The Multiple Dimensions of Mean-Payoff Games

Laurent Doyen
CNRS & LSV, ENS Paris-Saclay

RP 2017

About

Basics about mean-payoff games

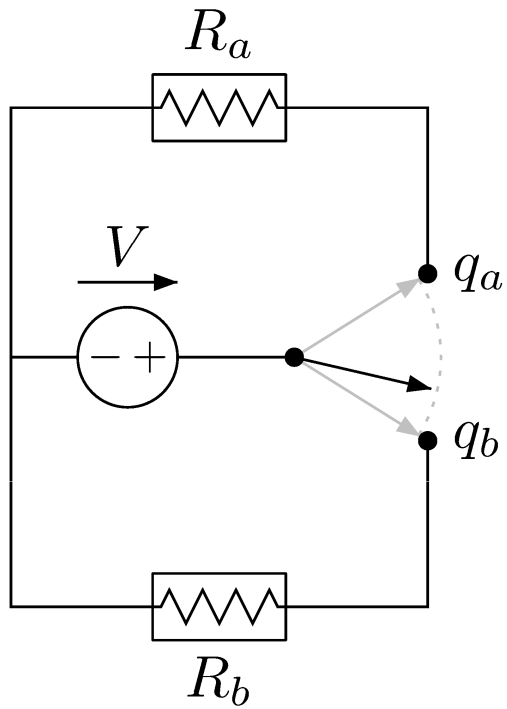
- Algorithms & Complexity
- Strategy complexity – Memory

Focus

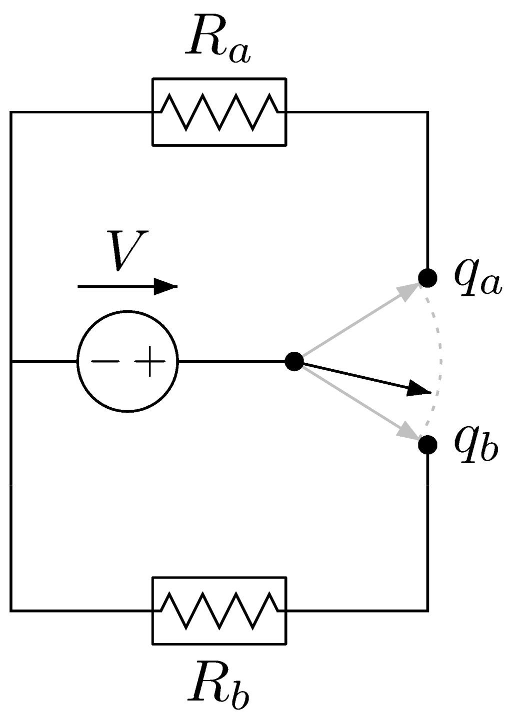
- Equivalent game forms
- Techniques for memoryless proofs

