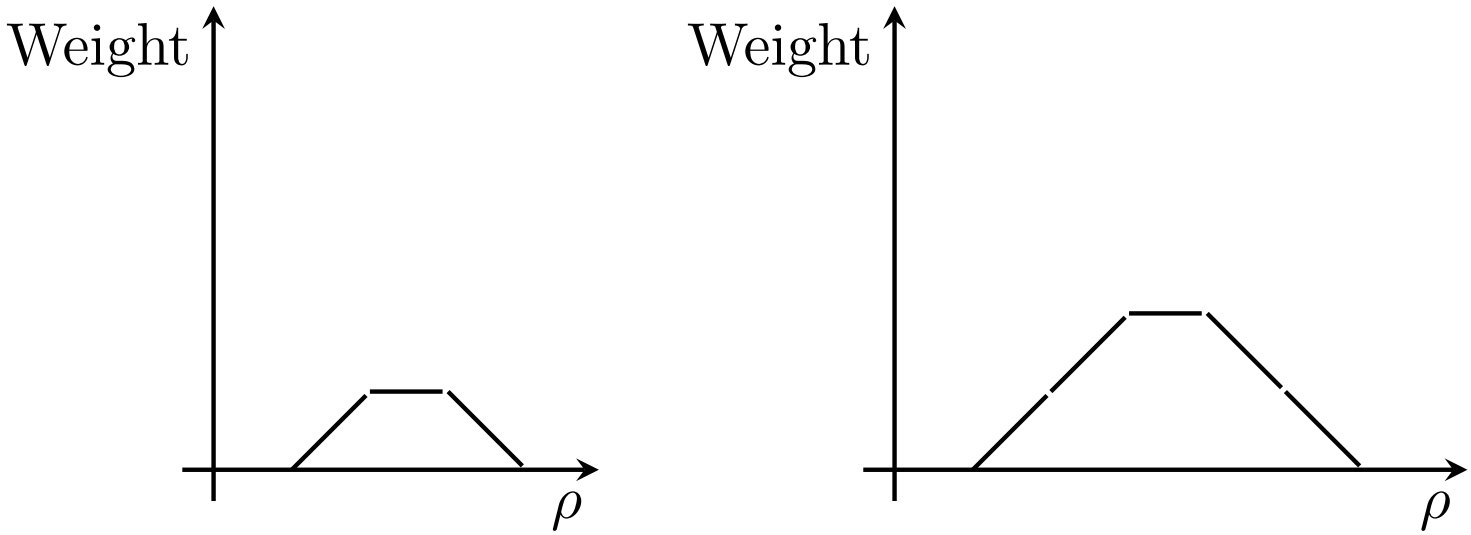






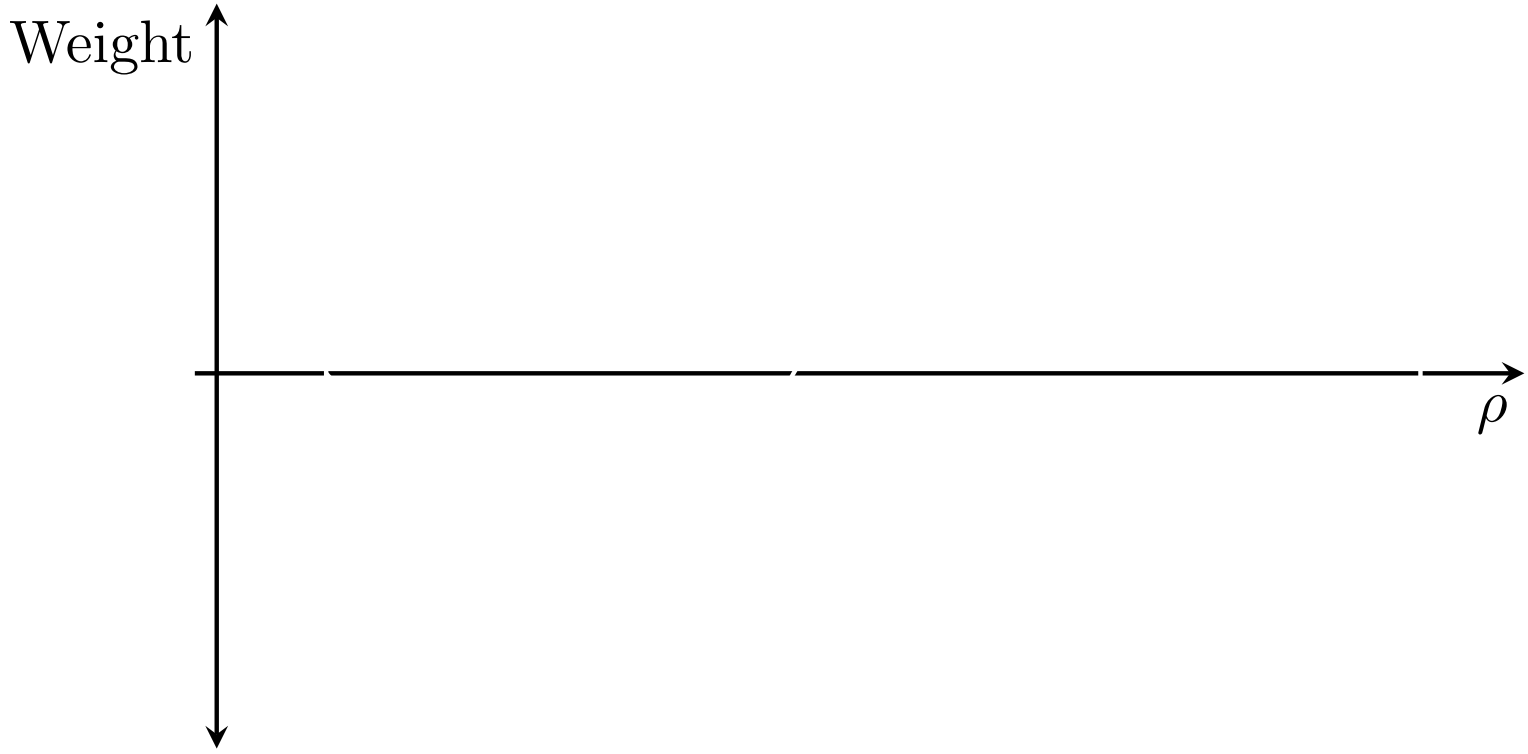
Cost Measure



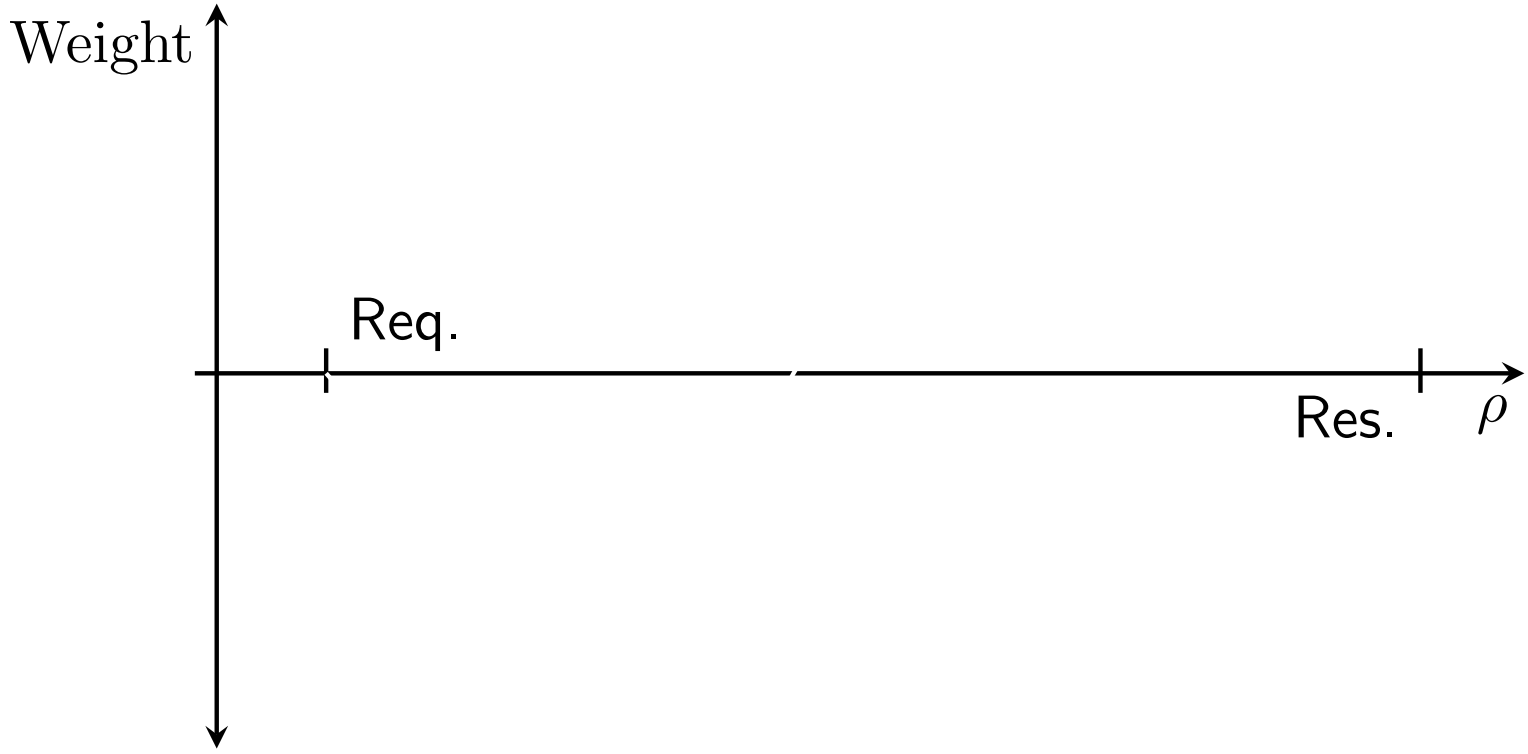


Intuitively:
Measure size of resource required to answer request

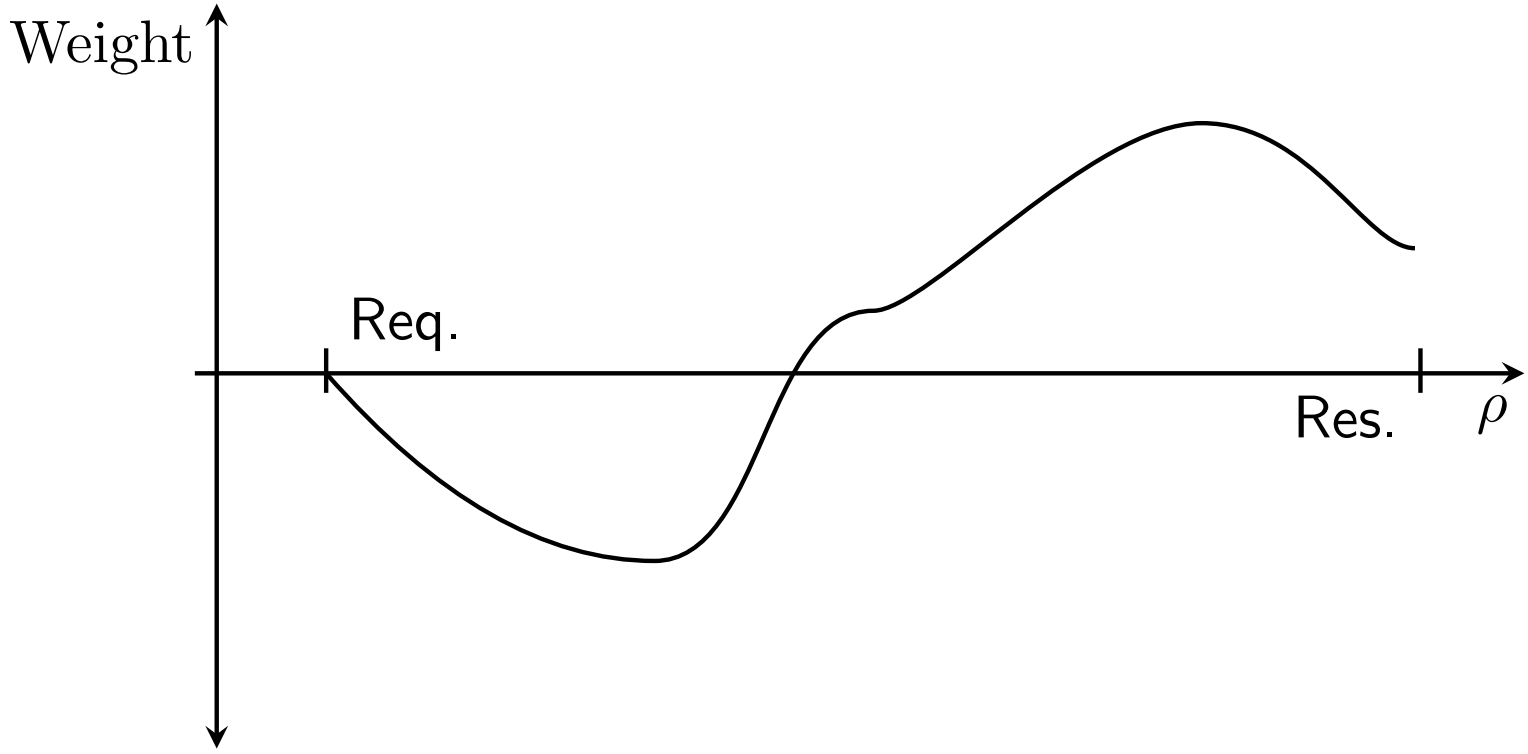
Cost Measure



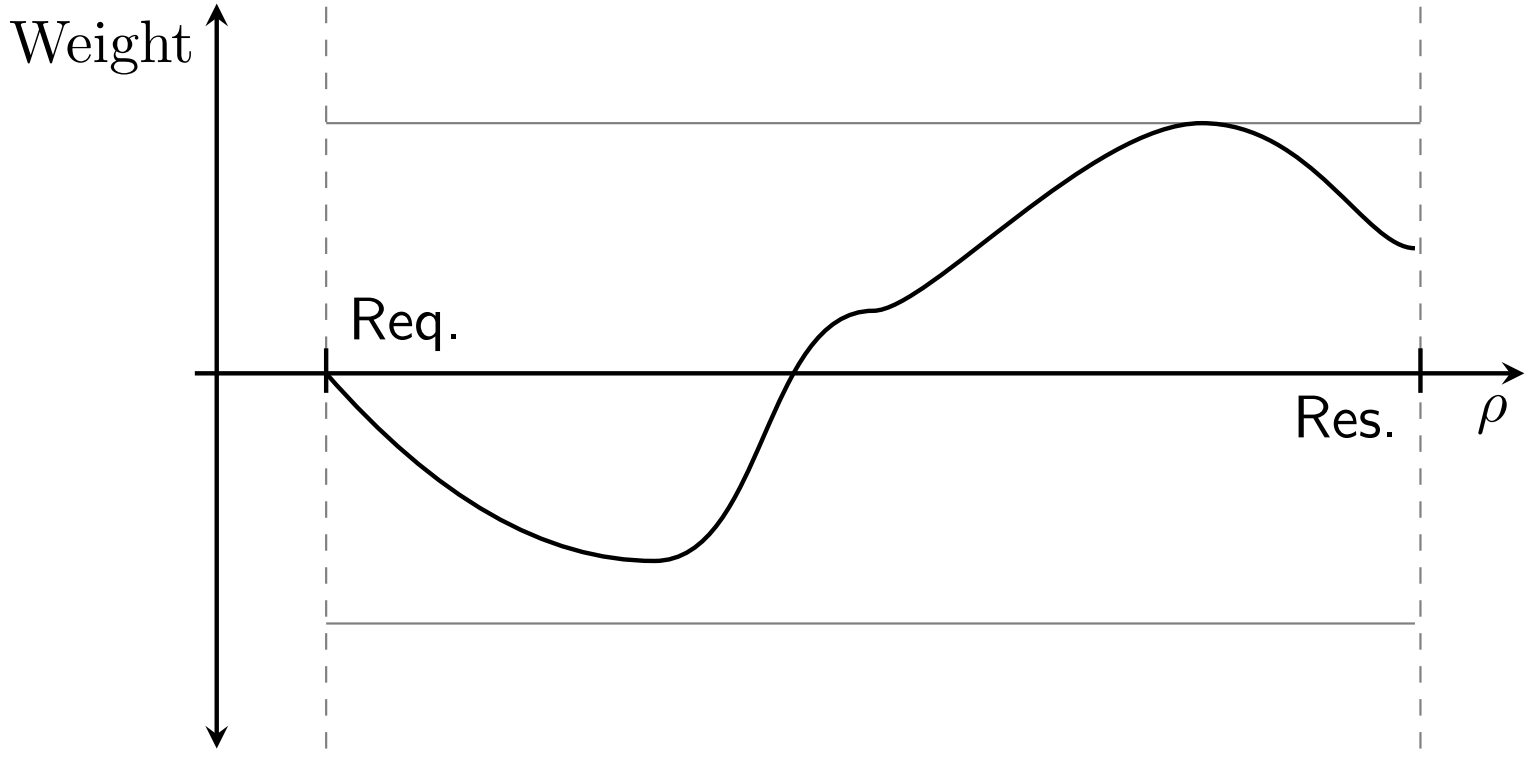
Cost Measure



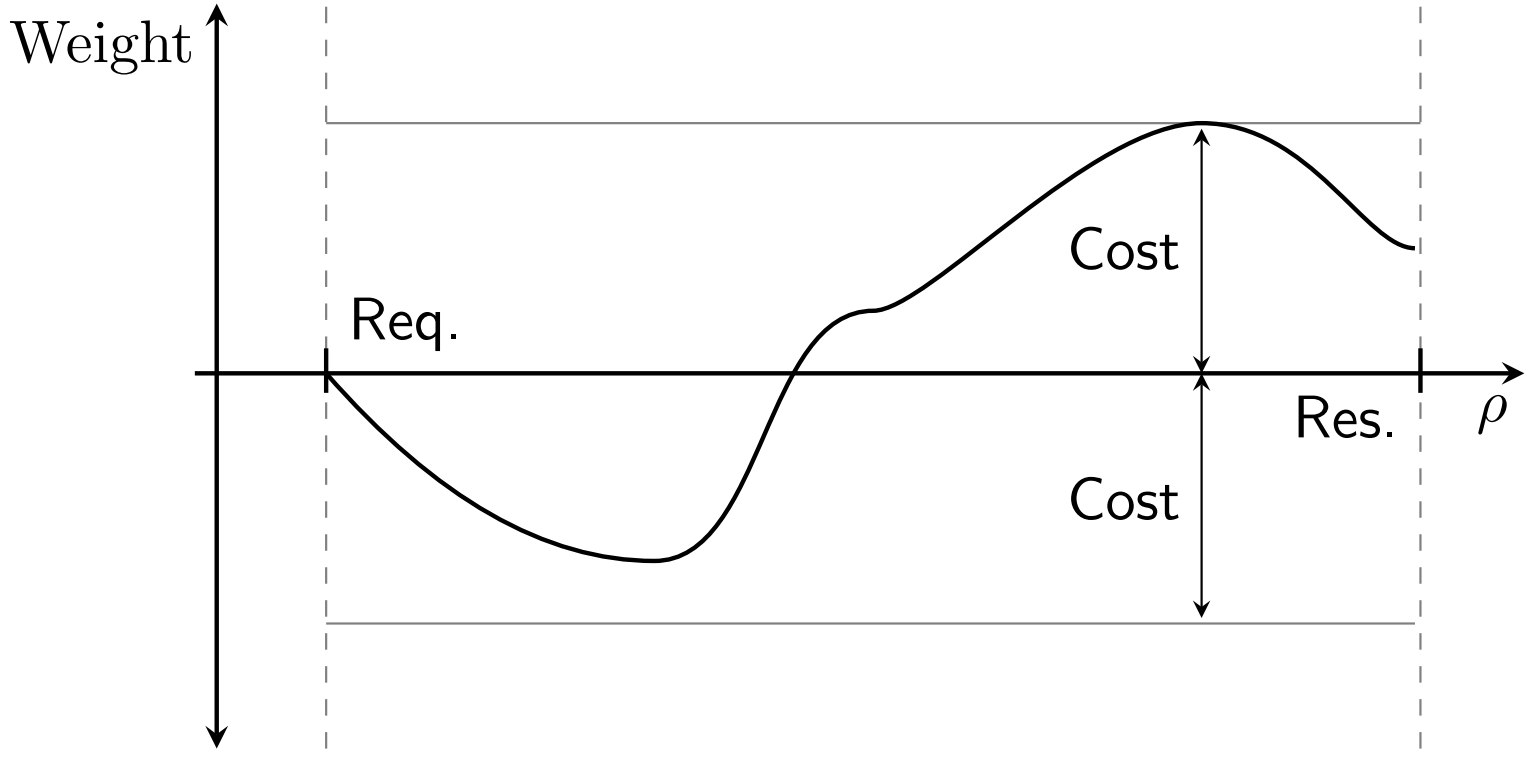
Cost Measure



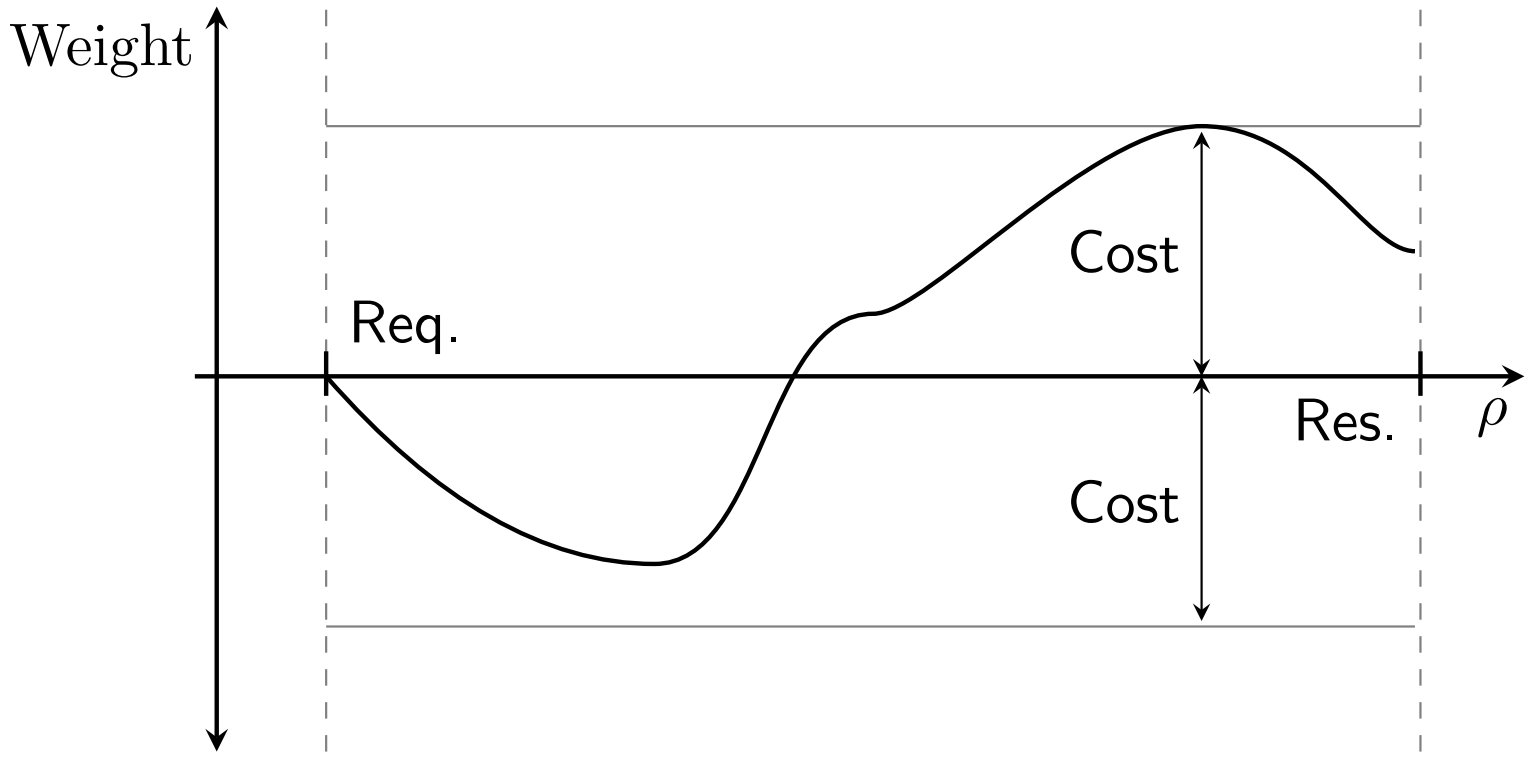
Cost Measure



Cost Measure



Cost Measure



Cost of answering: **Amplitude on path to answer**

Outline

- Definition
- Complexity
- Memory
- Bounds

Outline

- Definition ✓
- Complexity
- Memory
- Bounds

Solving Parity Games with Weights

Problem: Given a parity game with weights $\mathcal{G}$ and a vertex v of $\mathcal{G}$, does Player 0 win $\mathcal{G}$ from v?

Solving Parity Games with Weights

Problem: Given a parity game with weights $\mathcal{G}$ and a vertex v of $\mathcal{G}$, does Player 0 win $\mathcal{G}$ from v?

Problem: Given an arena $\mathcal{A}$ with colors and weights, can Player 0 bound the costs of answering odd colors with higher even ones?

Solving Parity Games with Weights

Problem: Given a parity game with weights $\mathcal{G}$ and a vertex v of $\mathcal{G}$, does Player 0 win $\mathcal{G}$ from v?

Problem: Given an arena $\mathcal{A}$ with colors and weights, can Player 0 bound the costs of answering odd colors with higher even ones?

Proof idea: Analogously to (Fijalkow and Zimmermann, 2014)

1. Strengthen parity condition with weights
2. Reduce to solving energy parity games

Bounding Parity Games with Weights

First Step: Strengthen parity condition with weights $\rightarrow$ bounded parity games with weights.

Bounding Parity Games with Weights

First Step: Strengthen parity condition with weights $\rightarrow$ bounded parity games with weights.

Parity with weights: Player 1 can win by causing infinitely many requests with infinite cost

Bounding Parity Games with Weights

