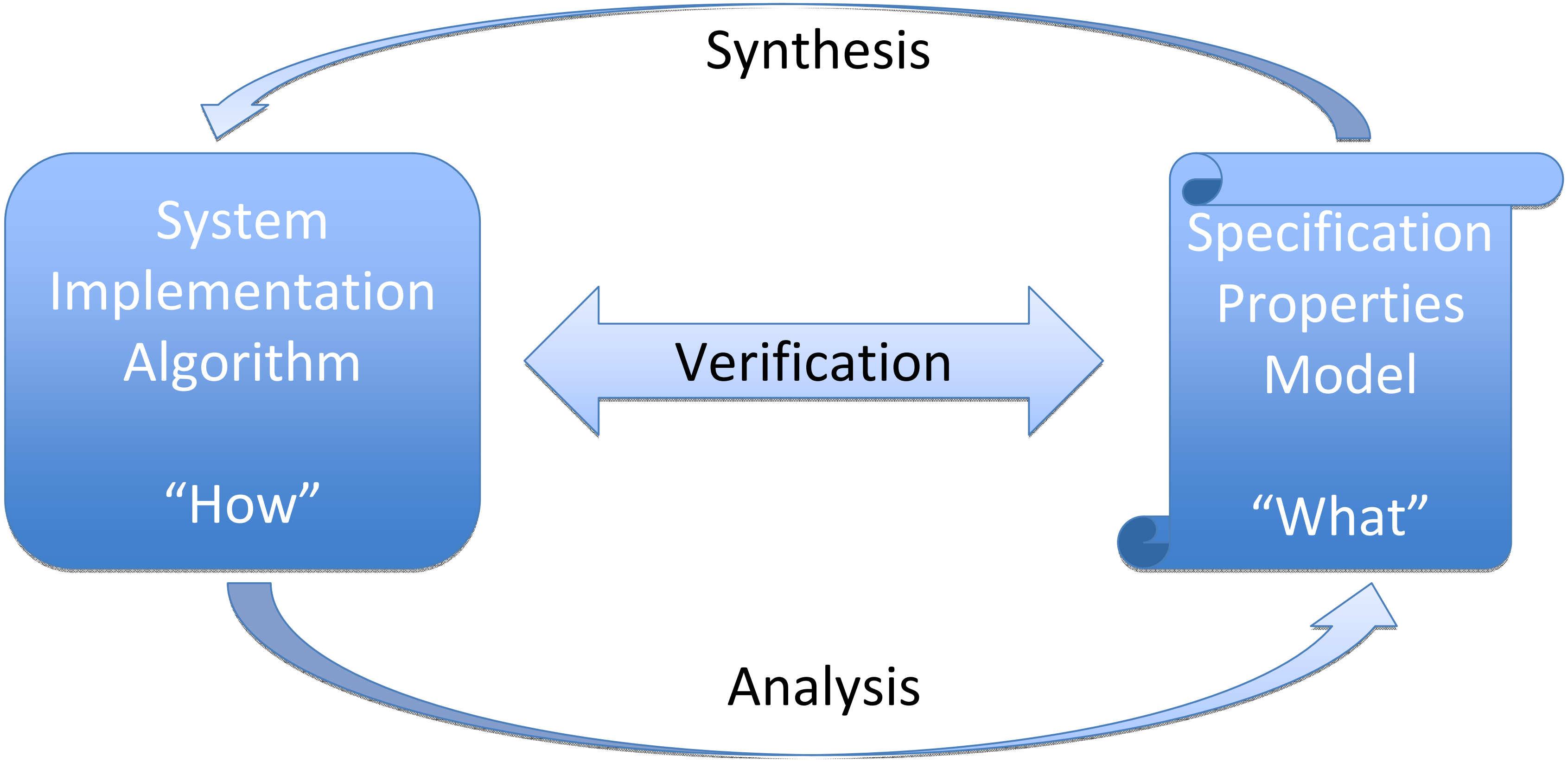
Energy Parity Games

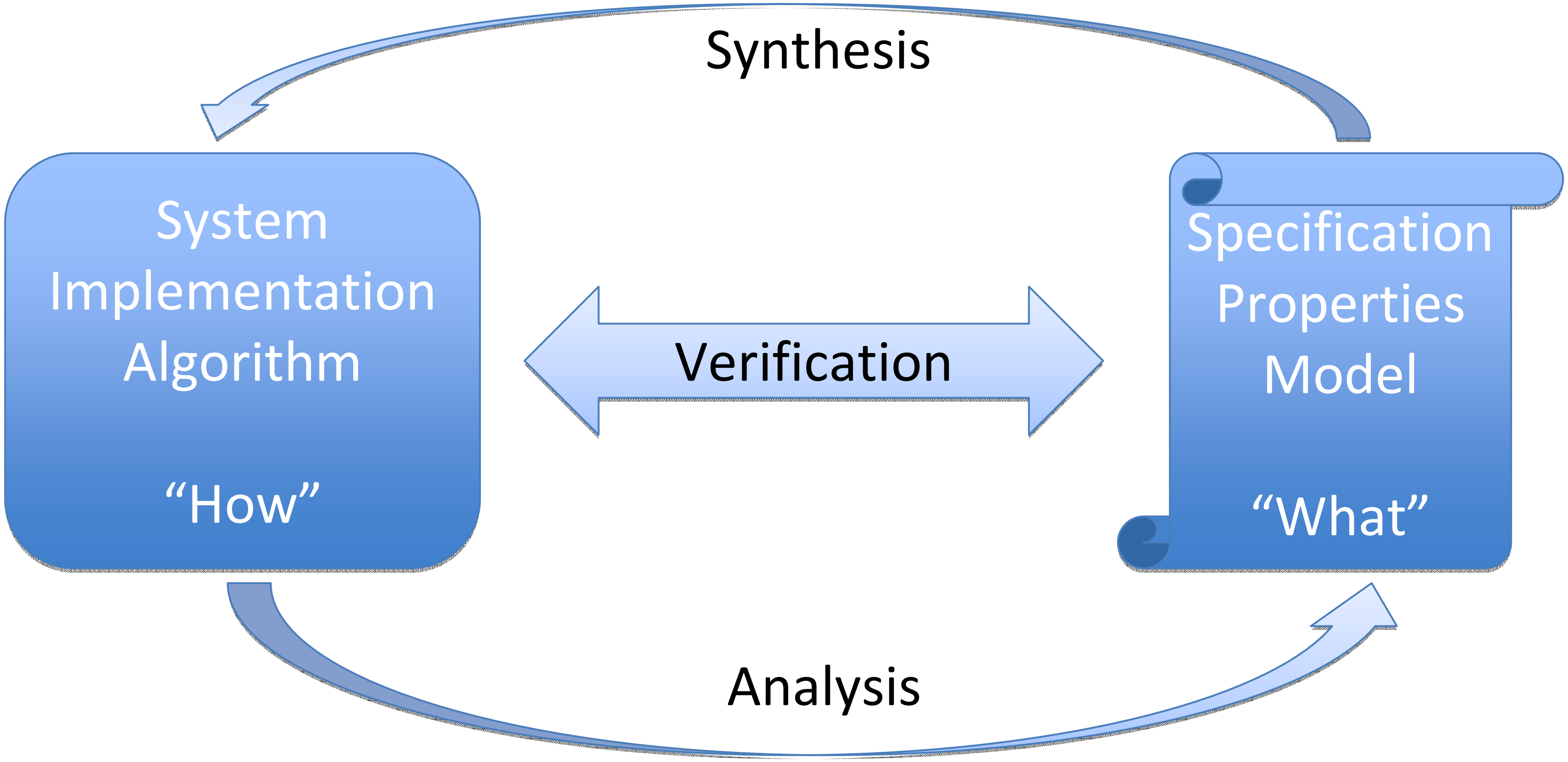
Laurent Doyen
LSV, ENS Cachan & CNRS

Krishnendu Chatterjee
IST Austria

Analysis - Synthesis



Analysis - Synthesis



Reactive Systems (e.g. servers, hardware,…)

- Interact with environment
- Non-terminating
- Finite data (or data abstractions)
- Control-oriented

Two-player games on graphs

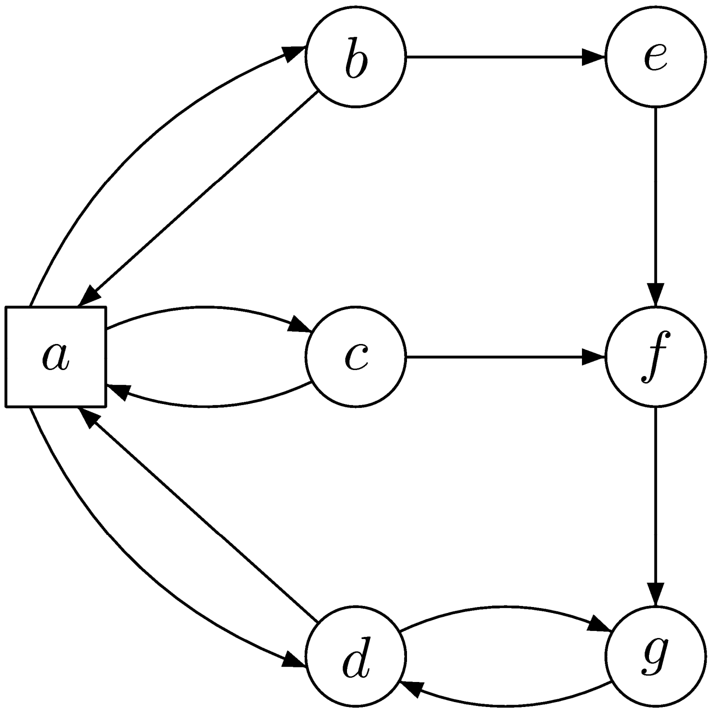
- Games for synthesis
  - Reactive system synthesis = finding a winning strategy in a two-player game
  - ω-regular spec : safety, reactivity, …

$$\Box\neg(g_1 \wedge g_2),\ \Box(r \rightarrow \Diamond g),\ \ldots$$

  - quantitative spec : resource constraints
- Game played on a finite graph
  - Infinite number of rounds
  - Player’s moves determine successor state
  - Outcome = infinite path in the graph

Two-player games

Two-player games

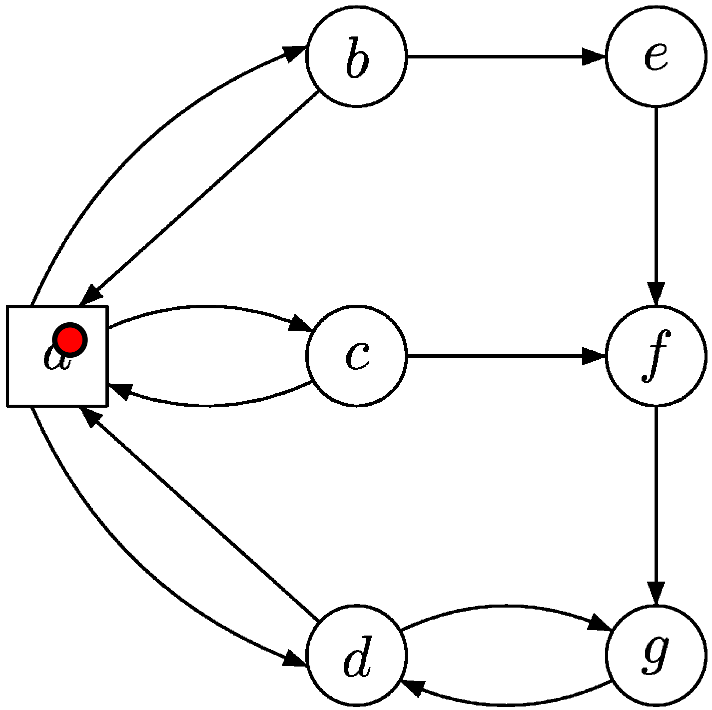


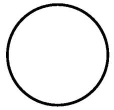
Player 1 (good guy)

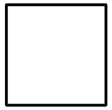
Player 2 (bad guy)

- Turn-based
- Infinite

Two-player games



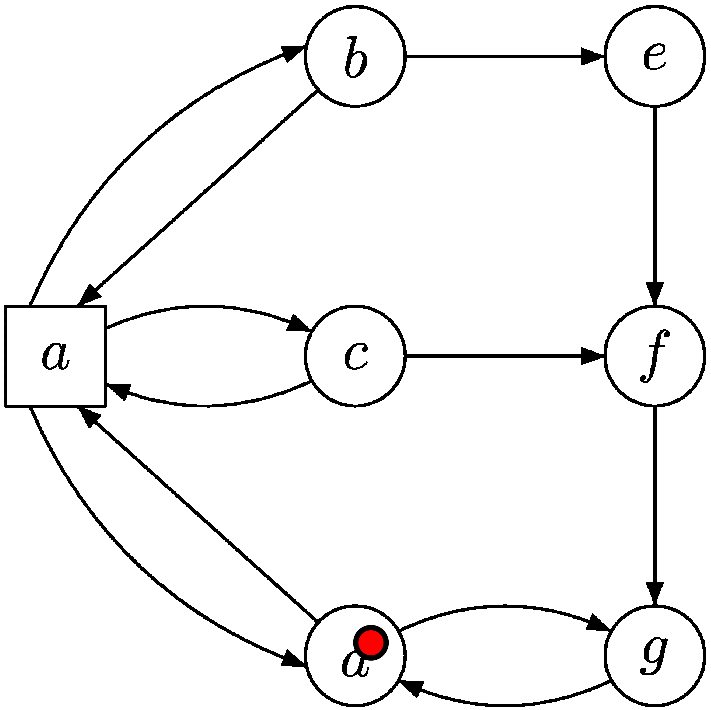
 Player 1 (good guy)

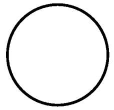
 Player 2 (bad guy)

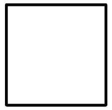
- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games