Mean-Payoff

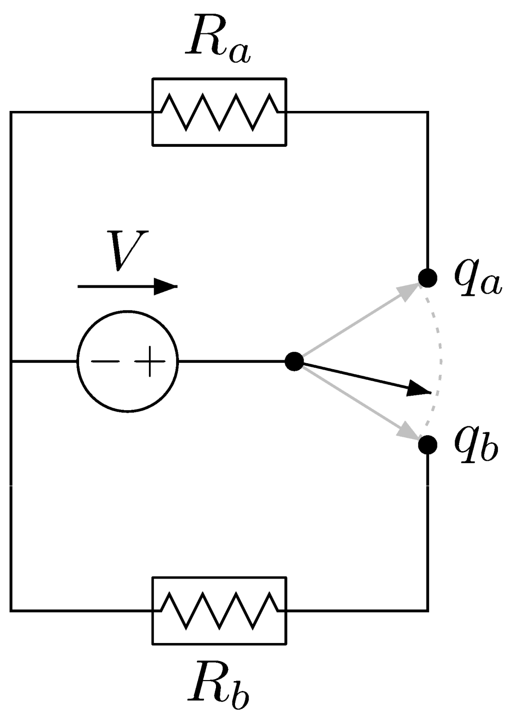
Mean-Payoff

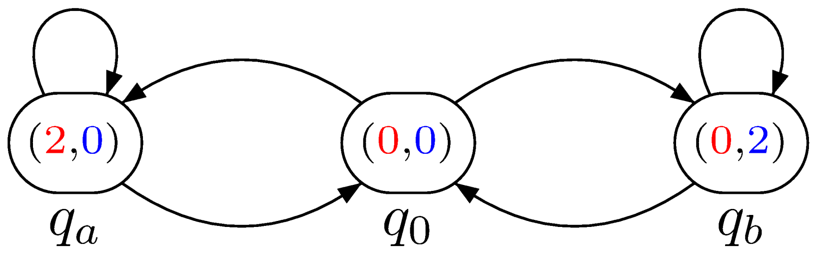


Mean-Payoff

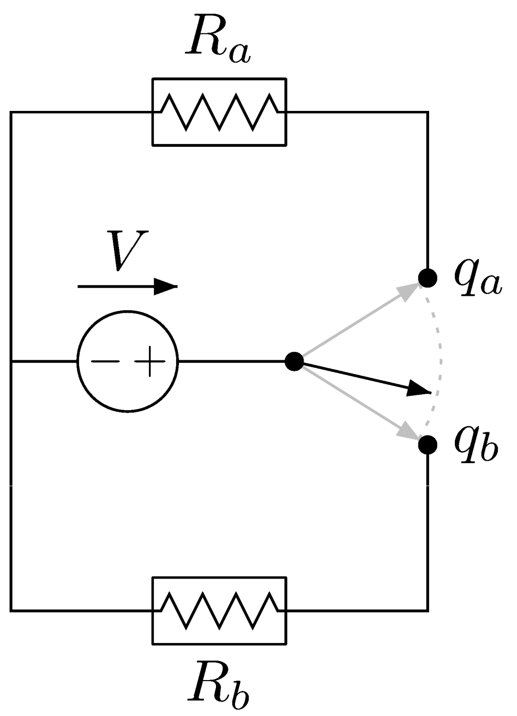


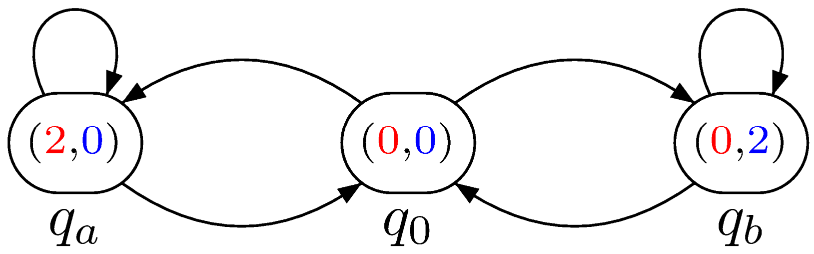
Mean-Payoff





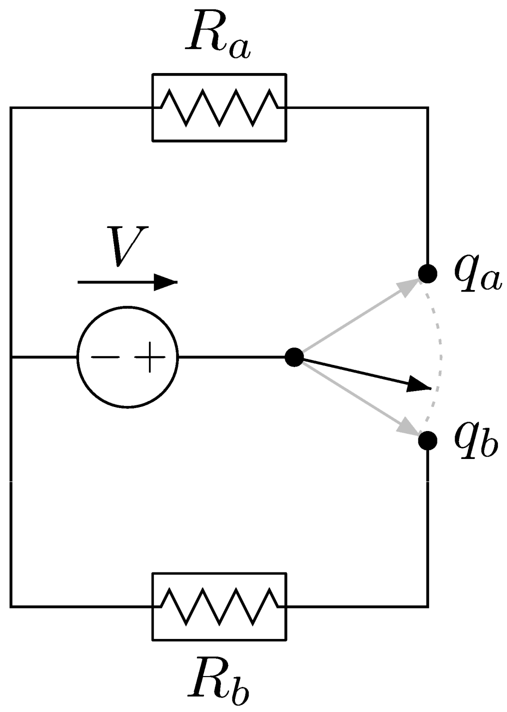
Mean-Payoff

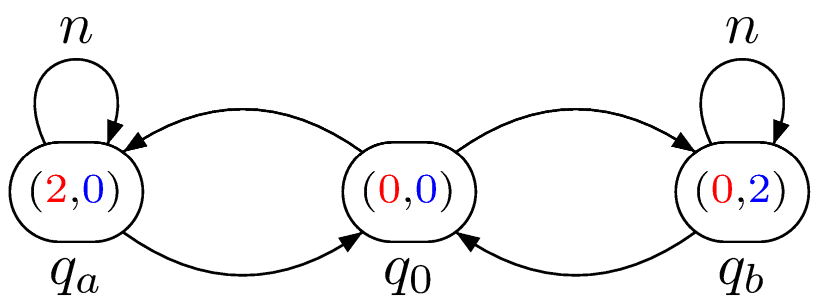




Switching policy to get average power (1,1) ?

Mean-Payoff