First Step: Strengthen parity condition with weights $\rightarrow$ bounded parity games with weights.

Parity with weights: Player 1 can win by causing infinitely many requests with infinite cost

Bounded parity with weights: Player 1 can win by causing a single request with infinite cost

Bounding Parity Games with Weights

First Step: Strengthen parity condition with weights $\rightarrow$ bounded parity games with weights.

Parity with weights: Player 1 can win by causing infinitely many requests with infinite cost

Bounded parity with weights: Player 1 can win by causing a single request with infinite cost

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Solving Bounded Parity Games with Weights

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Solving Bounded Parity Games with Weights

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Next step: Solving bounded parity games with weights via energy parity games.

Solving Bounded Parity Games with Weights

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Next step: Solving bounded parity games with weights via energy parity games.

Lemma

Bounded parity games with weights can be solved by solving polynomially many energy parity games of polynomial size.

Solving Bounded Parity Games with Weights

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Next step: Solving bounded parity games with weights via energy parity games.

Lemma

Bounded parity games with weights can be solved by solving polynomially many energy parity games of polynomial size.

Proposition (Chatterjee and Doyen, 2012)

Solving energy parity games is in $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Solving Bounded Parity Games with Weights

Lemma

Parity games with weights can be solved by solving polynomially many bounded parity games with weights of polynomial size.

Next step: Solving bounded parity games with weights via energy parity games.

Lemma

Bounded parity games with weights can be solved by solving polynomially many energy parity games of polynomial size.