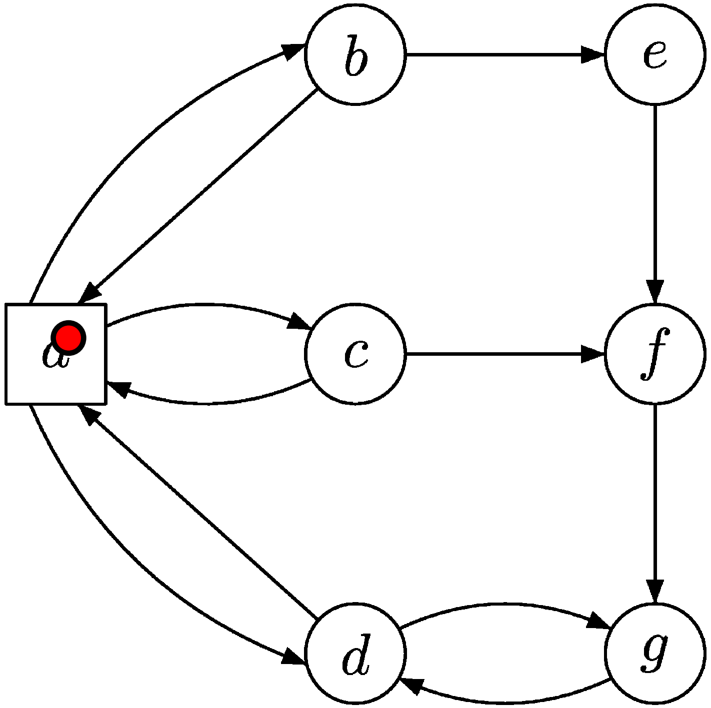
 Player 1 (good guy)

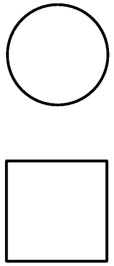
 Player 2 (bad guy)

- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games

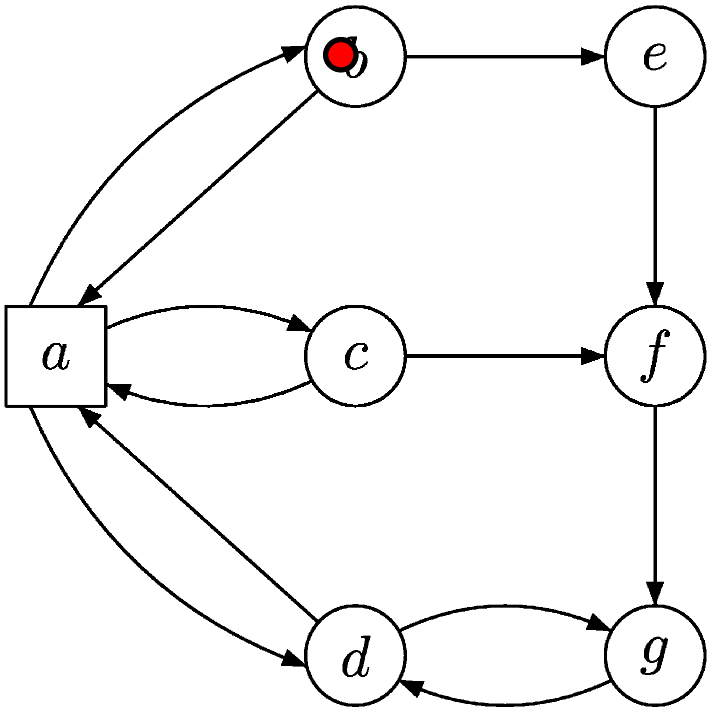




- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games



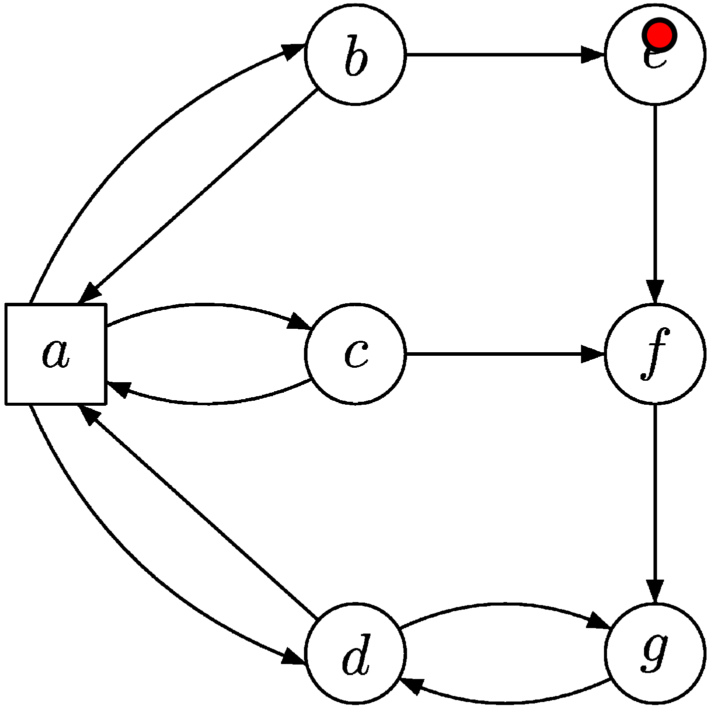
Player 1 (good guy)

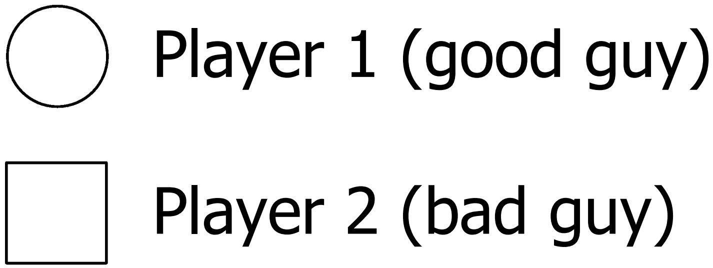
Player 2 (bad guy)

- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games

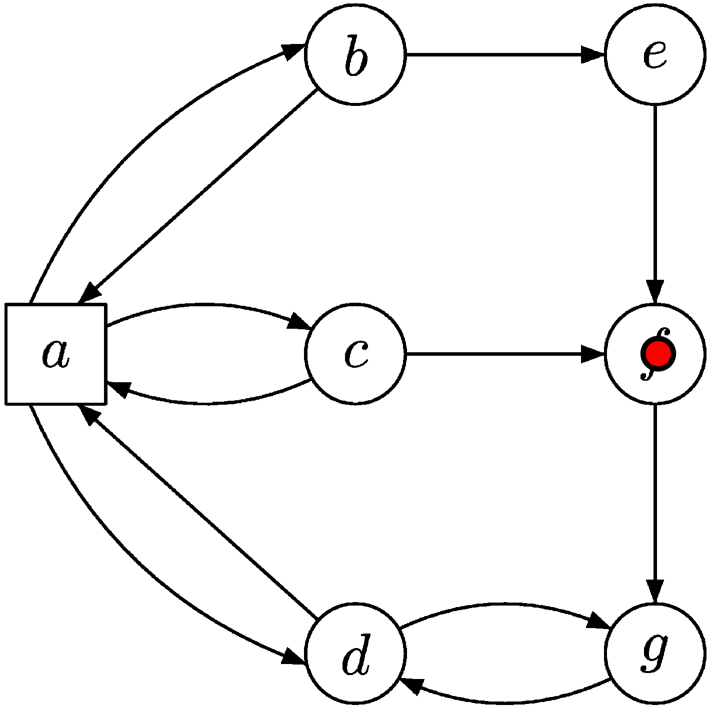


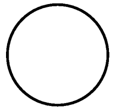


- Turn-based
- Infinite

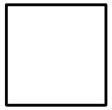
Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games





Player 1 (good guy)

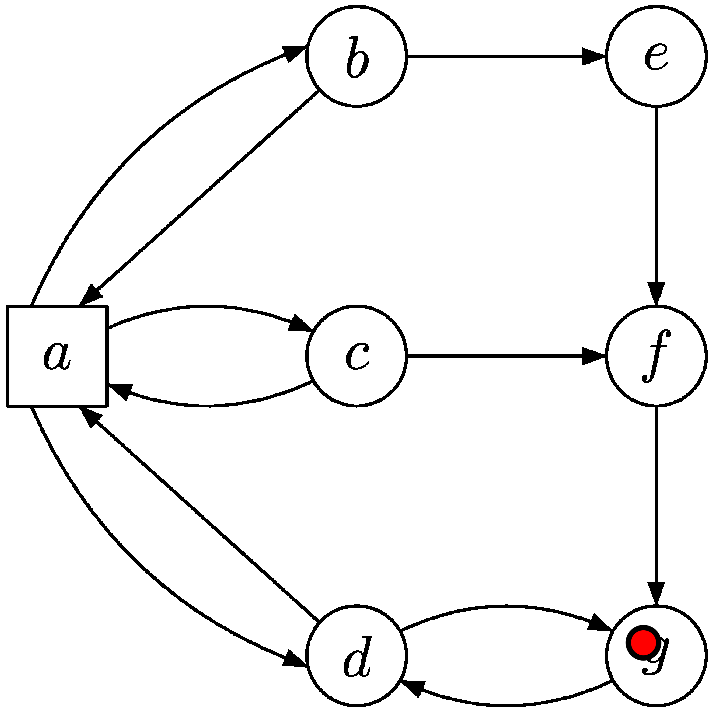


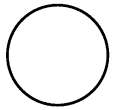
Player 2 (bad guy)

- Turn-based
- Infinite

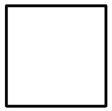
Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games





Player 1 (good guy)

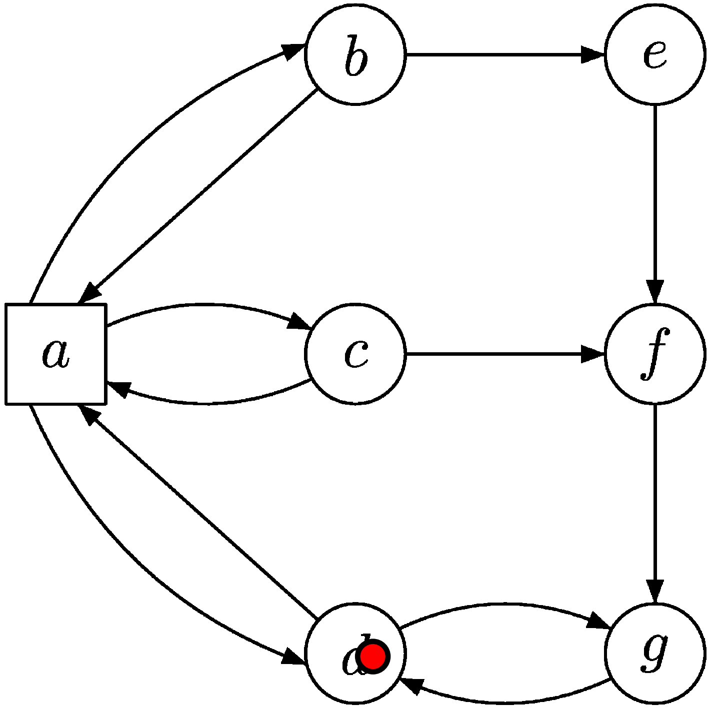


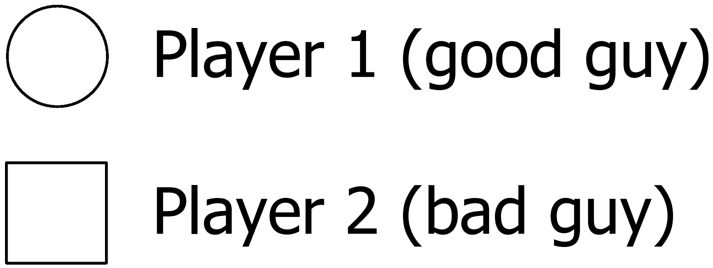
Player 2 (bad guy)

- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games

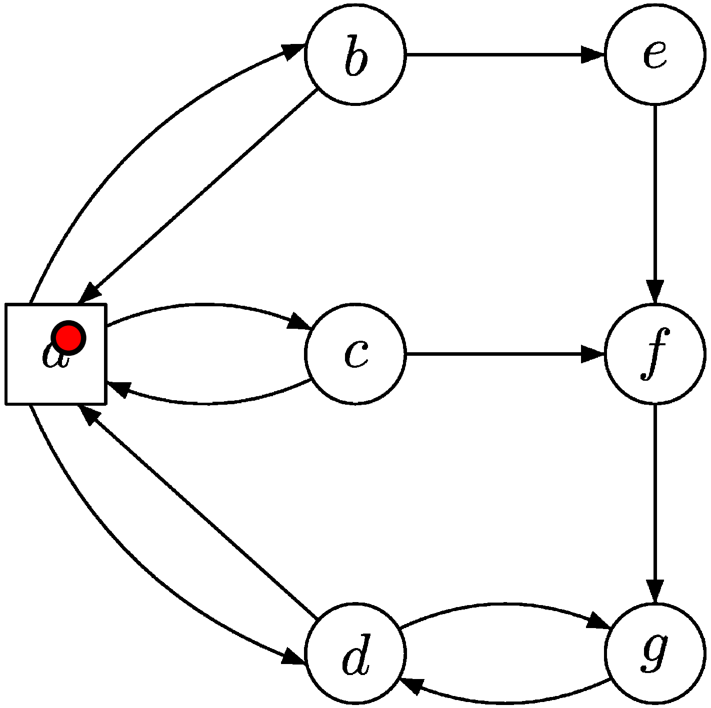


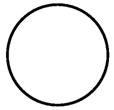


- Turn-based
- Infinite

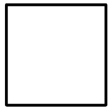
Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games





Player 1 (good guy)

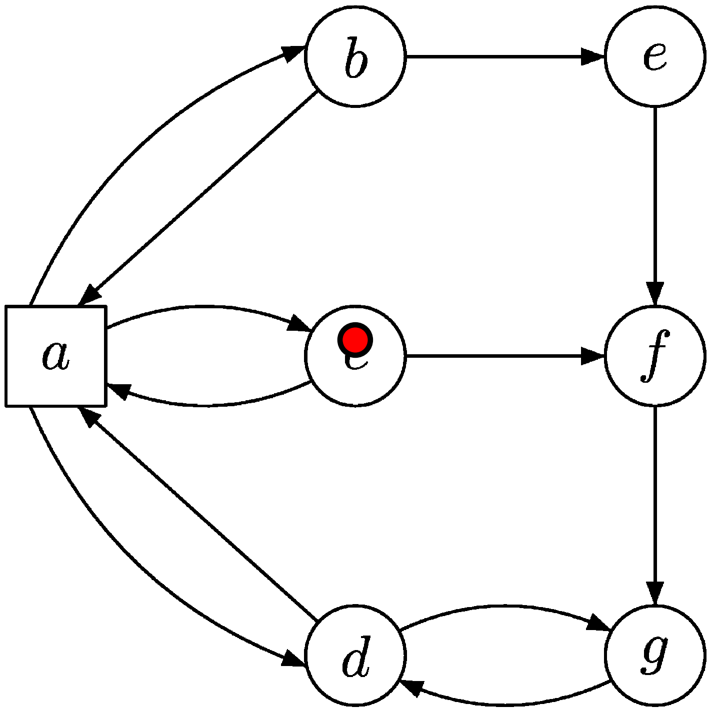


Player 2 (bad guy)

- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games



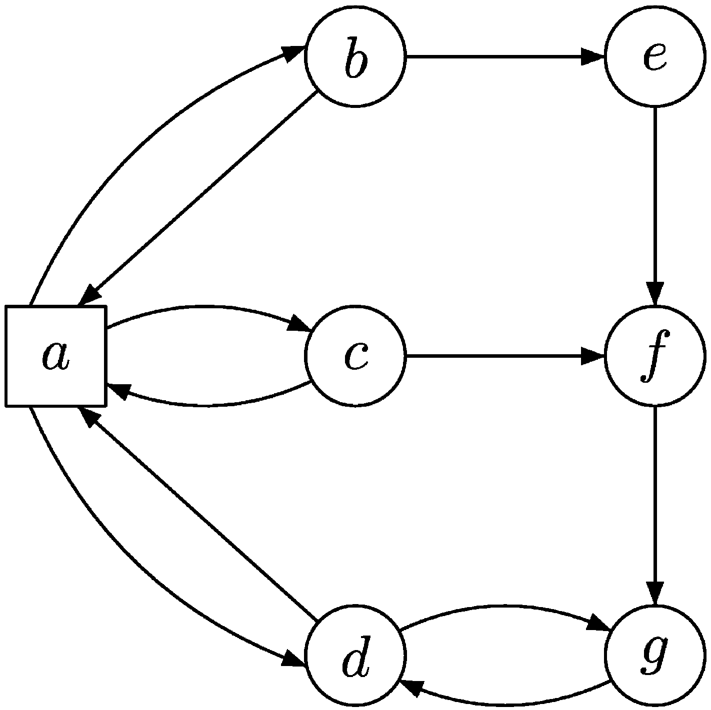
Player 1 (good guy)

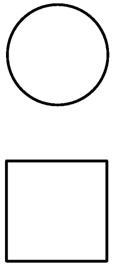
Player 2 (bad guy)

- Turn-based
- Infinite

Play: $a,\ d,\ a,\ b,\ e,\ f,\ g,\ d,\ a,\ c,\ \ldots$

Two-player games





Player 1 (good guy)

Player 2 (bad guy)

- Turn-based
- Infinite

Strategies = recipe to extend the play prefix

$$\left.\begin{array}{l}\text{Player 1: } \sigma : Q^* \cdot Q_{\circ} \to Q \\ \text{Player 2: } \pi : Q^* \cdot Q_{\Box} \to Q\end{array}\right\} \text{outcome of two strategies is a play}$$

Two-player games on graphs

Qualitative

Parity games

ω-regular specifications
(reactivity, liveness,…)

$\Box\neg(g_1 \wedge g_2)$

$\Box(r \rightarrow \Diamond g)$

$\Box\Diamond r \rightarrow \Box\Diamond g$

Quantitative

Energy games

Resource-constrained
specifications

Two-player games on graphs

Qualitative

Parity games

ω-regular specifications
(reactivity, liveness,…)

Quantitative

Energy games

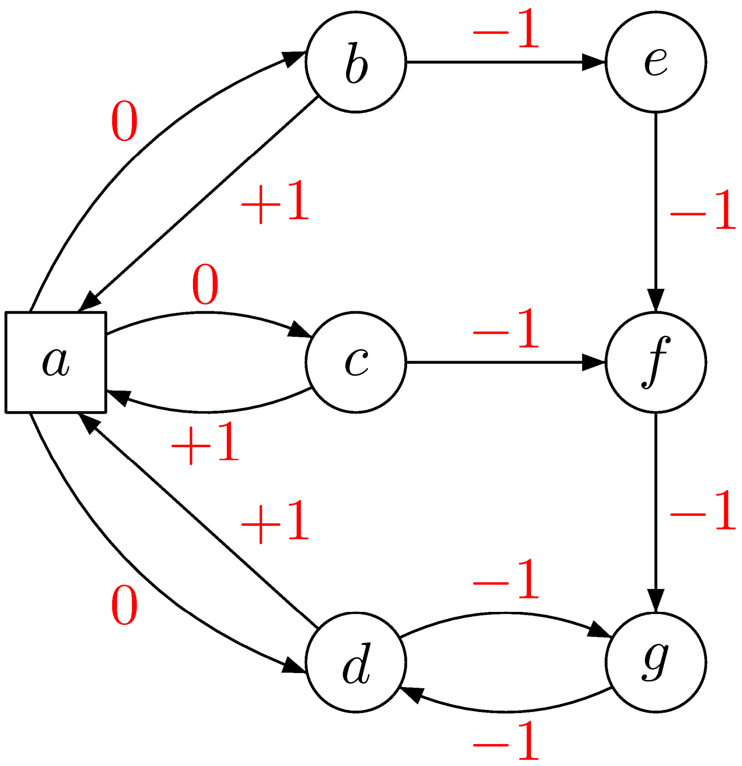
Resource-constrained
specifications

Mixed qualitative-quantitative

Energy parity games

Energy games

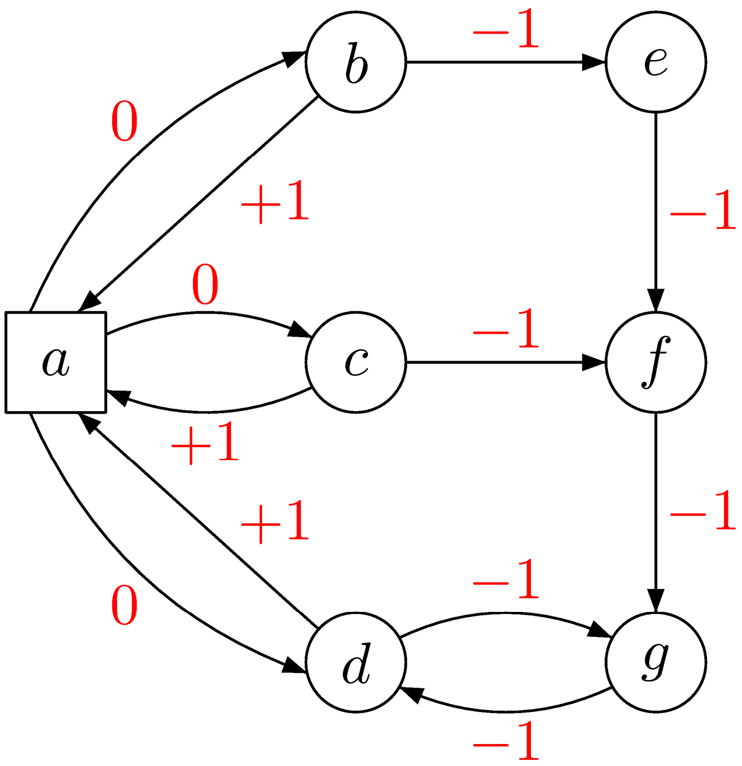
Energy games



Energy game:

Positive and negative weights

Energy games



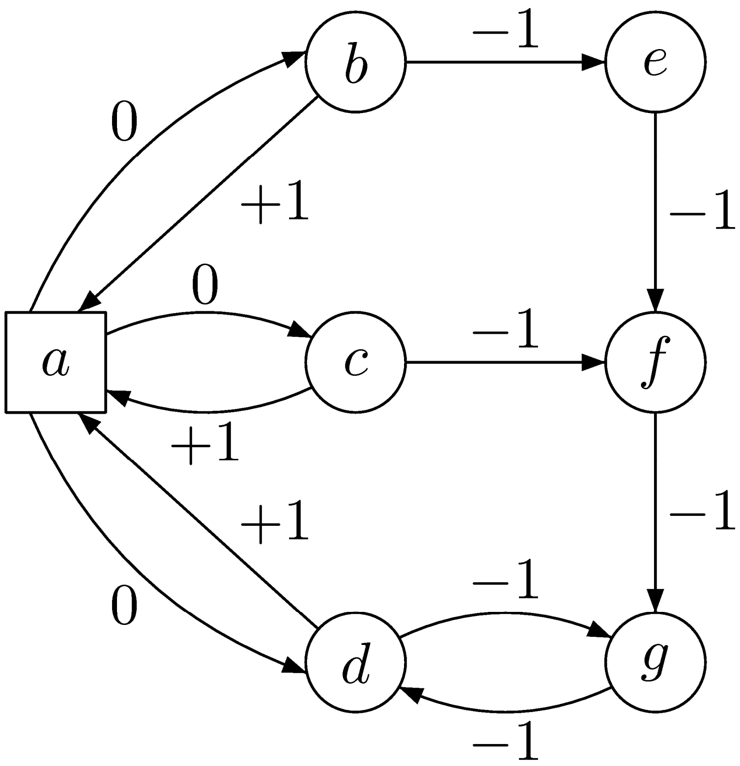
Energy game:

Positive and negative weights

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$

Energy level: 3, 3, 4, 4, 3, 2, 1,…
(sum of weights)

Energy games



Energy game:

Positive and negative weights

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$

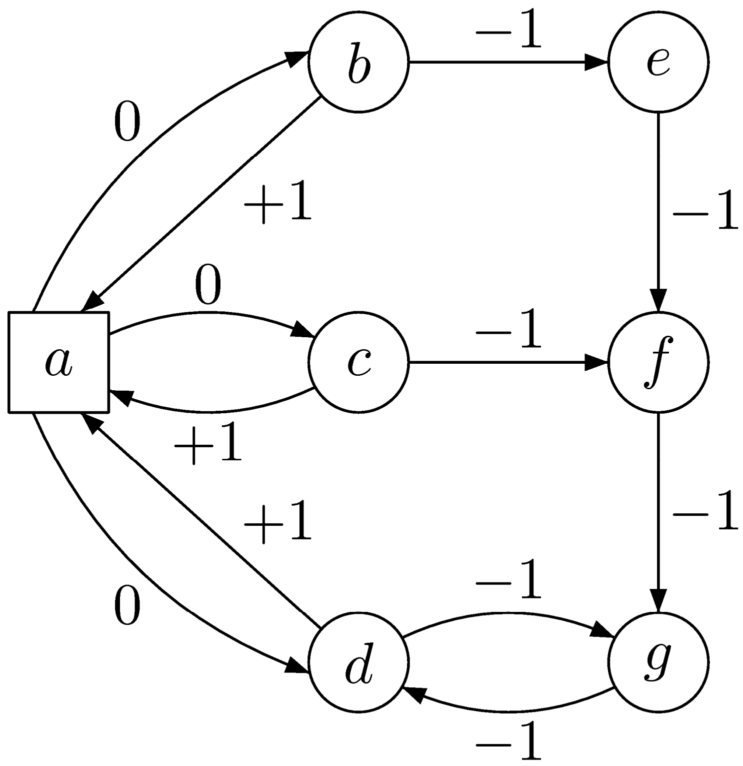
Energy level: **3** 3, 4, 4, 3, 2, 1,…

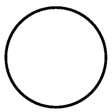
Initial credit

A play is **winning** if the energy level is always nonnegative.

“Never exhaust the resource (memory, battery, …)”

Energy games

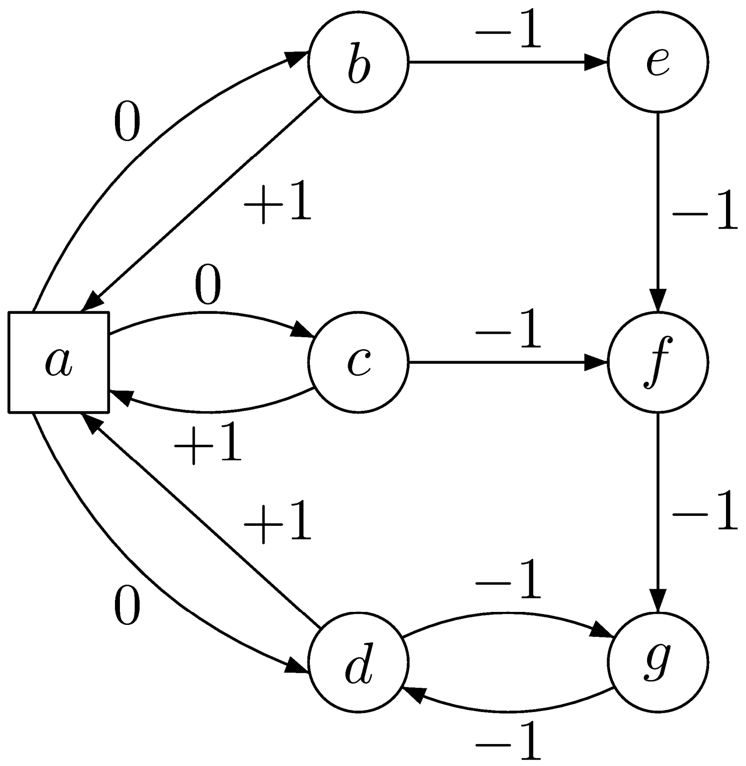


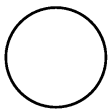
 Player 1 (good guy)

Initial credit problem:

Decide if there exist an initial credit c_0 and a strategy of player 1 to maintain the energy level always nonnegative.