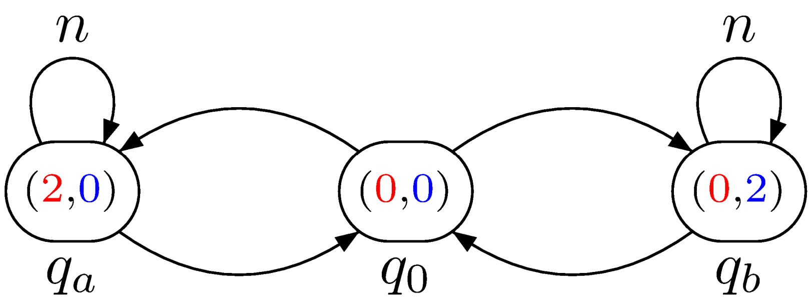




Switching policy to get average power (1,1) ?

$$\frac{n\cdot(2,0) + n\cdot(0,2) + 2\cdot(0,0)}{2n+2} = \frac{(2n,2n)}{2n+2} \to (1,1)$$

Mean-Payoff



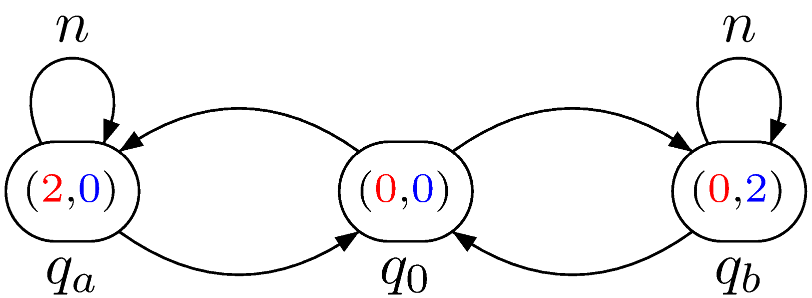
weight : $Q \to \mathbb{Z}^d$

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

$$\mathsf{MP}(q_0 q_1 \dots) = \lim_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$