Proposition (Chatterjee and Doyen, 2012)

Solving energy parity games is in $\mathrm{NP} \cap \mathrm{co\text{-}NP}$.

Theorem

Solving parity games with weights is in $\mathrm{NP} \cap \mathrm{co\text{-}NP}$.

Solving Bounded Parity Games with Weights

Proposition (Daviaud et al., 2018)

Energy parity games can be solved in quasi-pseudo-polynomial time.

Solving Bounded Parity Games with Weights

Proposition (Daviaud et al., 2018)

Energy parity games can be solved in quasi-pseudo-polynomial time.

Corollary (Daviaud et al., 2018)

Parity games with weights can be solved in quasi-pseudo-polynomial time.

There and Back Again

So far: Solved parity games with weights via iteratively solving energy parity games.

There and Back Again

So far: Solved parity games with weights via iteratively solving energy parity games.

Also possible: Solving energy parity games via iteratively solving parity games with weights.

There and Back Again

So far: Solved parity games with weights via iteratively solving energy parity games.

Also possible: Solving energy parity games via iteratively solving parity games with weights.

Proof idea: Again detour via bounded parity games with weights.

There and Back Again

So far: Solved parity games with weights via iteratively solving energy parity games.

Also possible: Solving energy parity games via iteratively solving parity games with weights.

Proof idea: Again detour via bounded parity games with weights.

Lemma

Energy parity games can be solved by solving polynomially many parity games with weights of polynomial size.

There and Back Again

Theorem

The problems of solving parity games with weights and of solving energy parity games are polynomial-time equivalent.

There and Back Again

Theorem

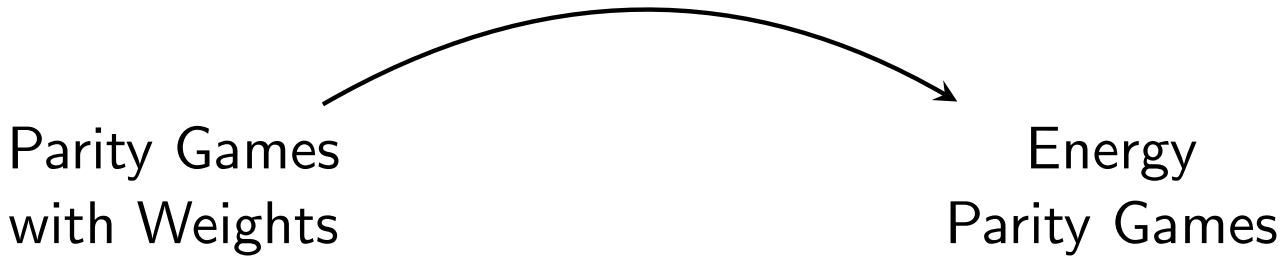
The problems of solving parity games with weights and of solving energy parity games are polynomial-time equivalent.