Energy games





Player 1 (good guy)

Initial credit problem:

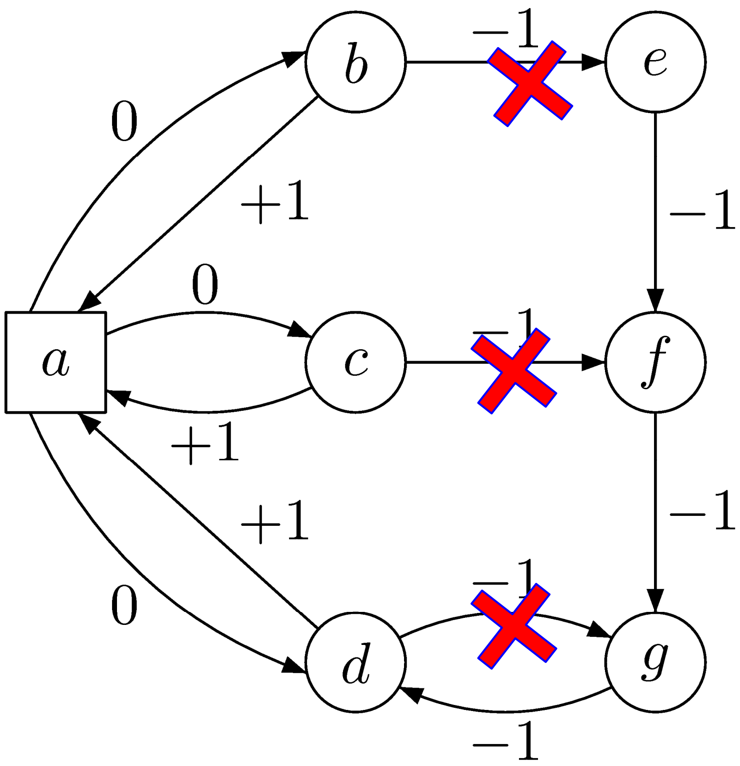
Decide if there exist an initial credit c_0 and a strategy of player 1 to maintain the energy level always nonnegative.

For energy games, memoryless strategies suffice.

$$\sigma : Q_{\Box} \to Q$$

$$\pi : Q_{\circ} \to Q$$

Energy games

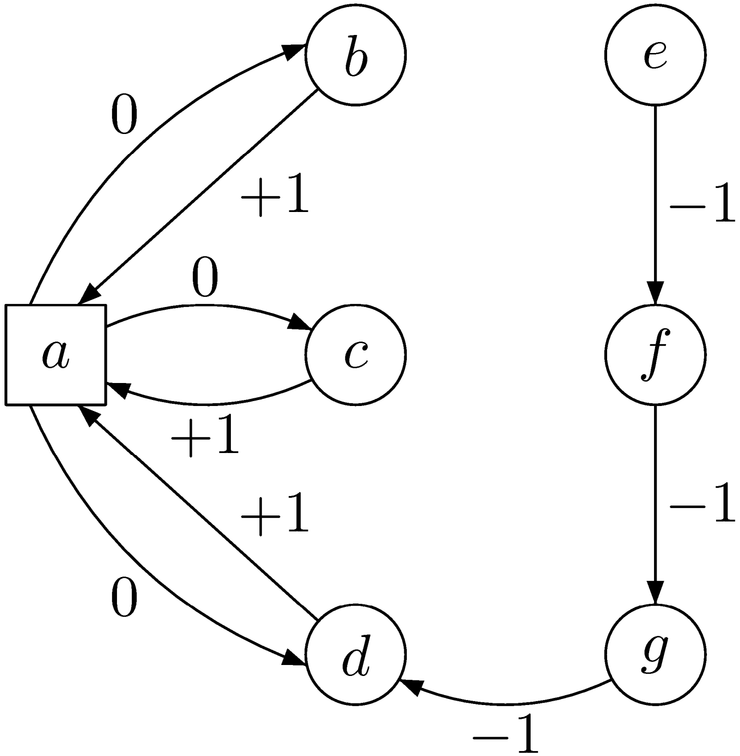


Initial credit problem:

Decide if there exist an initial credit c_0 and a strategy of player 1 to maintain the energy level always nonnegative.

A memoryless strategy σ is winning if all cycles are **nonnegative** when σ is fixed.

Energy games



Initial credit problem:

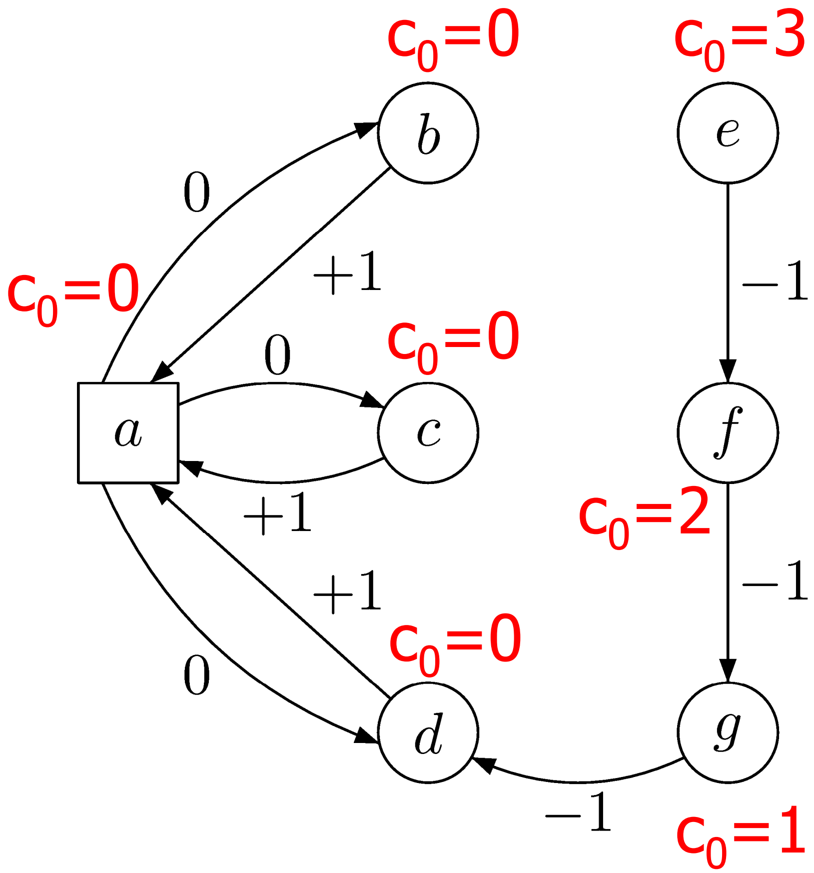
Decide if th… …ial cred… …of pla… …tain the energy leve… …ys nonnegative.

NP ∩ coNP

See [CdAHS03, BFLMS08]

A memoryless strategy σ is winning if all cycles are **nonnegative** when σ is fixed.

Energy games



Initial credit problem:

Decide if th... ial cred... of pla... ain the energy leve... ys nonnegative.

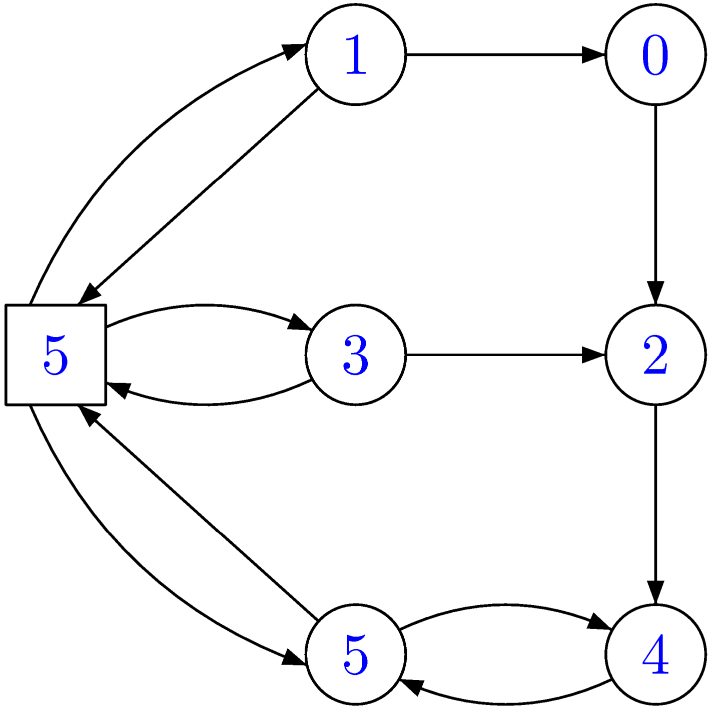
NP ∩ coNP

Minimum initial credit can be computed in $O(|E| \cdot |Q| \cdot W)$

A memoryless strategy σ is winning if all cycles are **nonnegative** when σ is fixed.

Parity games

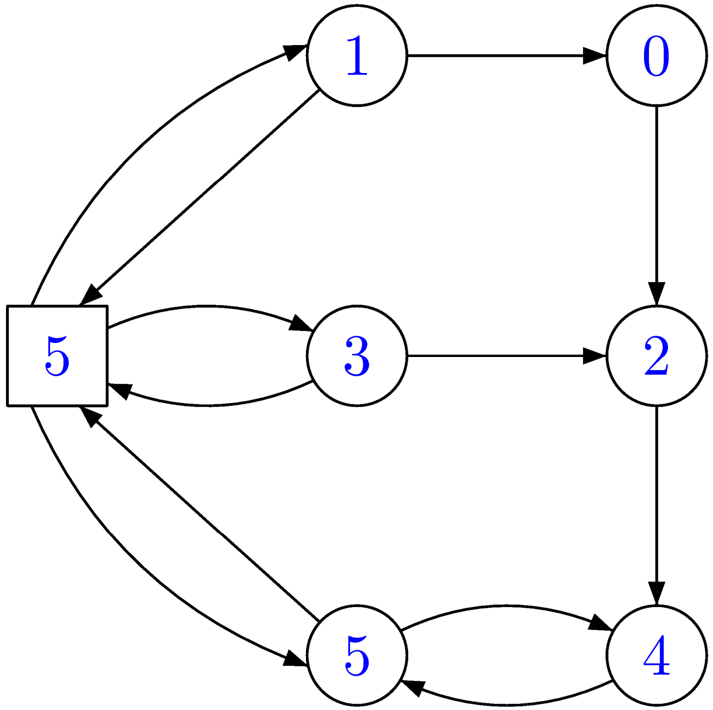
Parity games



Parity game:

integer priority on states

Parity games



Parity game:

integer priority on states

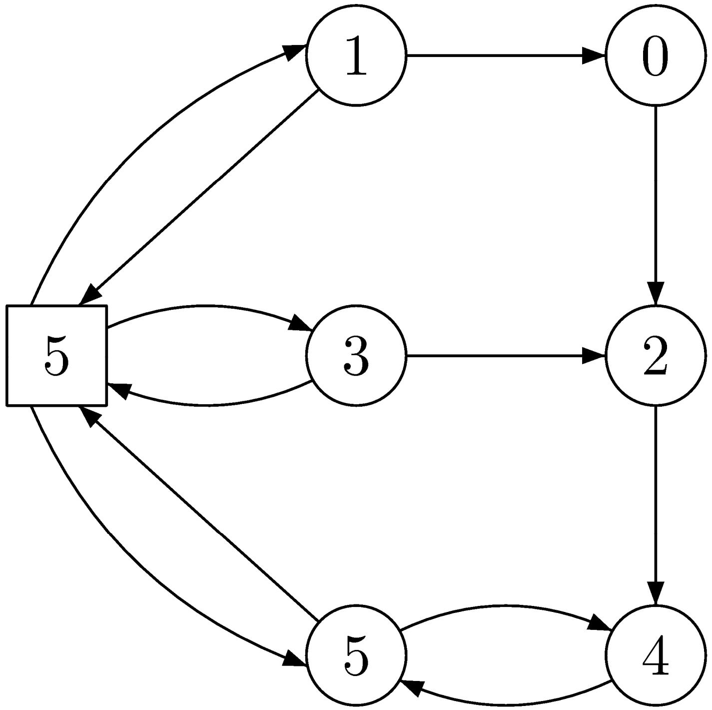
Play: $a, d, a, b, e, f, g, d, a, c, \ldots$

5, 5, 5, 1, 0, 2, 4, 5, 5, 3…

A play is **winning** if the least priority visited infinitely often is even.

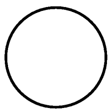
"Canonical representation of ω-regular specifications "
(e.g. all requests are eventually granted – G[r -> Fg])

Parity games



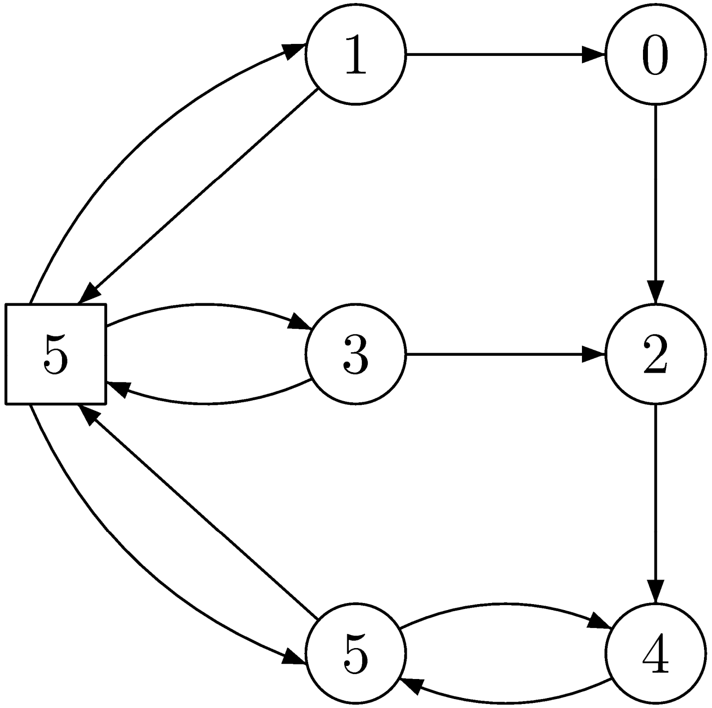
Decision problem:

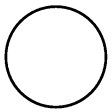
Decide if there exists a winning strategy of player 1 for parity condition.