$$1 \leq k \leq d$$

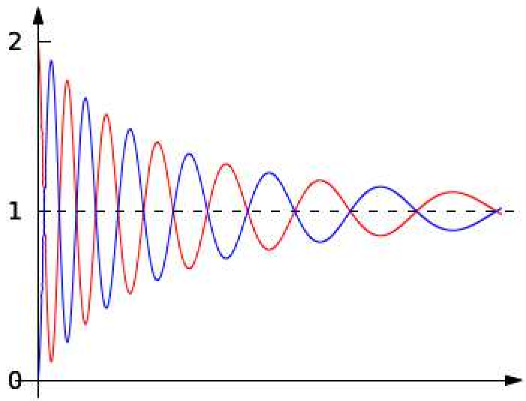
Mean-Payoff



weight : $Q \to \mathbb{Z}^d$

Switching policy

- Infinite memory: (1,1)   vanishing frequency in q_0

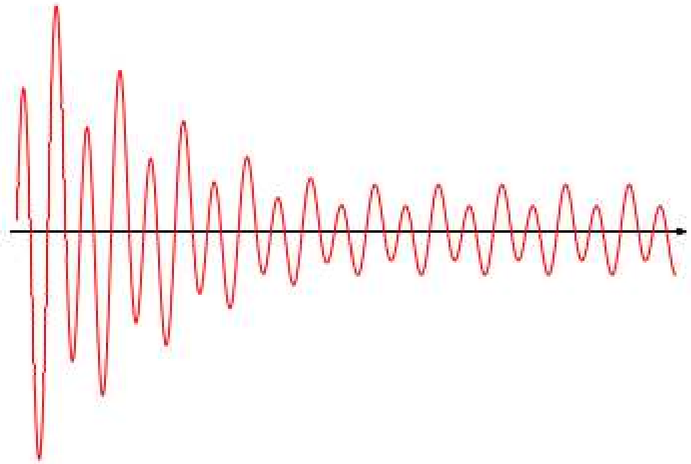


$$\mathsf{MP}(q_0 q_1 \dots) = \lim_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$

$$1 \le k \le d$$

Mean-Payoff

$$\mathsf{MP} = \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$

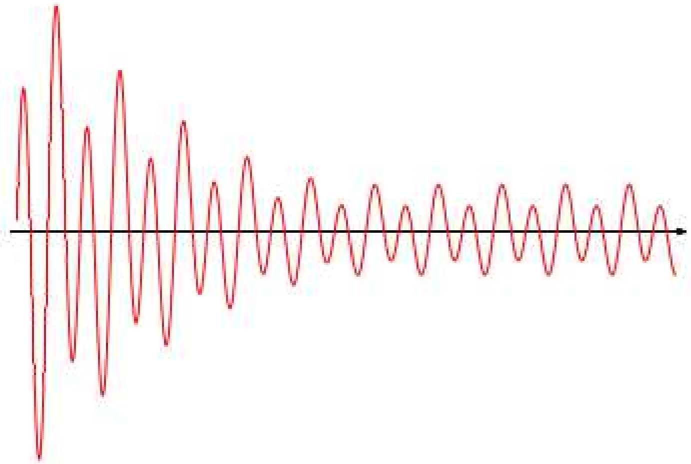


limit ?

Mean-Payoff

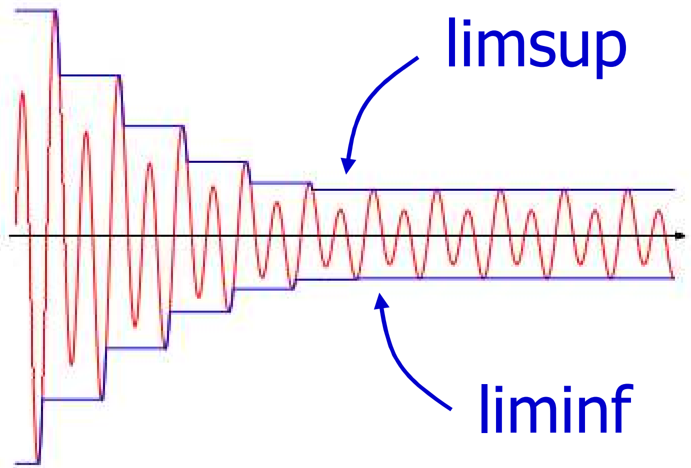
$$\mathsf{MP} = \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$

limit ?



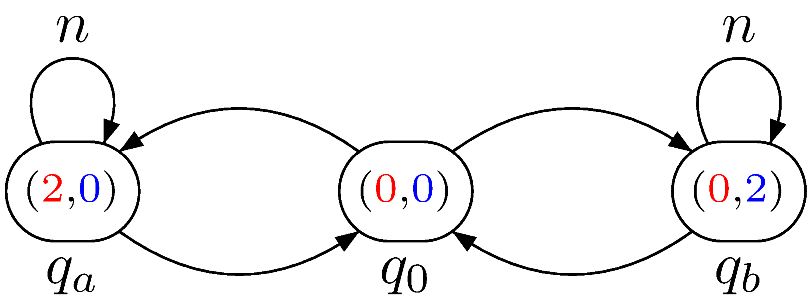
$$\overline{\mathsf{MP}} = \limsup_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$

$$\underline{\mathsf{MP}} = \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{weight}_k(q_i)$$