Parity Games
with Weights

There and Back Again

Theorem

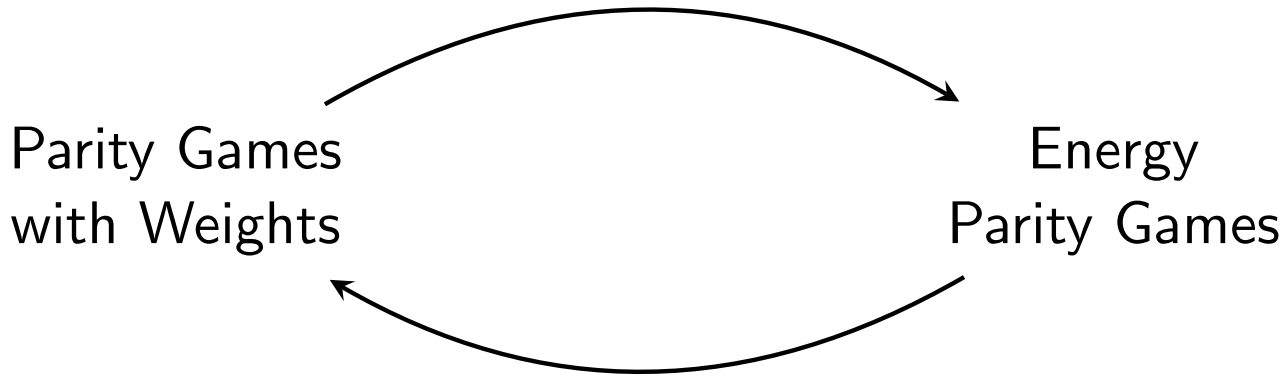
The problems of solving parity games with weights and of solving energy parity games are polynomial-time equivalent.



There and Back Again

Theorem

The problems of solving parity games with weights and of solving energy parity games are polynomial-time equivalent.



Outline

- Definition ✓
- Complexity
- Memory
- Bounds

Outline

- Definition ✓
- Complexity ✓
- Memory
- Bounds

Memory Requirements (Upper Bound)

Known so far: Bounded parity games with weights can be solved by solving polynomially many energy parity games.

Memory Requirements (Upper Bound)

Known so far: Bounded parity games with weights can be solved by solving polynomially many energy parity games.

Intuition:

1. Each energy parity game $\mathcal{G}'_v$ is "tasked" with answering the request posed at vertex v with bounded cost.

Memory Requirements (Upper Bound)

Known so far: Bounded parity games with weights can be solved by solving polynomially many energy parity games.

Intuition:

1. Each energy parity game $\mathcal{G}'_v$ is "tasked" with answering the request posed at vertex v with bounded cost.
2. Player 0 wins $\mathcal{G}$ by simulating winning $\mathcal{G}'_v$ for vertex carrying largest requested color.

Memory Requirements (Upper Bound)

Proposition (Chatterjee and Doyen, 2012)

In an energy parity game with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(ndW)$ suffices for her to implement a winning strategy from her winning region.

Memory Requirements (Upper Bound)

Proposition (Chatterjee and Doyen, 2012)

In an energy parity game with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(ndW)$ suffices for her to implement a winning strategy from her winning region.

Lemma

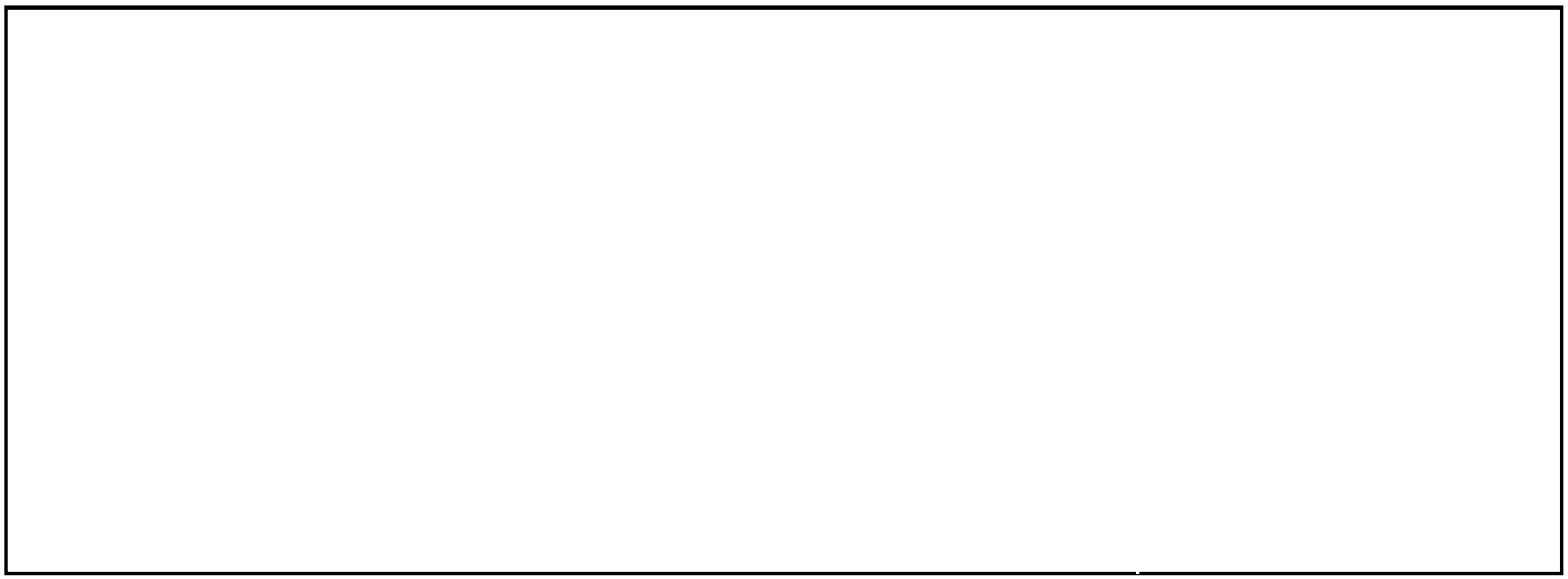
In a bounded parity game with weights with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(nd^2W)$ suffices for Player 0 to implement a winning strategy for her from her winning region.

Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.

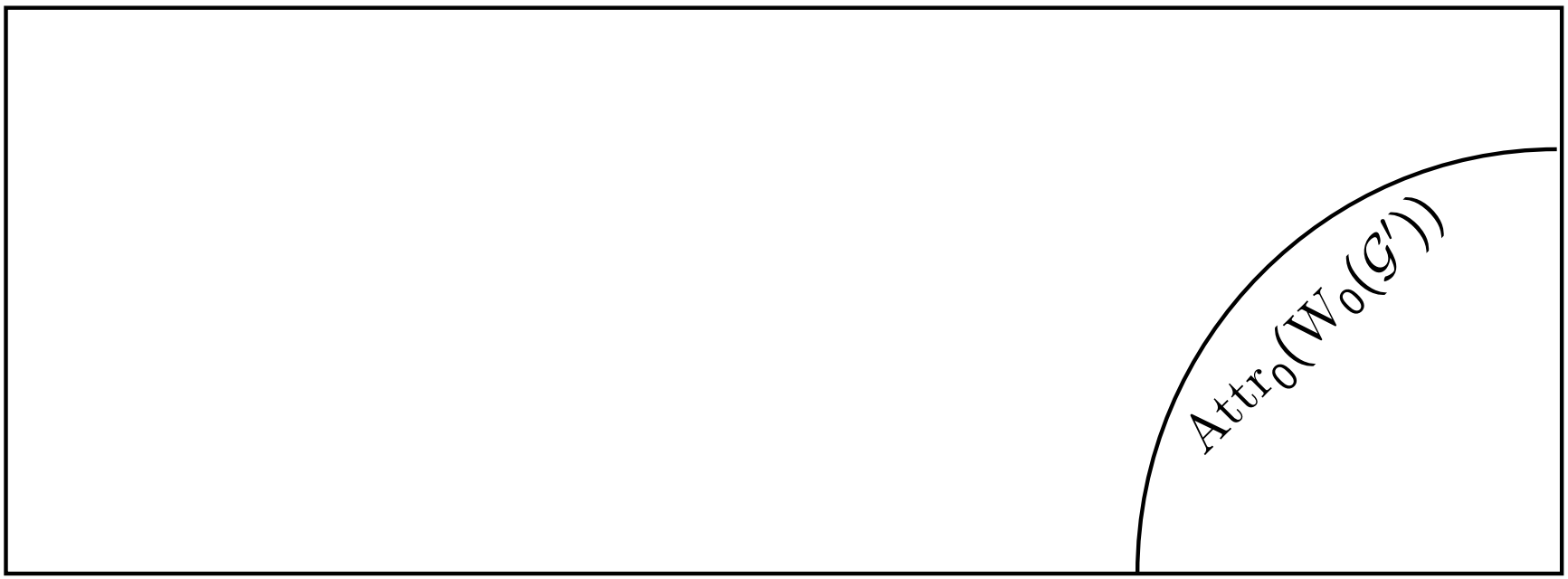
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