Player 1 (good guy)

Parity games





Player 1 (good guy)

Decision problem:

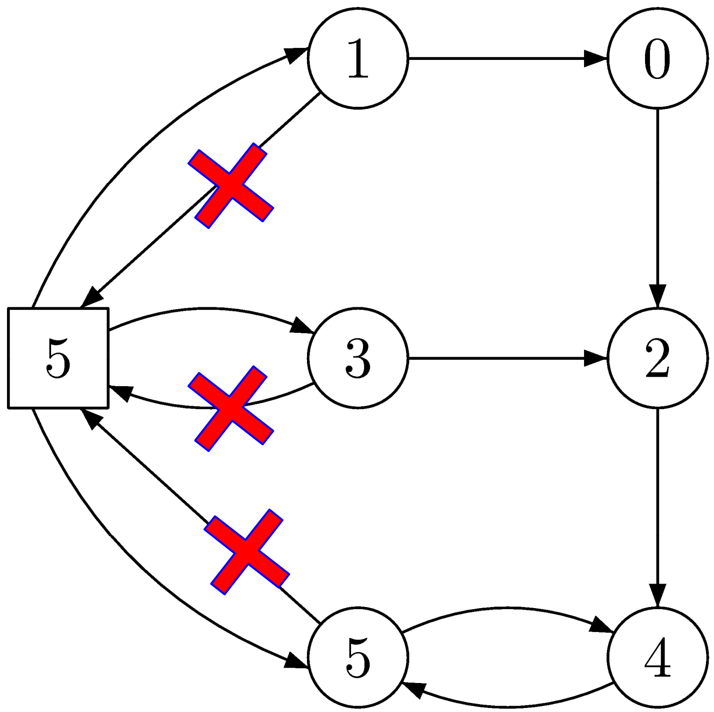
Decide if there exists a winning strategy of player 1 for parity condition.

For parity games, memoryless strategies suffice.

$$\sigma : Q_{\Box} \longrightarrow Q$$

$$\pi : Q_{\circ} \longrightarrow Q$$

Parity games

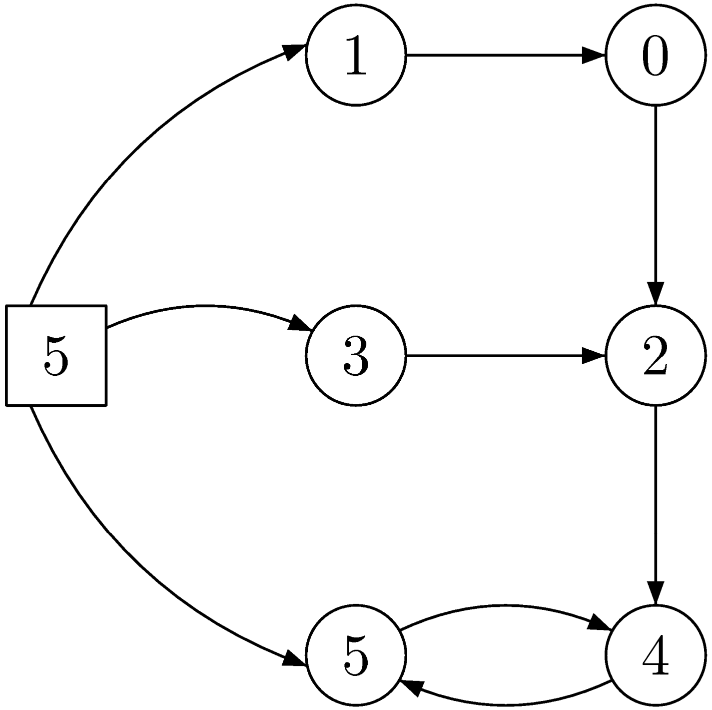


Decision problem:

Decide if there exists a winning strategy of player 1 for parity condition.

A memoryless strategy σ is winning if all cycles are **even** when σ is fixed.

Parity games



Decision problem:

Decide if th… …ning
stra… …parity
co…

NP ∩ coNP

See [EJ91]

A memoryless strategy σ is winning if all cycles are **even** when σ is fixed.

Summary

Energy games - “never exhaust the resource”

Parity game – “always eventually do something useful”

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy games	memoryless	memoryless	NP $\cap$ coNP
Parity games	memoryless	memoryless	NP $\cap$ coNP

Summary

Energy games - “never exhaust the resource”

Parity game – “always eventually do something useful”

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy games	memoryless	memoryless	NP $\cap$ coNP
Parity games	memoryless	memoryless	NP $\cap$ coNP
Energy parity games	?	?	?

Energy parity games

Energy parity games

Energy parity games: “never exhaust the resource”
and
“always eventually do something useful”

Energy parity games

Energy parity games: "never exhaust the resource"
and
"always eventually do something useful"

Decision problem:

Decide the existence of a finite initial credit sufficient to win.

Energy games – a story of cycles

Parity games – a story of cycles

Energy parity games - ?

A story of cycles

A good cycle ?

A story of cycles

A good cycle ?

A bad cycle ?

- energy is (strictly) negative
- or, least priority is odd

A story of cycles

A good cycle ?

A bad cycle ?

- energy is (strictly) negative
- or, least priority is odd

Player 1 looses energy parity game iff the opponent can force a **bad** cycle.

The opponent can force bad cycles **without memory**.

A story of cycles

In energy parity games, memoryless strategies are sufficient for Player 2.

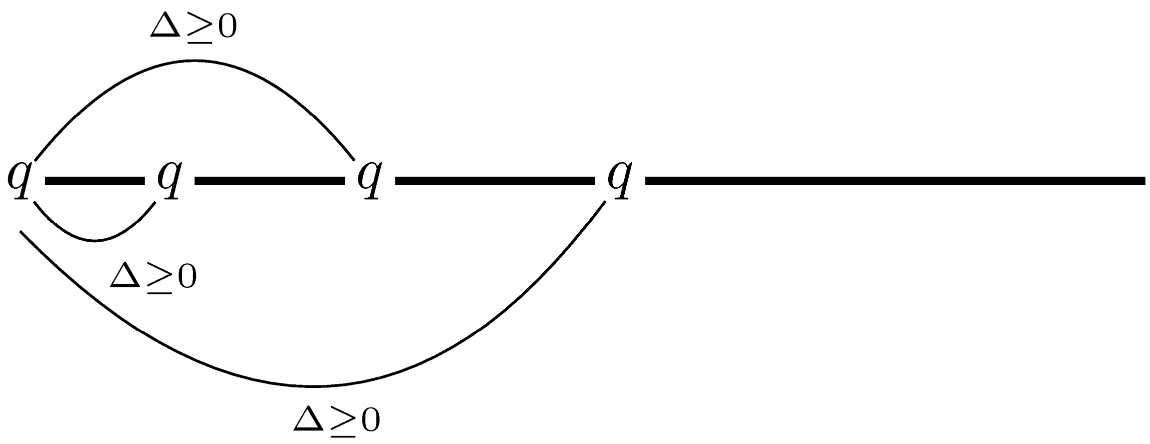
Proof

A story of cycles

In energy parity games, memoryless strategies are sufficient for Player 2.

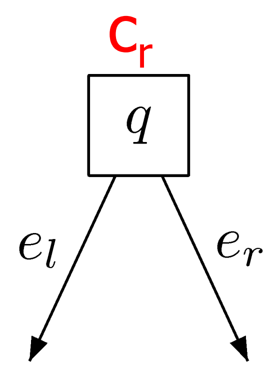
Proof

Preliminary fact: under optimal strategy, energy in q is always greater than on first visit to q.



A story of cycles

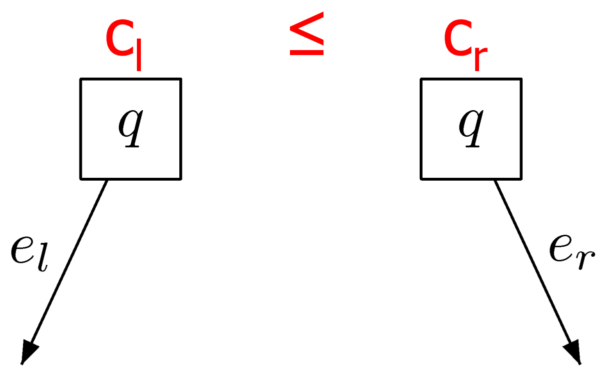
In energy parity games, memoryless strategies are sufficient for Player 2.



Proof

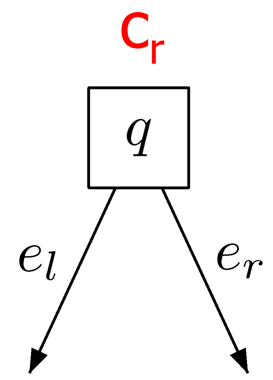
Assume player 1 looses with initial credit c_r,

then show that player 1 looses also against one of the "memoryless strategies in q":



A story of cycles

In energy parity games, memoryless strategies are sufficient for Player 2.



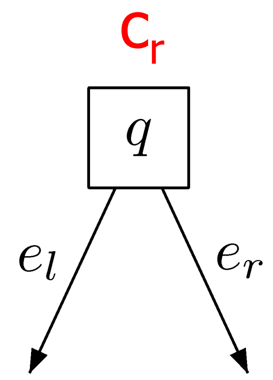
Proof

Fix winning strategy of Player 2,

all outcomes are loosing for Player 1:

A story of cycles

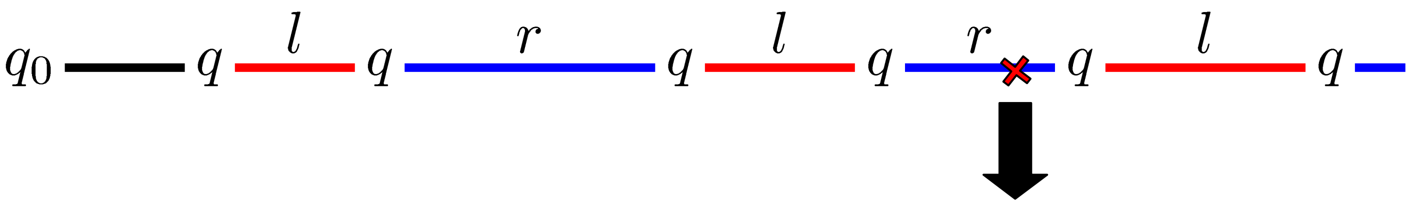
In energy parity games, memoryless strategies are sufficient for Player 2.



Proof

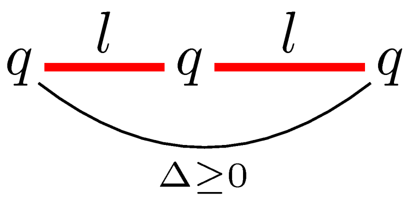
Fix winning strategy of Player 2,

all outcomes are loosing for Player 1:



Energy < 0 ?

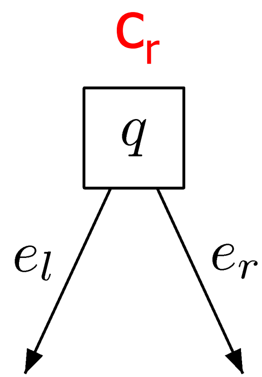
Then also in 

since 

A story of cycles

In energy parity games, memoryless strategies are sufficient for Player 2.

Proof

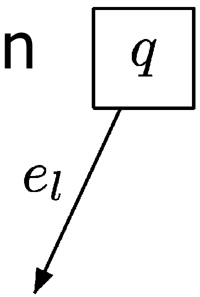
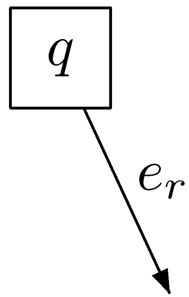


Fix winning strategy of Player 2,

all outcomes are loosing for Player 1:

$$q_0 \;\text{——}\; q \overset{l}{\text{——}} q \overset{r}{\text{————}} q \overset{l}{\text{——}} q \overset{r}{\text{——}} q \overset{l}{\text{———}} q \;\text{—}$$

Least ∞-visited priority is odd ?

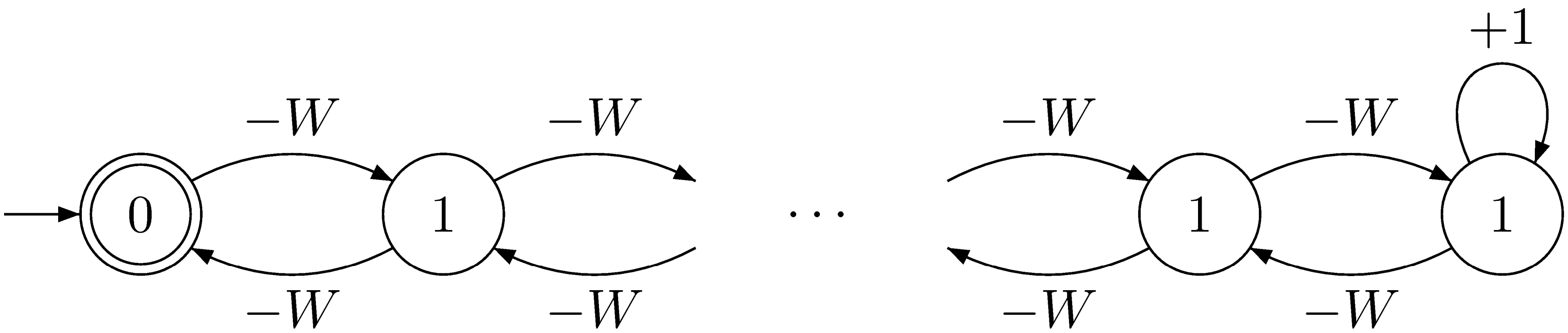
Then also in  or 

Complexity

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy games	memoryless	memoryless	NP $\cap$ coNP
Parity games	memoryless	memoryless	NP $\cap$ coNP
Energy parity games		memoryless	coNP

A story of cycles

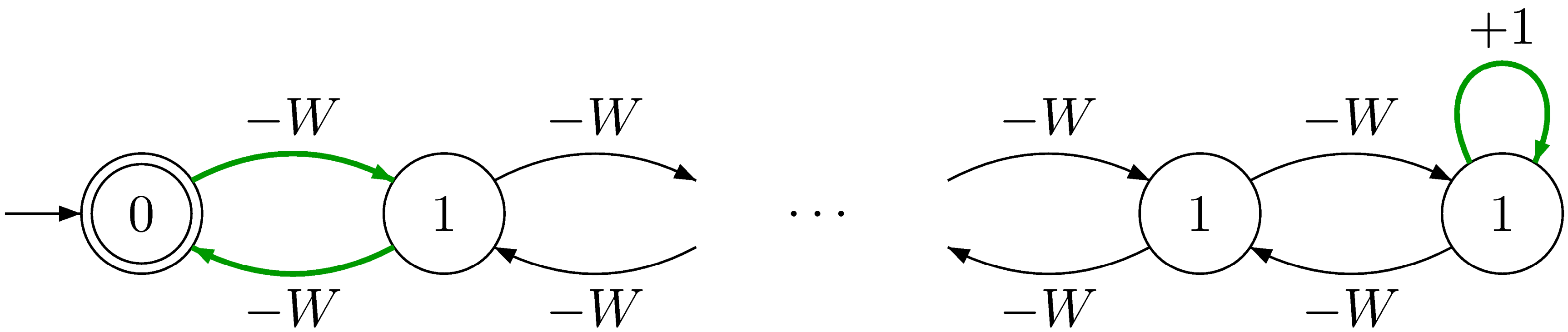
A good cycle ?