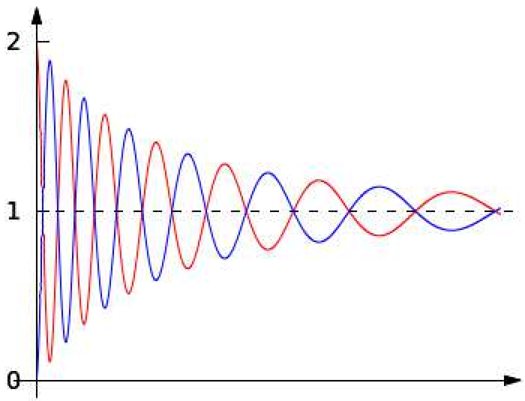
Mean-payoff is prefix-independent

Mean-Payoff

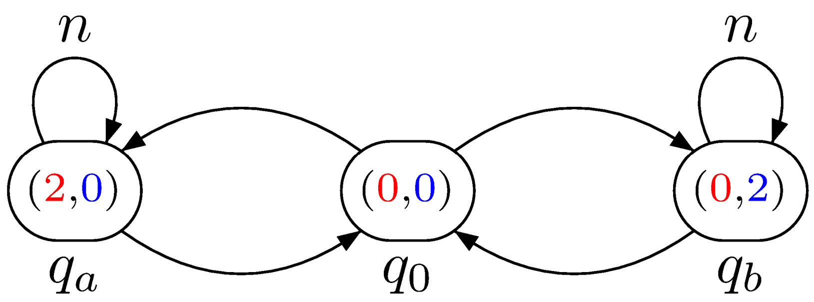


Switching policy

- Infinite memory: (1,1) for liminf

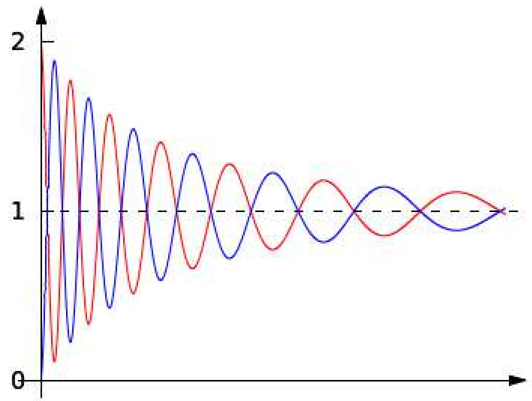


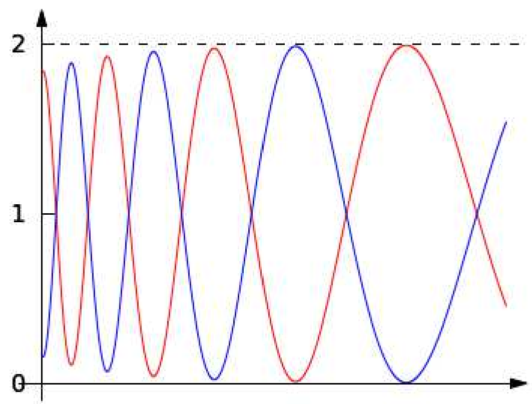
Mean-Payoff



Switching policy

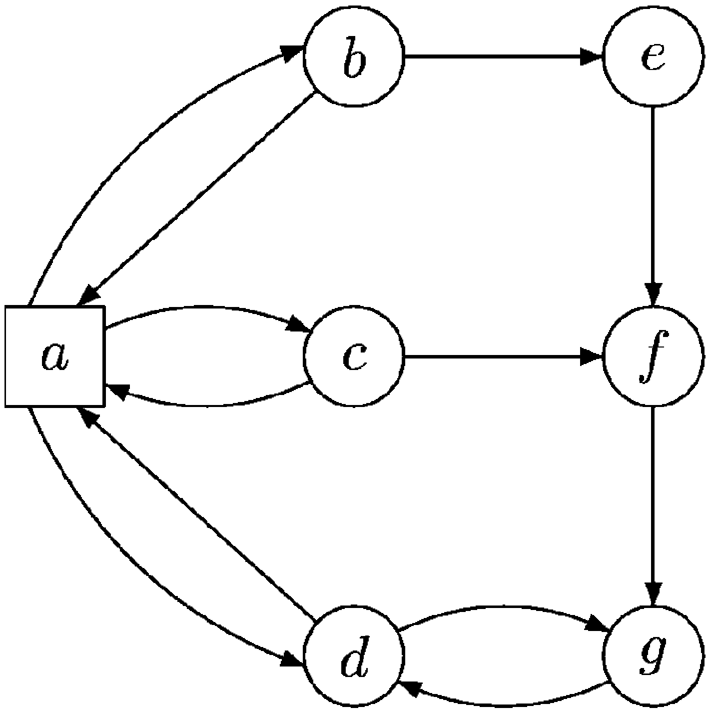
- Infinite memory: (1,1) for liminf & (2,2) for limsup