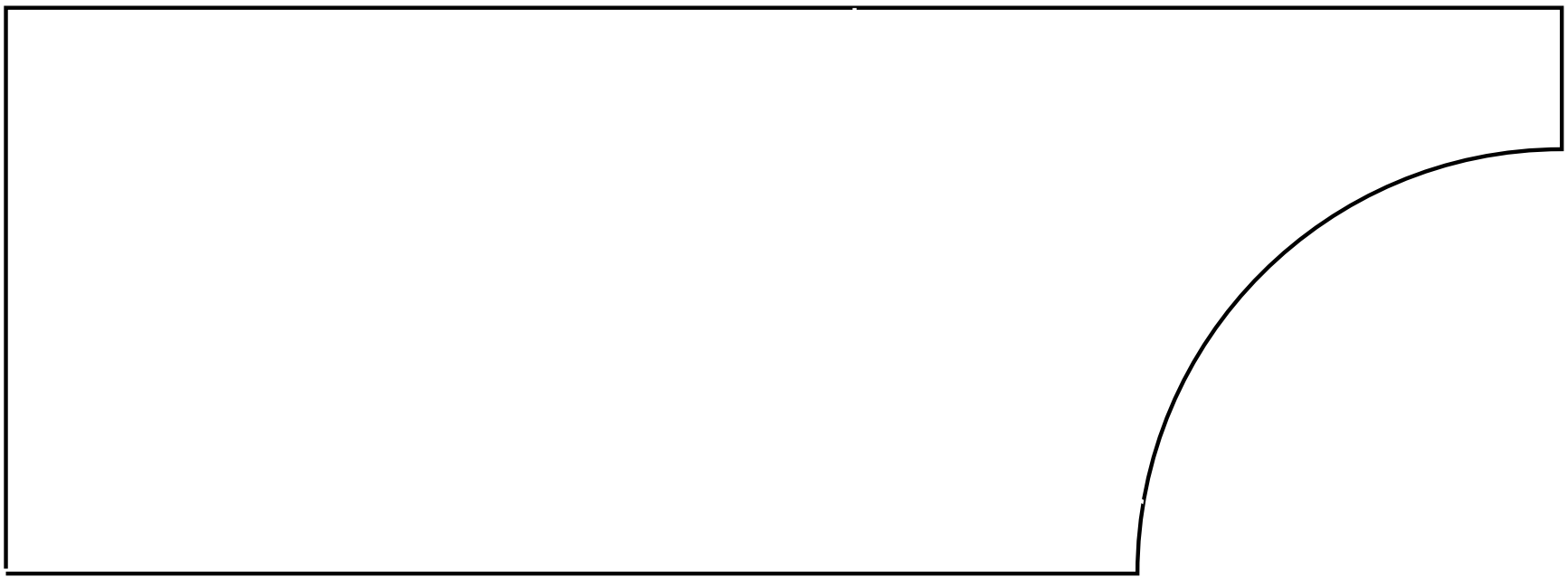
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



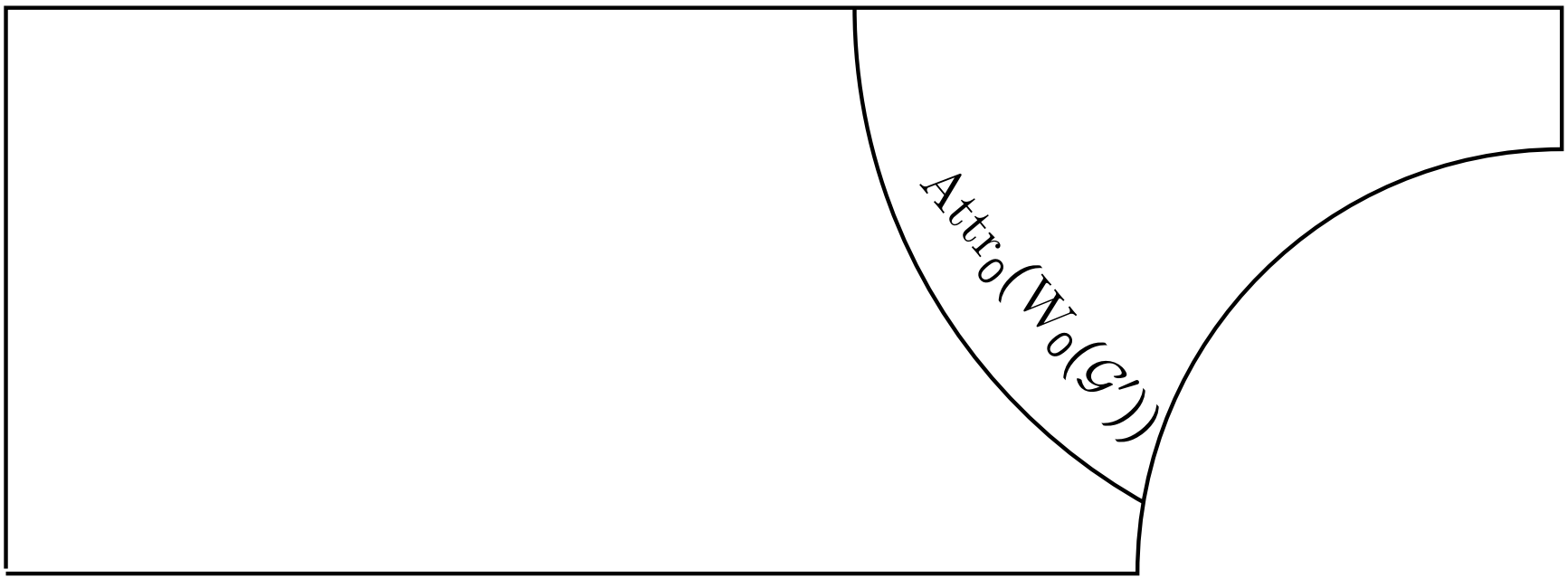
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



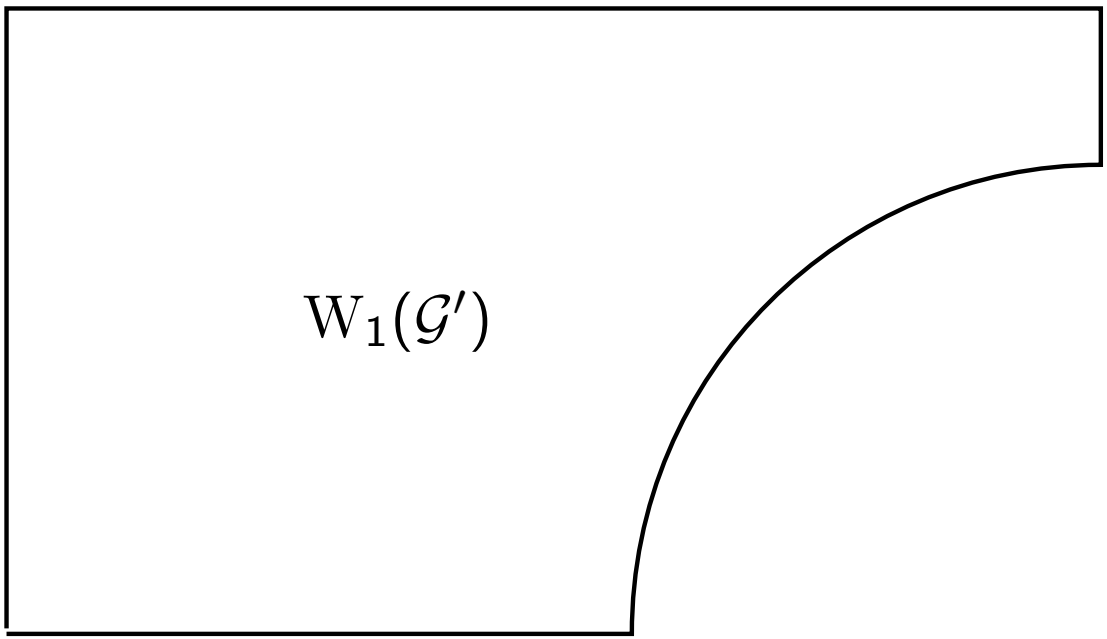
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



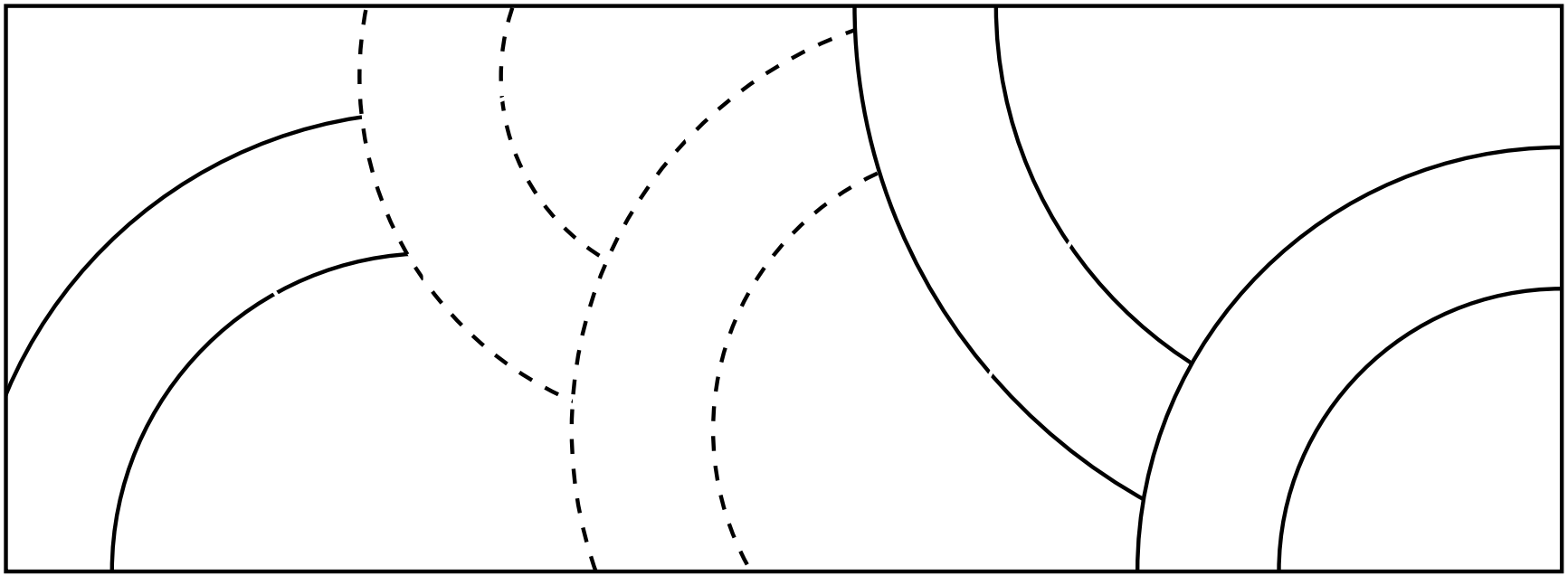
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