A one-player energy Büchi game

A story of cycles

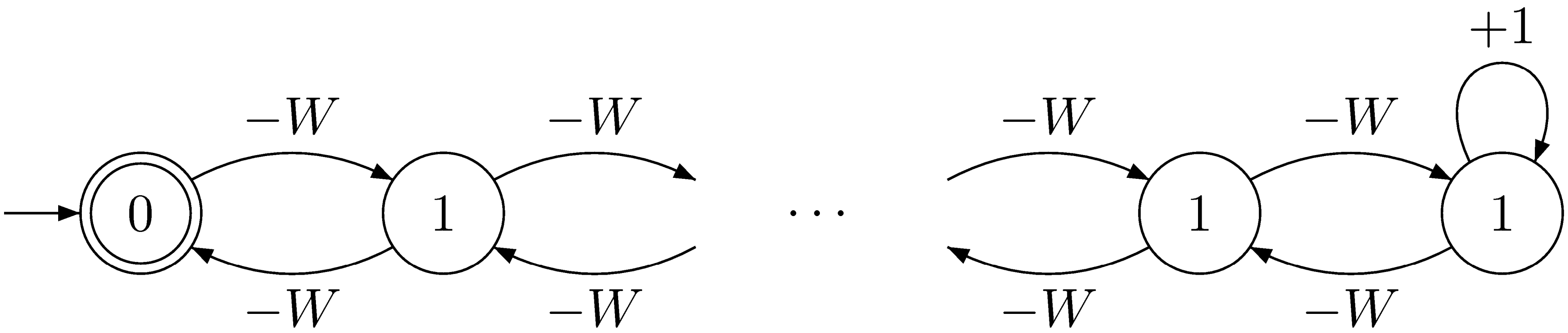
A good cycle ?



good for parity

good for energy

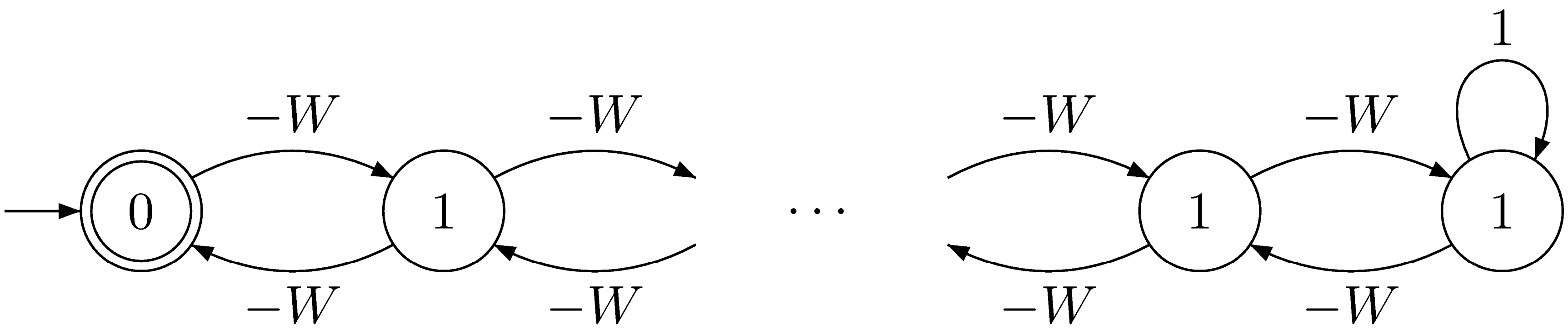
A story of cycles



Winning strategy:

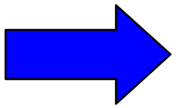
1. reach and repeat positive cycle to increase energy;
2. visit priority 0;
3. goto 1;

A story of cycles



Winning strategy:

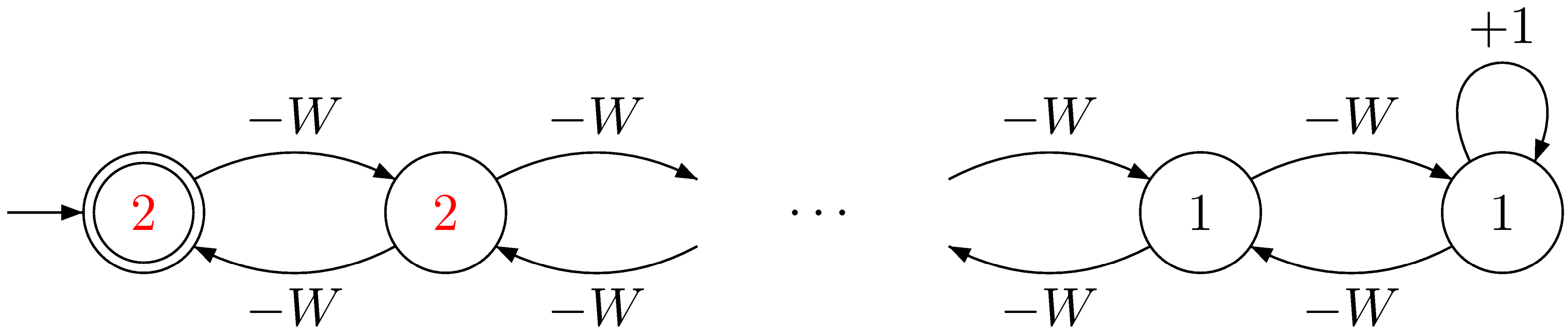
1. reach and repeat O(2nW) times the positive cycle;
2. visit priority 0;
3. goto 1;

 requires exponential memory !

Complexity

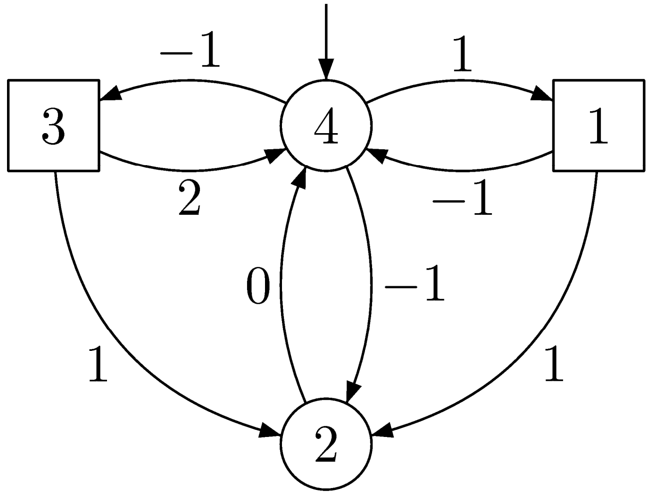
	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy games	memoryless	memoryless	NP $\cap$ coNP
Parity games	memoryless	memoryless	NP $\cap$ coNP
Energy parity games	exponential	memoryless	NP $\cap$ coNP

Note

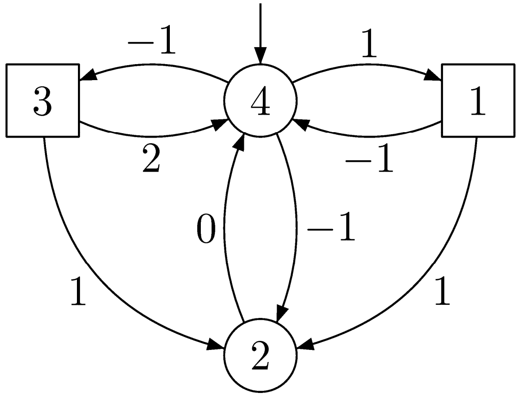


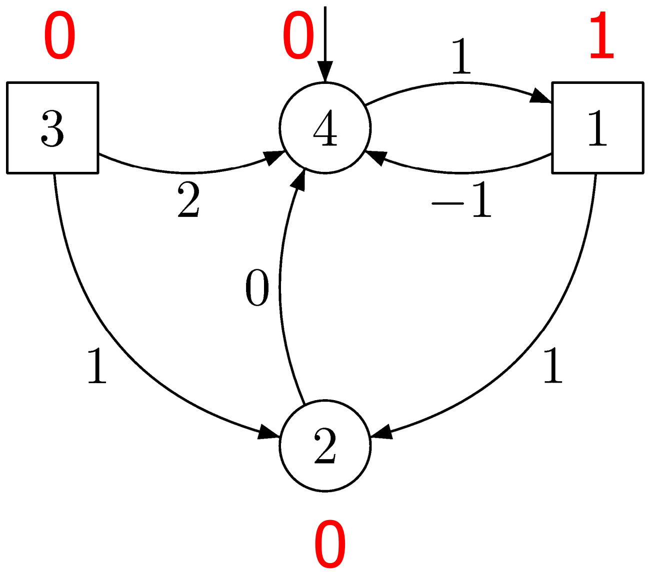
Energy-winning and parity-winning ≠ EnergyParity-winning