Games

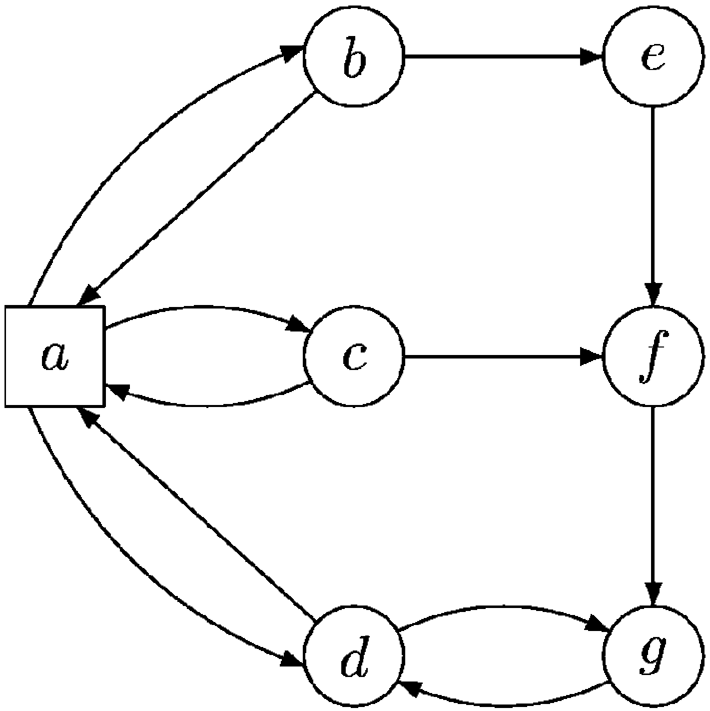
Two-player games

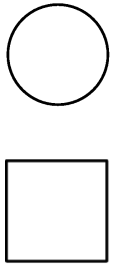


$$G = (Q, E)$$

$$\begin{cases} Q = Q_\circ \cup Q_\Box \\ E \subseteq Q \times Q \end{cases}$$

Two-player games



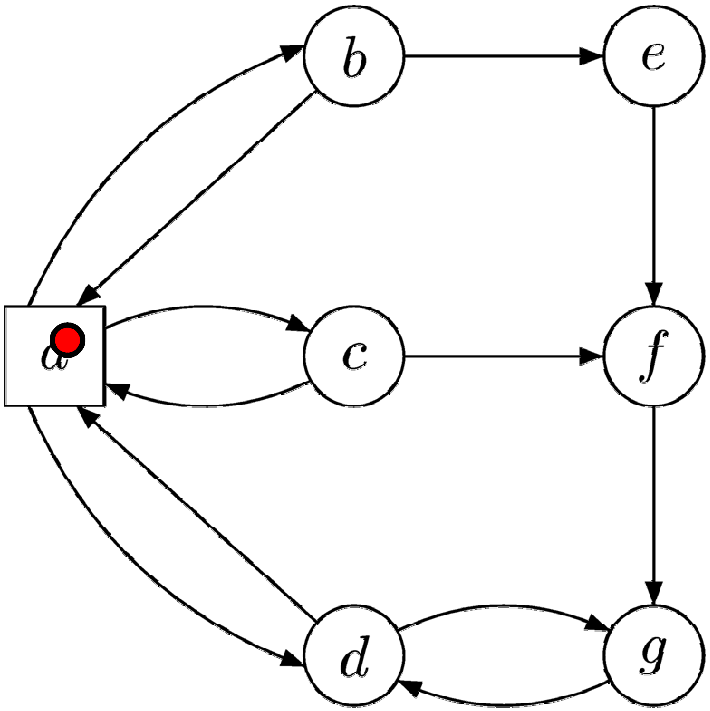


- Turn-based
- Infinite duration

$$G = (Q, E)$$