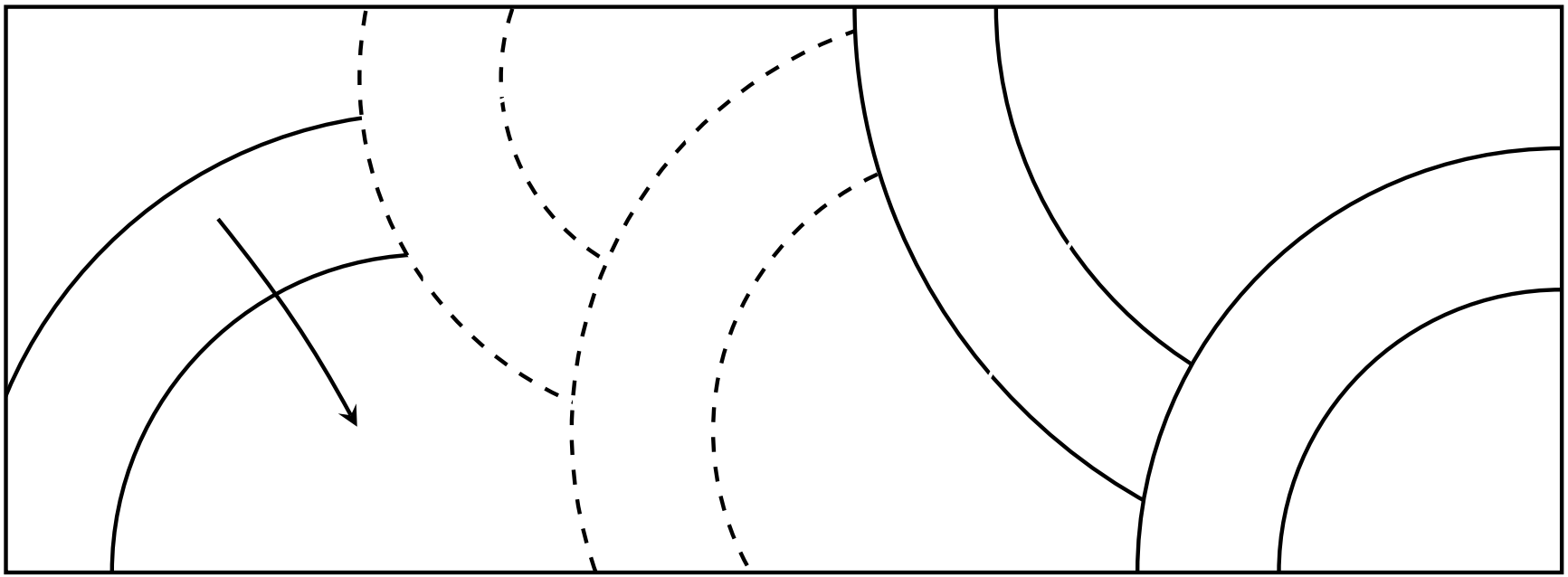
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



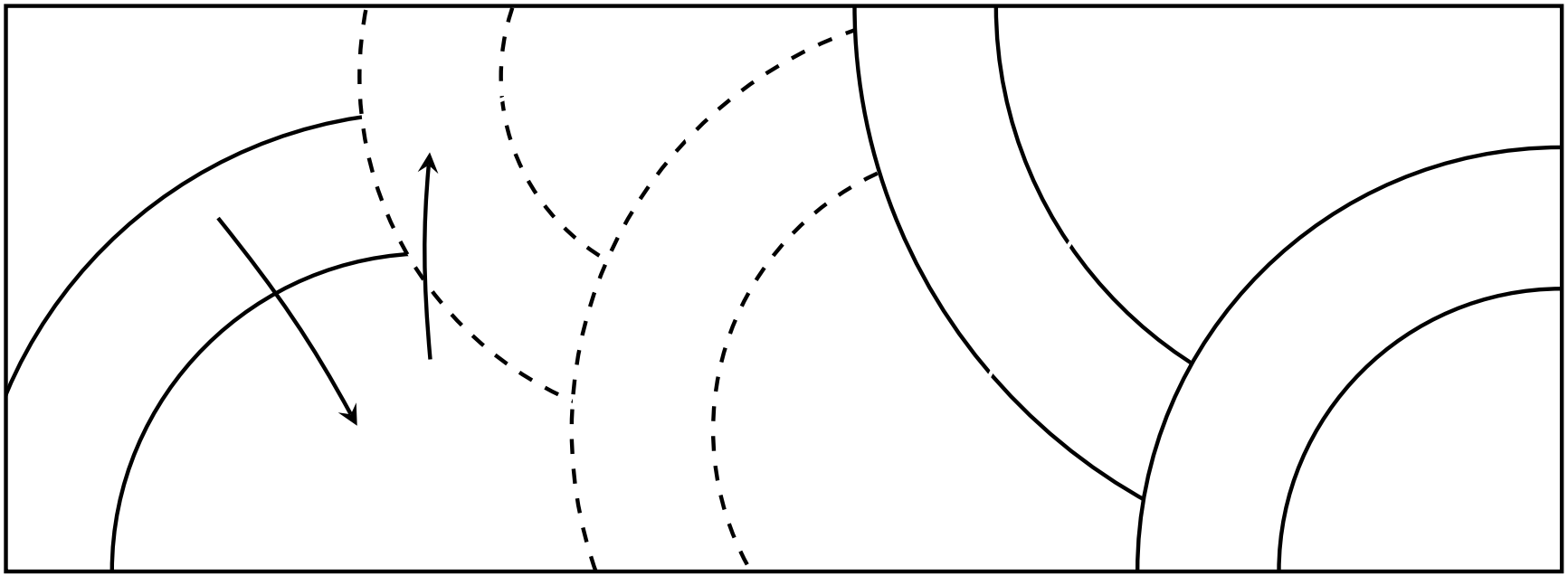
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



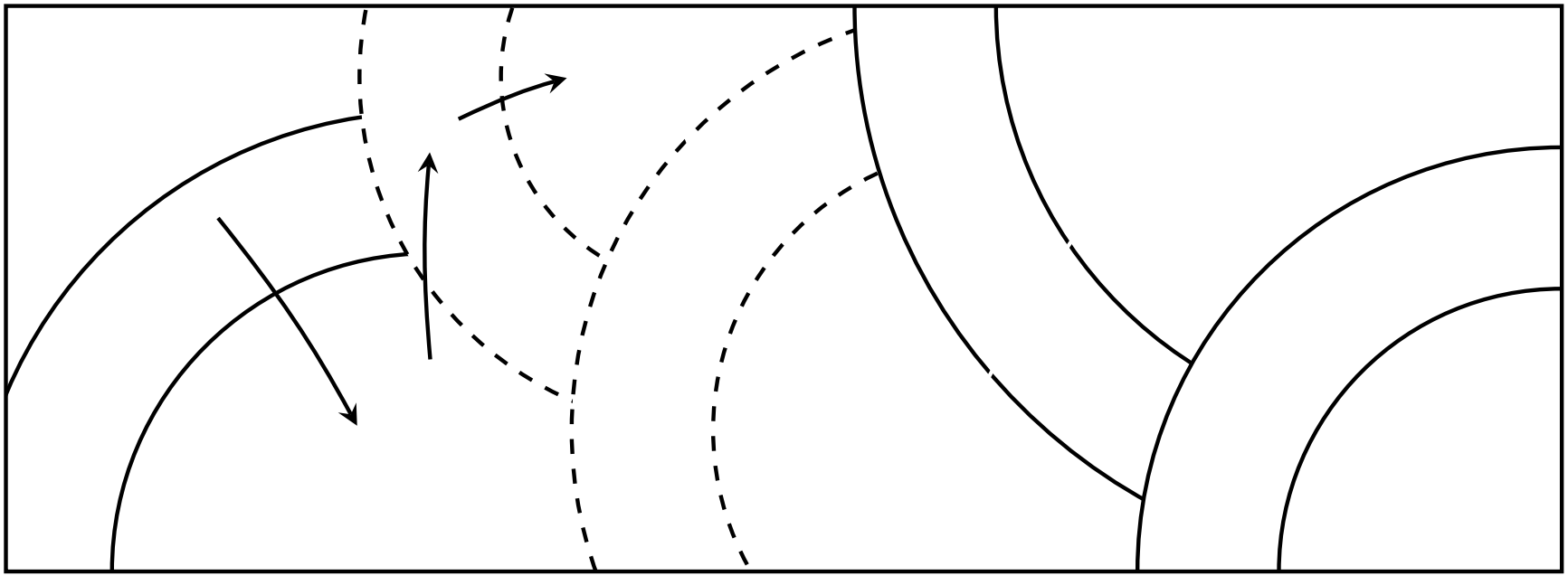
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



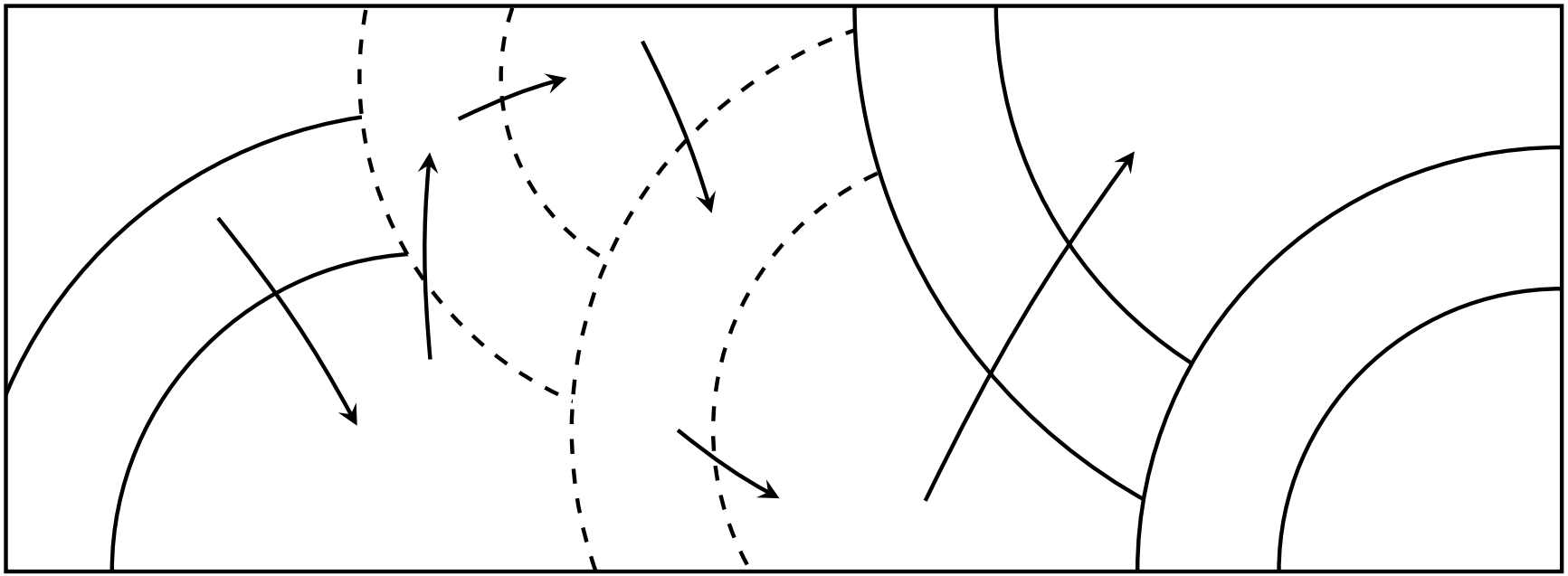
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