Structure of strategies



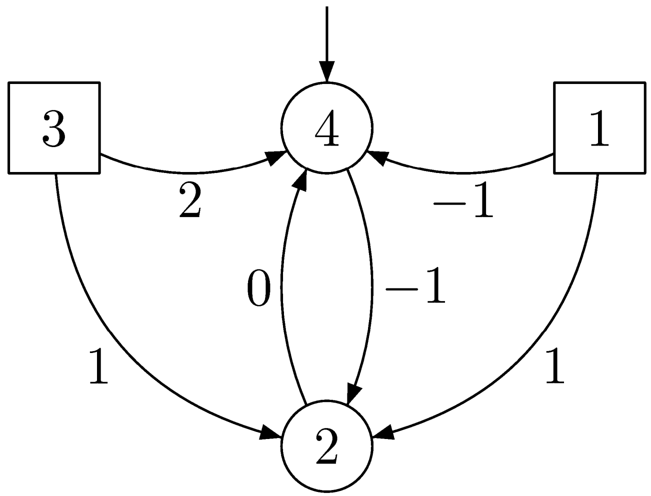
Structure of strategies





(optimal) energy-winning

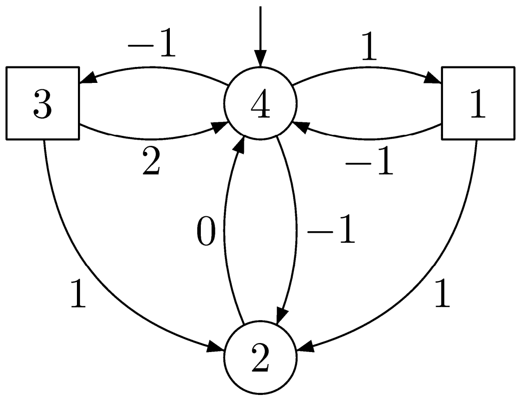
≥0 & min.pr. 1

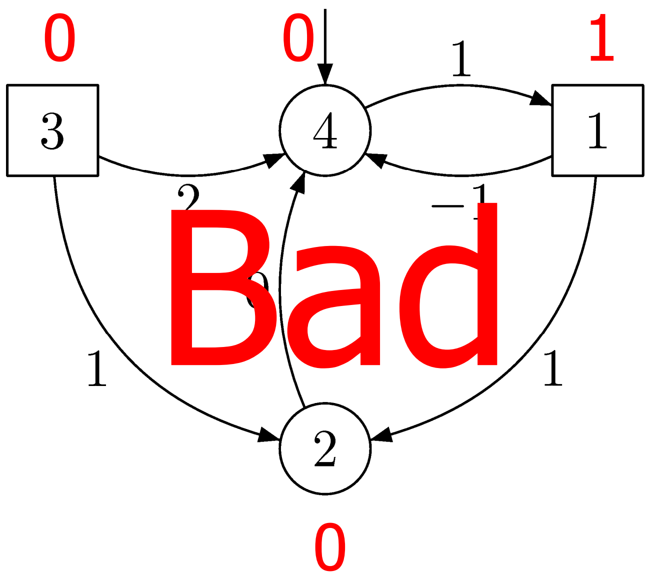


parity-winning

<0 & min.pr. 2

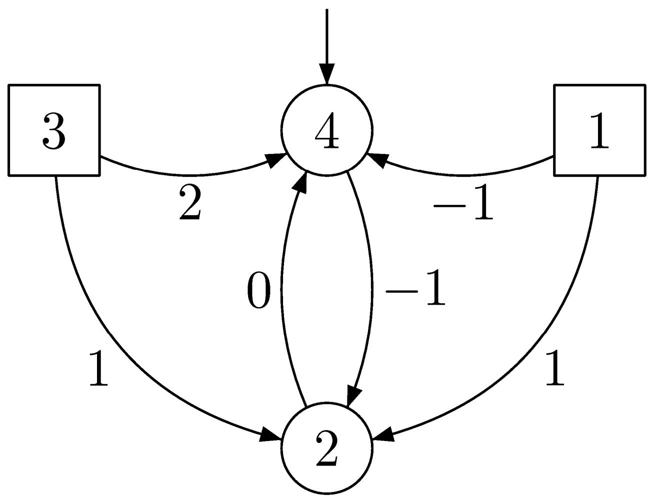
Structure of strategies





(optimal) energy-winning

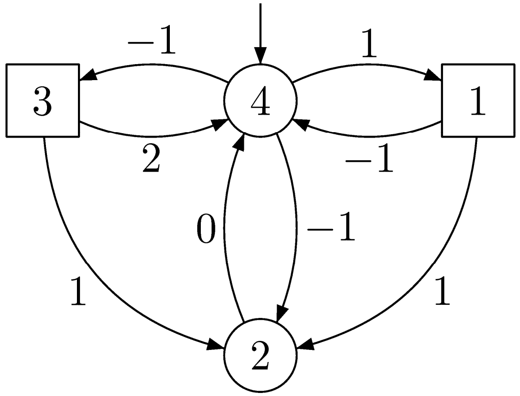
≥0 & min.pr. 1

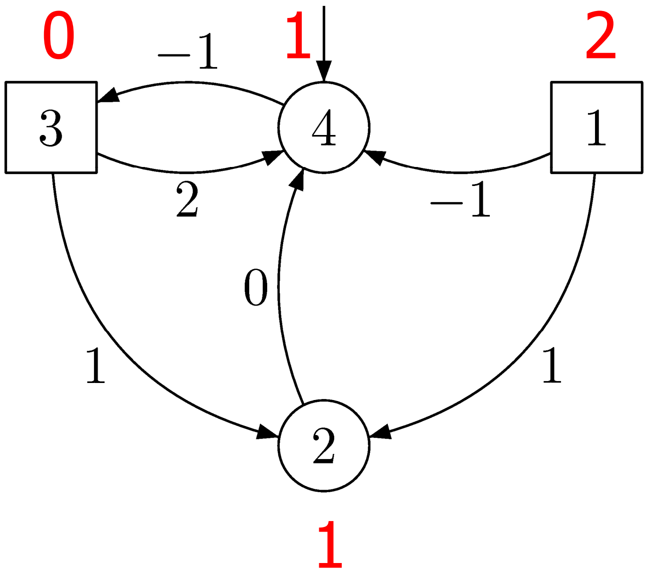


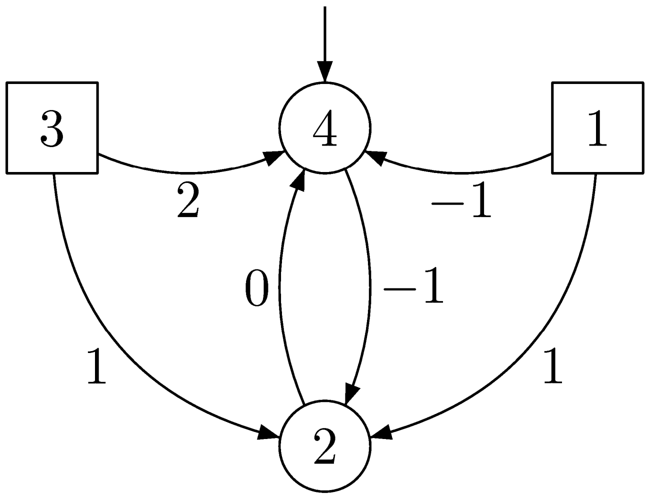
parity-winning

<0 & min.pr. 2

Structure of strategies







energy-winning
≥0 & least priority is even if =0

parity-winning
<0 & min.pr. 2

Good-for-energy strategies

A **good-for-energy** strategy is a winning strategy
in the following cycle-forming game:

Cycle-forming game: play the game until a cycle is formed

Player 1 wins if energy of the cycle is positive,
or energy is 0 and least priority is even.

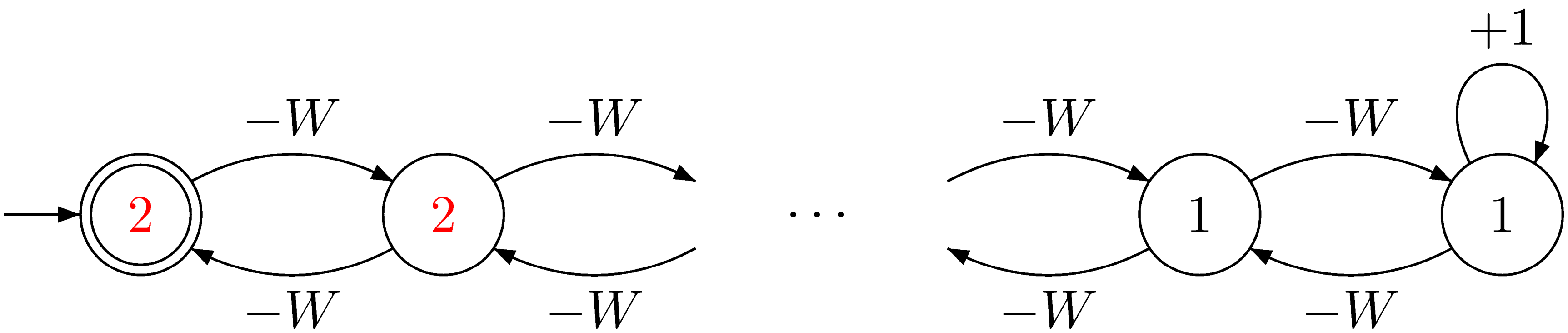
If Player wins in an energy parity game, then a **memoryless** good-for-energy strategy exists.

(not iff)

Good-for-energy strategies

Good-for-energy and parity-winning strategies are necessary to win…

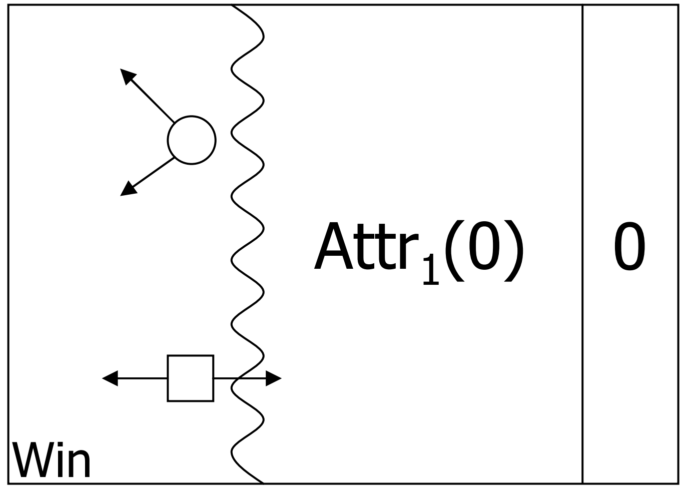
Winning strategy = alternate good-for-energy strategy and parity-winning strategy ?



good-for-energy and parity-winning ≠ EnergyParity-winning

Structure of strategies

- If least priority is 0



1. Play good-for-energy.
   Either energy stabilizes and then least priority is even,
   or energy (strictly) increases.

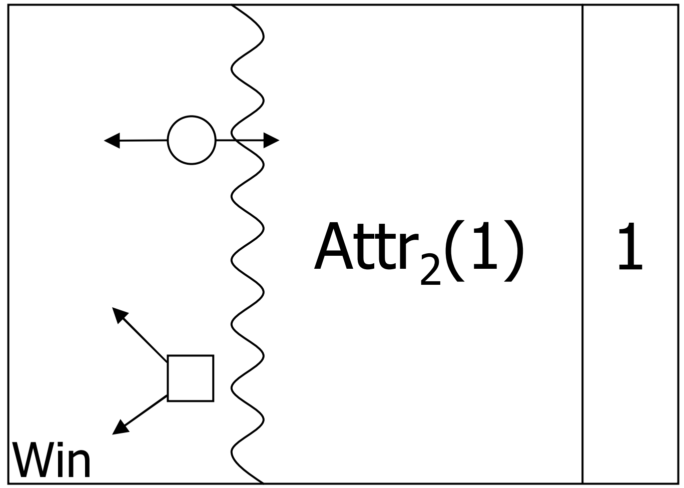
2. When energy is high enough (+2nW):

2a. If (and while) game is in Q \ Attr(0), play a winning strategy in subgame defined by Q \ Attr(0).

2b. Whenever game is in Attr(0), reach 0 and start over.

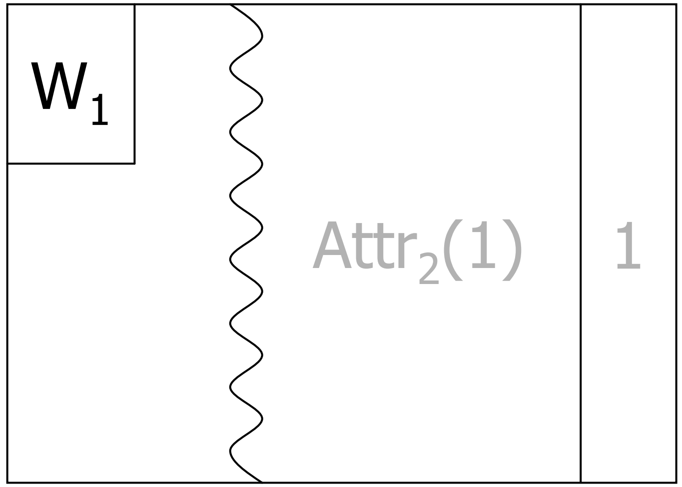
Structure of strategies

- If least priority is 1



Structure of strategies

- If least priority is 1



1. Game can be partitionned into winning regions and their attractor.
2. Winning strategy combines subgame winning stragies and reachability strategies.

Structure of strategies

- If least priority is 1

W_1		$Attr_1(W_1)$	
W_2		$Attr_1(W_2)$ $Attr_2(1)$	1
W_3		$Attr_1(W_3)$	

1. Game can be partitionned into winning regions and their attractor.
2. Winning strategy combines subgame winning stragies and reachability strategies.

Corollary: memoryless strategies are sufficient in coBüchi energy games.

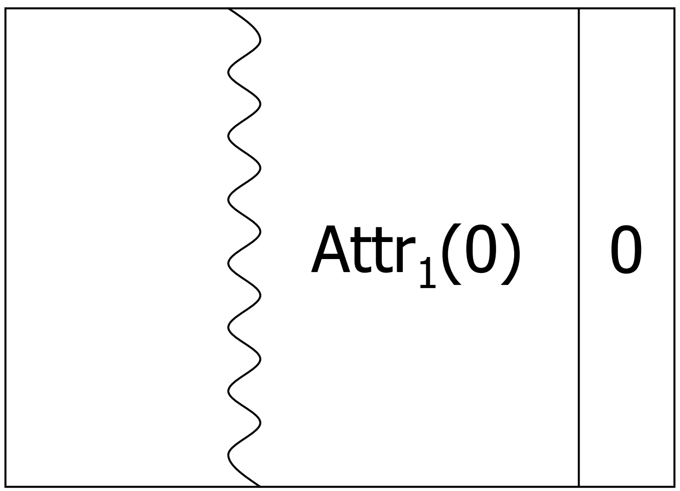
An NP solution

Assume NP-algorithm for d-1 priorities

least priority 0

NP-algorithm for d priorities:

- guess the winning set and good-for-energy strategy
- compute 0-attractor, and solve subgame in NP



$C_d(n) \leq p(n) + C_{d-1}(n)$

An NP solution

Assume NP-algorithm for d-1 priorities

least priority 1

NP-algorithm for d priorities:

- guess the winning set and partition
- compute 1-attractor, and solve subgames in NP

W_1		$Attr_1(W_1)$	
W_2		$Attr_1(W_2)$	1
W_3		$Attr_1(W_3)$	

$$C_d(n) \leq p(n) + C_{d-1}(n_1) + \dots + C_{d-1}(n_k)$$
$$\leq p(n) + C_{d-1}(n_1 + \dots + n_k)$$
$$\leq p(n) + C_{d-1}(n-1)$$

Complexity

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy games	memoryless	memoryless	NP $\cap$ coNP
Parity games	memoryless	memoryless	NP $\cap$ coNP
Energy parity games	exponential	memoryless	NP $\cap$ coNP

Algorithm

Algorithm for solving energy parity games ?

Determine good-for-energy winning states – a story of cycles

Good-for-energy:

All cycles are either >0, or $=0$ and even

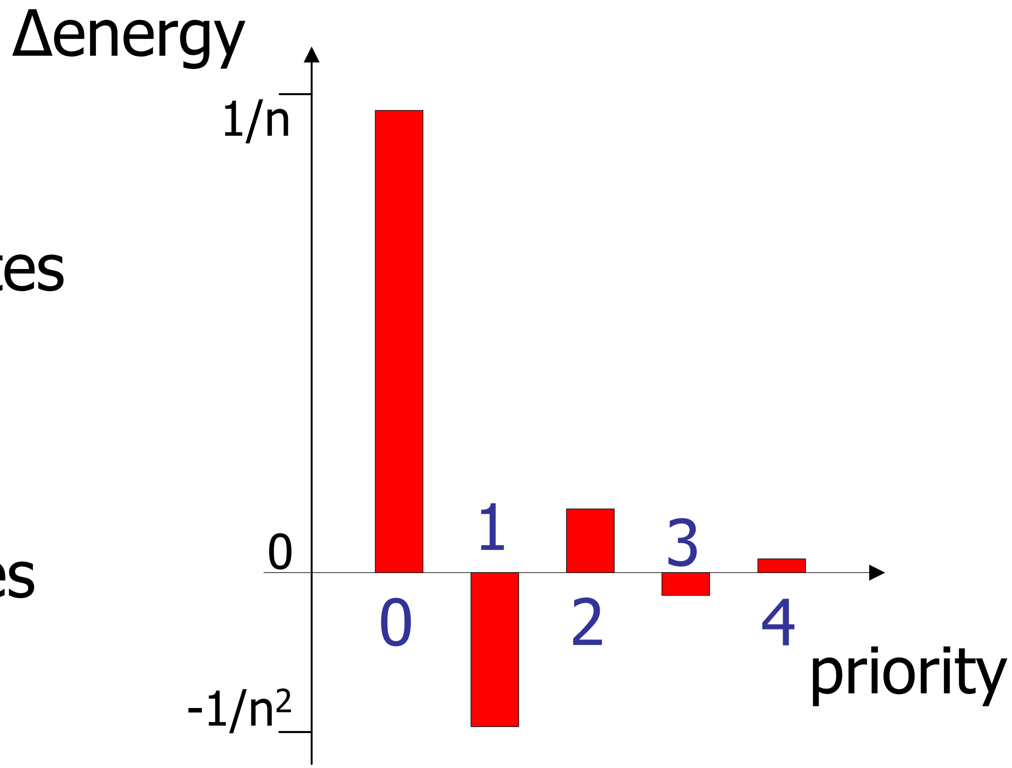
Reduction to (pure) energy games with modified weights:

cycles $=0$ and even $\rightarrow$ >0
cycles $=0$ and odd $\rightarrow$ <0
other cycles remain >0 or <0

Algorithm

Positive increment for transitions from even states

Negative increment for transitions from odd states



Reduction to energy games with modified weights:

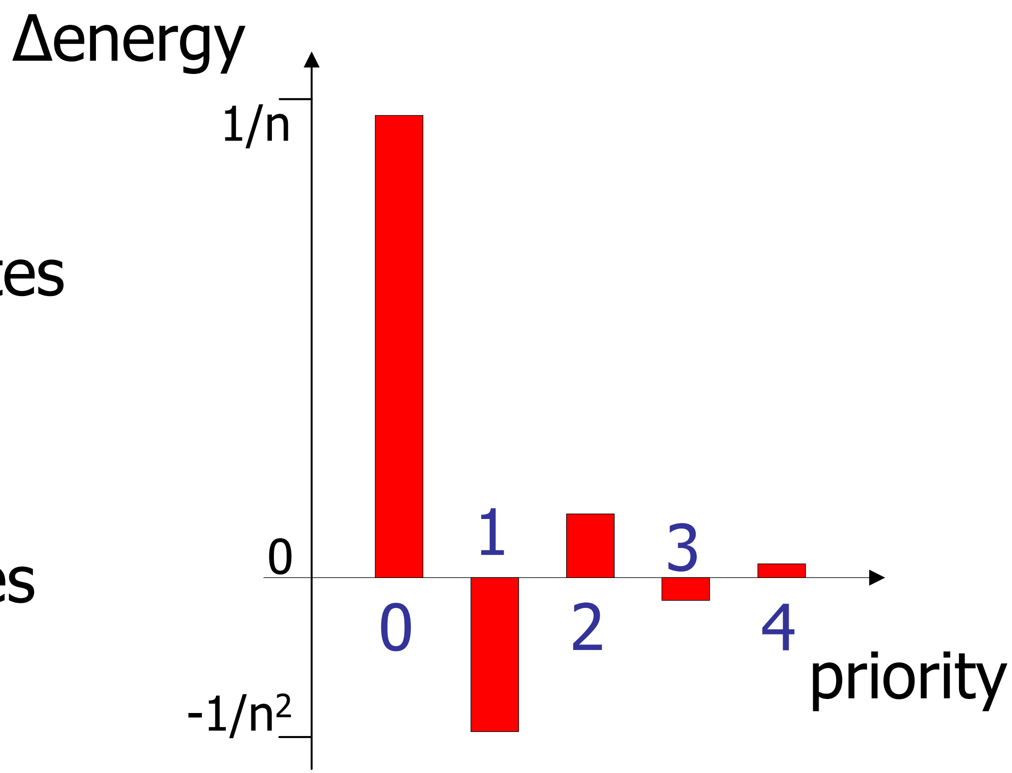
cycles =0 and even → >0
cycles =0 and odd → <0
other cycles remain >0 or <0

Algorithm

Positive increment for transitions from even states

Negative increment for transitions from odd states

$$\Delta(q) \approx \frac{(-1)^k}{n^{k+1}}$$



Increment is **exponential** (in nb. of priorities)

Algorithm

Algorithm for solving energy parity games ?

Determine good-for-energy winning states

by solving modified-energy game in $O(E.Q^{d+2}.W)$

Recursive fixpoint algorithm, flavour of McNaughton-Zielonka

Note: a reduction to parity games (making energy explicit) would give complexity $O(E.(Q^2W)^d)$.

Relationship with mean-payoff parity games

Two-player games on graphs

Qualitative

Parity games

Quantitative

Energy games

Mean-payoff games

Mixed qualitative-quantitative

Energy parity games

Mean-payoff parity games

Mean-payoff

Mean-payoff value of a play = limit-average of the visited weights

$$\lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} w_i$$

Optimal mean-payoff value can be achieved with a **memoryless** strategy.

Decision problem:

> Given a rational threshold ν, decide if there exists a strategy for player 1 to ensure mean-payoff value at least ν.

Mean-payoff

Memoryless strategy σ ensures nonnegative mean-payoff value

iff

all cycles are nonnegative in G_σ

iff

memoryless strategy σ is winning in energy game

⇒ Mean-payoff games with threshold 0 are equivalent to energy games.

Mean-payoff parity

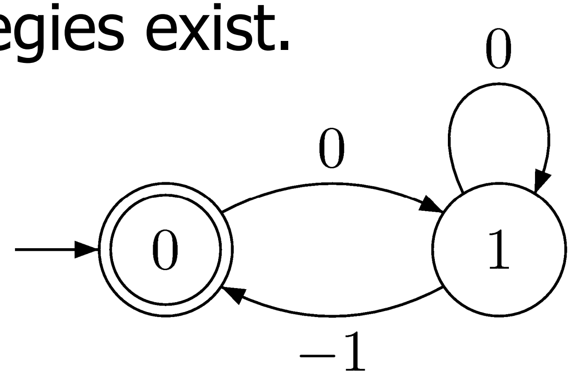
Mean-payoff parity games [CHJ05]

Objective:

- satisfy parity condition
- maximize mean-payoff

Infinite memory may be necessary !

However, finite-memory ε-optimal strategies exist.



Mean-payoff parity

Mean-payoff parity games are polynomially equivalent to energy parity games.

Reduction idea:

from MPP game, construct EP game
by incrementing all weights by $\varepsilon=1/(n+1)$

If Player 1 wins MPP ≥ 0, then finite-memory ε-optimal strategy exists, which is winning in EP game.

If Player 1 wins EP game, then he wins MPP $\geq -\varepsilon$ and then also MPP ≥ 0 since the value in MPP has denomiantor $\leq n$.

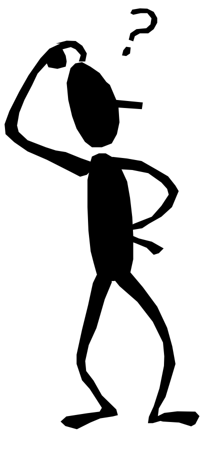
Complexity

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy	memoryless	memoryless	NP $\cap$ coNP
Parity	memoryless	memoryless	NP $\cap$ coNP
Energy parity	exponential	memoryless	NP $\cap$ coNP
Mean-payoff parity	infinite	memoryless	NP $\cap$ coNP

By-product: a conceptually simple algorithm for mean-payoff parity games.

The end

Thank you !



Questions ?

References

[CdAHS03] A.Chakrabarti, L. de Alfaro, T.A. Henzinger, and M. Stoelinga. **Resource interfaces**, Proc. of EMSOFT: Embedded Software, LNCS 2855, Springer, pp.117-133, 2003

[EM79] A. Ehrenfeucht, and J. Mycielski, **Positional Strategies for Mean-Payoff Games**, International Journal of Game Theory, vol. 8, pp. 109-113, 1979

[BFL+08] P. Bouyer, U. Fahrenberg, K.G. Larsen, N. Markey, and J. Srba, **Infinite Runs in Weighted Timed Automata with Energy Constraints**, Proc. of FORMATS: Formal Modeling and Analysis of Timed Systems, LNCS 5215, Springer, pp. 33-47, 2008

[CHJ05] K. Chatterjee, T.A. Henzinger, M. Jurdzinski. Mean-payoff Parity Games, Proc. of LICS: Logic in Computer Science, IEEE, pp. 178-187, 2005.

Energy parity

Algorithm 1: SolveEnergyParityGame

Input : An energy parity game $\langle G, p, w \rangle$ with state space Q.
Output: The set of winning states in $\langle G, p, w \rangle$ for player 1.
begin
  if $Q = \varnothing$ **then return** $\varnothing$;
  Let k^* be the minimal priority in G. Assume w.l.o.g. that $k^* \in \{0, 1\}$;
  Let G_0 be the game G ;
  $i \leftarrow 0$;
  if $k^* = 0$ **then**
    $A_0 \leftarrow Q$ `/* over-approximation of Player-1 winning states */` ;
    repeat
      $A'_i \leftarrow$ SolveEnergyGame(G_i, w') (where w' is defined in Lemma **??**) ;
      $X_i \leftarrow Attr_1(A'_i \cap p^{-1}(0))$;
      Let G'_i be the subgraph of G_i induced by $A'_i \setminus X_i$;
      $Z_i \leftarrow (A'_i \setminus X_i) \setminus$ SolveEnergyParityGame(G'_i, p, w) ;
      $A_{i+1} \leftarrow A'_i \setminus Attr_2(Z_i)$;
      Let G_{i+1} be the subgraph of G_i induced by A_{i+1} ;
      $i \leftarrow i + 1$;
    until $A_i = A_{i-1}$;
    return A_i;
  if $k^* = 1$ **then**
    $B_0 \leftarrow Q$ `/* over-approximation of Player-2 winning states */` ;
    repeat
      $Y_i \leftarrow Attr_2(B_i \cap p^{-1}(1))$;
      Let G_{i+1} be the subgraph of G_i induced by $B_i \setminus Y_i$;
      $B_{i+1} \leftarrow B_i \setminus Attr_1($SolveEnergyParityGame$(G_{i+1}, p, w))$;
      $i \leftarrow i + 1$;
    until $B_i = B_{i-1}$;
    return $Q \setminus B_i$;
end

Mean-payoff parity

Algorithm 1 ComputeLeastValueClass

Input: a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$ such that $p^{-1}(0) \neq \emptyset$
and the game is parity winning for player 1.
Output: a nonempty 1-closed subset of $\mathcal{LV}$, and $MP_1(v)$ for all $v \in \mathcal{LV}$.
1. $F = Attr_1(p^{-1}(0), \mathcal{G})$.
2. $H = V \setminus F$ and $\mathcal{H} = \mathcal{G} \upharpoonright H$.
3. **MeanPayoffParitySolve**$(\mathcal{H})$ (Algorithm 3).
4. Construct the mean-payoff game $\widetilde{\mathcal{G}}$ as described in Subsection 3.1 and **Solve**
5. Let $\mathcal{LV}_{\widetilde{\mathcal{G}}}$ be the least value class in $\widetilde{\mathcal{G}}$ and $\widetilde{l}$ be the least value.
6. $\mathcal{LV} = \mathcal{LV}_{\widetilde{\mathcal{G}}} \cap V$, and $MP_1(v) = \widetilde{l}$ for all $v \in \mathcal{LV}$.
7. **return** $(\mathcal{LV}, \widetilde{l})$.

Algorithm 2 ComputeGreatestValueClass

Input: a mean-payoff parity game $\mathcal{MP} = (\mathcal{G}, p, r)$ such that $p^{-1}(0) = \emptyset$ and $p^{-1}(1) \neq \emptyset$,
and the game is parity winning for player 1.
Output: a nonempty
1. $F = Attr_2(p^{-1}(1$
2. $H = V \setminus F$ and $\mathcal{H}$
3. **MeanPayoffParit**
4. Let $\mathcal{GV}_{\mathcal{H}}$ be the gr
5. $\mathcal{GV} = \mathcal{GV}_{\mathcal{H}}$, and
6. **return** $(\mathcal{GV}, \hat{g})$.

Subroutine *SetValues*(J_i, j_i)

1. $g = \max\{ Val(w) : w \in W_0$ and $\exists v \in J_i \cap V_1.\ (v, w) \in E\}$.

2.1 **if** $g > j_i$ **then**

2.2 $T_1 = \{v \in J_i \cap V_1 : \exists w \in W_0.\ Val(w) = g$ and $(v, w) \in E\}$; and $W_0 = W_0 \cup UnivReach(T_1)$.

2.3 For every vertex $v \in UnivReach(T_1)$, set $Val(v) = g$.

2.4 **goto** Step 6.3. of **MeanPayoffParitySolve**.

3. $l = \min\{ Val(w) : w \in W_0$ and $\exists v \in J_i \cap V_2.\ (v, w) \in E\}$.

4.1 **if** $l < j_i$ **then**

4.2 $T_2 = \{v \in J_i \cap V_2 : \exists w \in W_0.\ Val(w) = l$ and $(v, w) \in E\}$; and $W_0 = W_0 \cup UnivReach(T_2)$.

4.3 For every vertex $v \in UnivReach(T_2)$, set $Val(v) = l$.

4.4 **goto** Step 6.3. of **MeanPayoffParitySolve**.

Algorithm 3 MeanPayoffParitySolve

Input: a mean-payoff party game $\mathcal{MP}$. **Output:** the value function MP_1.
1. Compute W_1 and W_2 by any algorithm for solving parity games.
2. For every vertex $v \in W_2$, set $Val(v) = -\infty$. 3. $W_0 = \emptyset$. 4. $\mathcal{G}_0 = \mathcal{G} \upharpoonright W_1$ and $V^0 = W_1$. 5. $i = 0$.
6. **repeat**

6.1. **while** $(p^{-1}(0) \cup p^{-1}(1)) \cap V_i = \emptyset$ **do** set $p = p - 2$. **end while**

6.2. Let (W_1^i, W_2^i) be the partition of the parity winning sets in $\mathcal{G}_i$.

6.2.a. **if** $W_2^i \neq \emptyset$ **then**

6.2.a.1. $g = \max\{ Val(w) : w \in W_0$ and $\exists v \in W_2^i \cap V_1.\ (v, w) \in E\}$.

6.2.a.2. $T_1 = \{v \in L_i \cap V_1 : \exists w \in W_0.\ Val(w) = g$ and $(v, w) \in E\}$.

6.2.a.3. $W_0 = W_0 \cup UnivReach(T_1)$.

6.2.a.4. For every vertex $v \in UnivReach(T_1)$, set $Val(v) = g$.

goto Step. 6.3.

6.2.b. **else**

6.2.b.1. **if** $p^{-1}(0) \cap V_i \neq \emptyset$ **then** let $(L_i, l_i) =$ **ComputeLeastValueClass**$(\mathcal{G}_i)$.

6.2.b.1.a. Subroutine *SetValues*(L_i, l_i)

6.2.b.1.b. $W_0 = W_0 \cup L_i$, and for every vertex $v \in L_i$, set $Val(v) = l_i$.

6.2.b.2. **else** $(G_i, g_i) =$ **ComputeGreatestValueClass**$(\mathcal{G}_i)$.

6.2.b.2.a. Subroutine *SetValues*(G_i, g_i)

6.2.b.2.b. $W_0 = W_0 \cup G_i$, and for every vertex $v \in G_i$, set $Val(v) = g_i$.

6.3. $V^{i+1} = V^i \setminus W_0$ and $\mathcal{G}_i = \mathcal{G} \upharpoonright V^i$.

6.4. $i = i + 1$.

until $V_i = \emptyset$ **(end repeat)**

7. $MP_1 = Val$.

Complexity

	Strategy		Algorithmic complexity
	Player 1	Player 2	
Energy	memoryless	memoryless	NP $\cap$ coNP
Parity	memoryless	memoryless	NP $\cap$ coNP
Energy parity	exponential	memoryless	NP $\cap$ coNP
Mean-payoff	memoryless	memoryless	NP $\cap$ coNP
Mean-payoff parity	infinite	memoryless	NP $\cap$ coNP