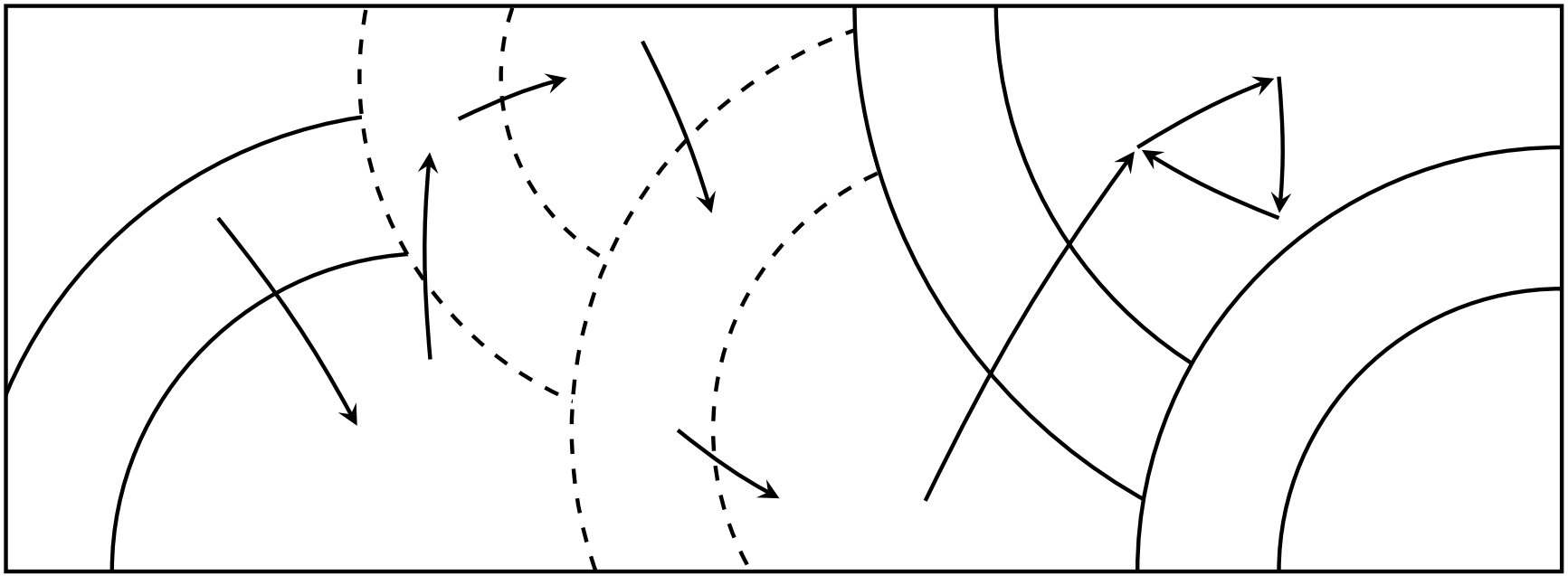
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



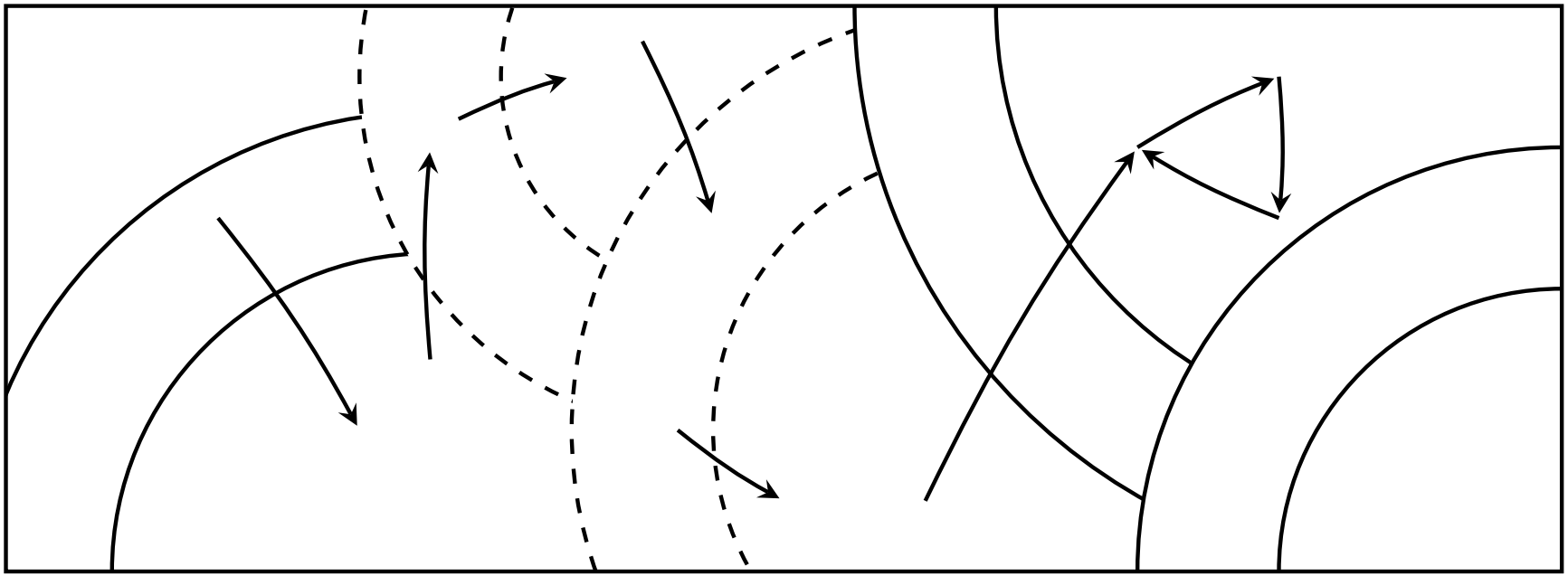
Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



Memory Requirements (Upper Bound)

Known so far: Parity games with weights can be solved by solving polynomially many bounded parity games with weights.



Winning Strategy: At every "border", restart winning bounded parity game with weights

Memory Requirements (Upper Bound)

Lemma

In a bounded parity game with weights with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(nd^2W)$ suffices for Player 0 to implement a winning strategy for her from her winning region.

Memory Requirements (Upper Bound)

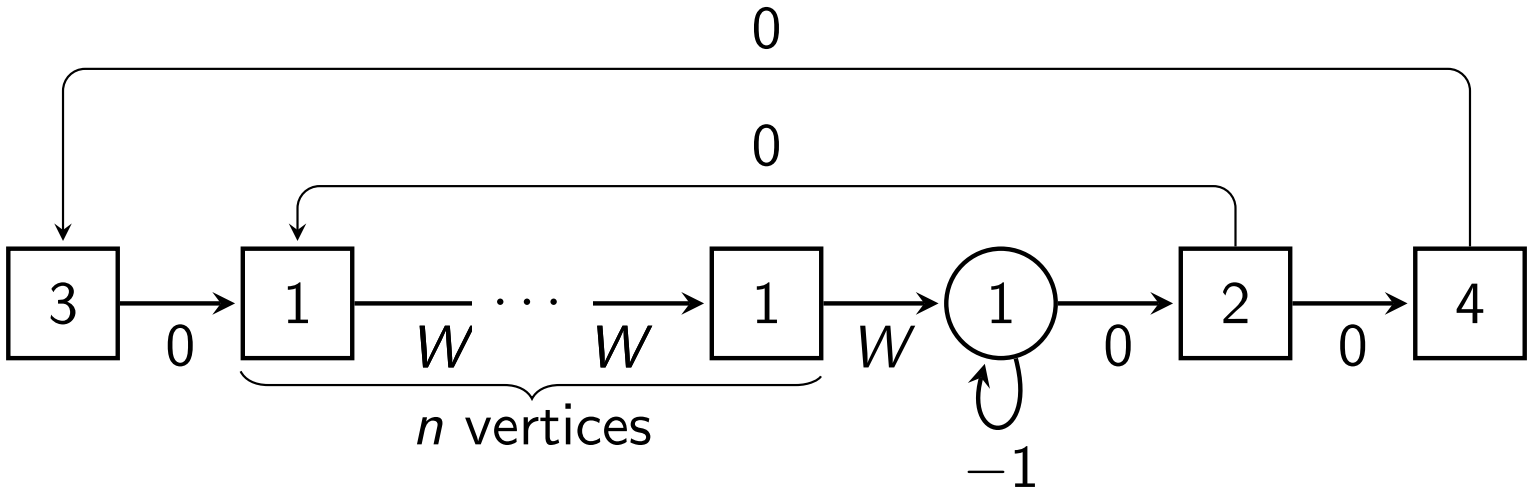
Lemma

In a bounded parity game with weights with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(nd^2W)$ suffices for Player 0 to implement a winning strategy for her from her winning region.

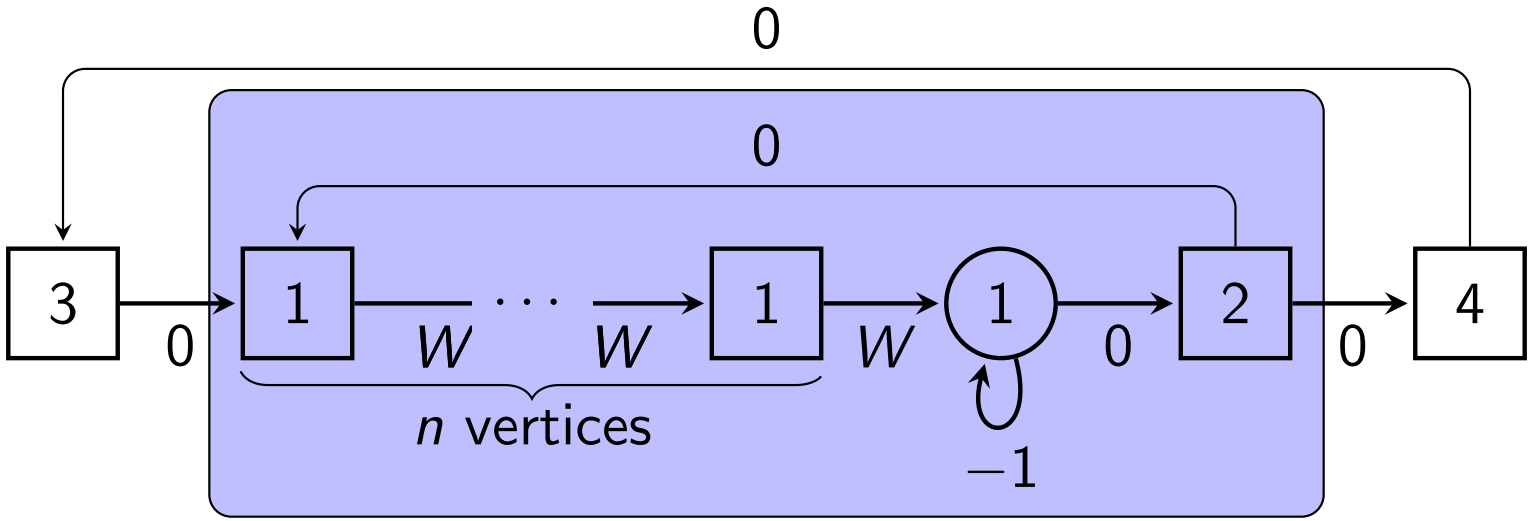
Theorem

In a parity game with weights with n vertices, d odd colors, and largest absolute weight W, memory of size $\mathcal{O}(nd^2W)$ suffices for her to implement a winning strategy from her winning region.

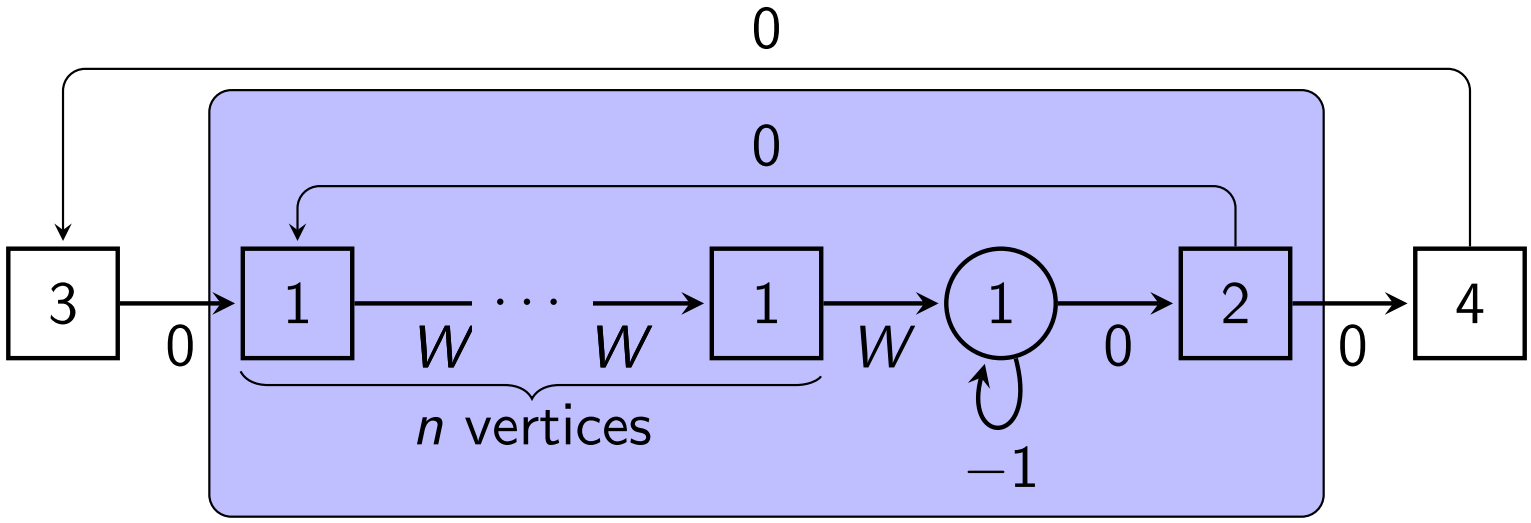
Memory Requirements (Lower Bound)



Memory Requirements (Lower Bound)



Memory Requirements (Lower Bound)



Theorem

Player 0 *requires* $nW + 1$ *memory states to implement a winning strategy in the above game.*

Outline

- Definition ✓
- Complexity ✓
- Memory
- Bounds

Outline

- Definition ✓
- Complexity ✓
- Memory ✓
- Bounds

Weight Bounds (Upper Bounds)

Proposition (Chatterjee and Doyen, 2012)

In an energy parity game $\mathcal{G}$ with n vertices, d odd colors, and largest absolute weight W, if Player 0 wins $\mathcal{G}$ from some vertex v, then she has a strategy that bounds the energy level from below by $-nW$.

Weight Bounds (Upper Bounds)

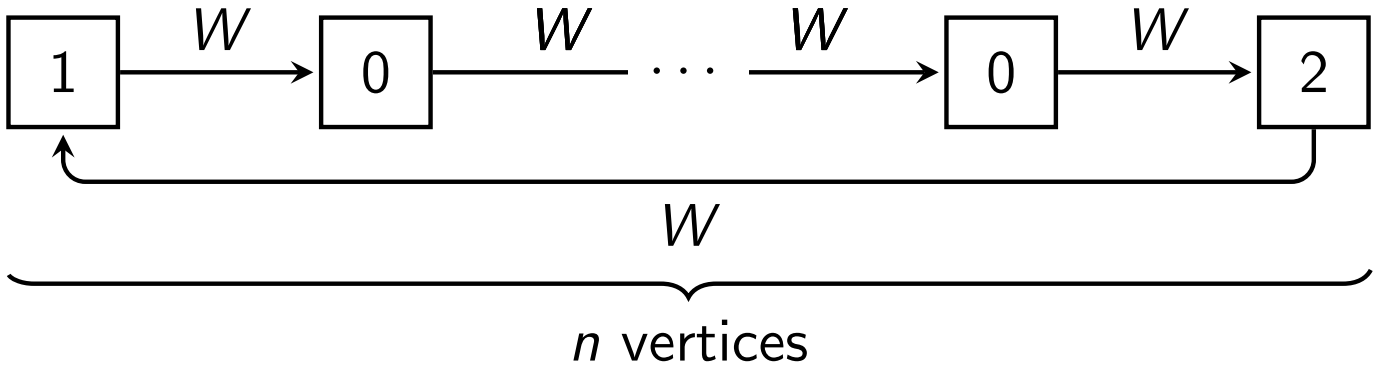
Proposition (Chatterjee and Doyen, 2012)

In an energy parity game $\mathcal{G}$ with n vertices, d odd colors, and largest absolute weight W, if Player 0 wins $\mathcal{G}$ from some vertex v, then she has a strategy that bounds the energy level from below by $-nW$.

Theorem

In a parity game with weights $\mathcal{G}$ with n vertices, d odd colors, and largest absolute weight W, if Player 0 wins $\mathcal{G}$ from some vertex v, the she has a strategy that bounds the cost of plays by some value in $\mathcal{O}((ndW)^2)$.

Weight Bounds (Lower Bounds)



Theorem

Player 0 *wins the above game, but every play has a cost of* $(n-1)W$.

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$
Memory	positional	positional	

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Conclusion

	Finitary	Costs	Weights
Complexity	$\mathrm{P}\textsc{Time}$	$\mathrm{UP} \cap \textsc{co-UP}$	$\mathrm{NP} \cap \textsc{co-NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Future Work:

- Tighten bounds
  - Lower bound on memory: $nW + 1$
  - Lower bound on cost: nW

Conclusion

	Finitary	Costs	Weights
Complexity	$\mathrm{P}\textsc{Time}$	$\mathrm{UP} \cap \textsc{co-UP}$	$\mathrm{NP} \cap \textsc{co-NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Future Work:

- Tighten bounds
  - Lower bound on memory: $nW + 1$
  - Lower bound on cost: nW
- Different cost measures

Conclusion

	Finitary	Costs	Weights
Complexity	$\mathrm{P}\textsc{Time}$	$\mathrm{UP} \cap \mathrm{co\text{-}UP}$	$\mathrm{NP} \cap \mathrm{co\text{-}NP}$
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Future Work:

- Tighten bounds
  - Lower bound on memory: $nW + 1$
  - Lower bound on cost: nW
- Different cost measures
- Multidimensional parity games with weights

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	UP ∩ CO-UP	NP ∩ CO-NP
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Future Work:

- Tighten bounds
  - Lower bound on memory: $nW + 1$
  - Lower bound on cost: nW
- Different cost measures
- Multidimensional parity games with weights
- Threshold problem: Enforce cost at most b

Conclusion

	Finitary	Costs	Weights
Complexity	PTIME	UP ∩ CO-UP	NP ∩ CO-NP
Memory	positional	positional	$\mathcal{O}(nd^2W)$
Bounds	n	nW	$\mathcal{O}((ndW)^2)$

Future Work:

- Tighten bounds
  - Lower bound on memory: $nW + 1$
  - Lower bound on cost: nW
- Different cost measures
- Multidimensional parity games with weights
- Threshold problem: Enforce cost at most b (EXPTIME-complete)