$$\begin{cases} Q = Q_\circ \cup Q_\Box \\ E \subseteq Q \times Q \end{cases}$$

Two-player games



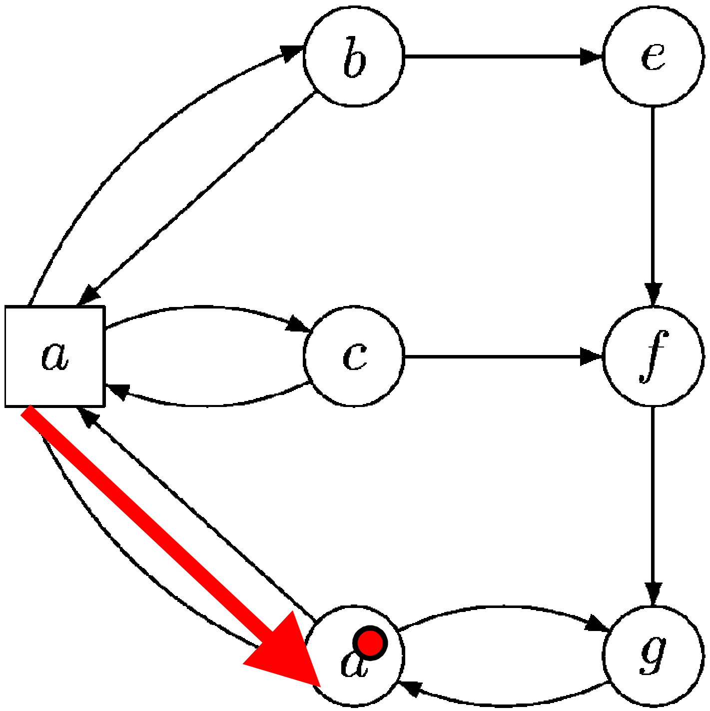
Player 1 (maximizer)

Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games



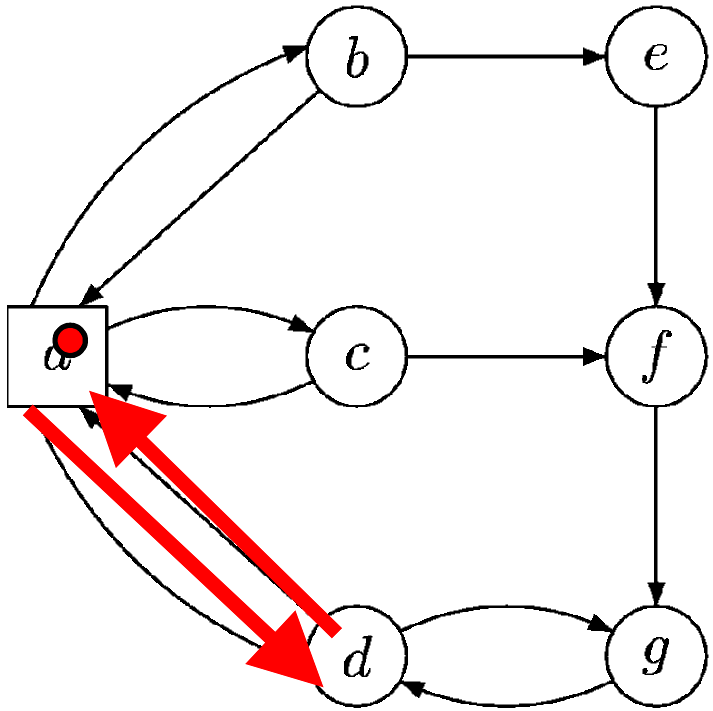
Player 1 (maximizer)

Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games



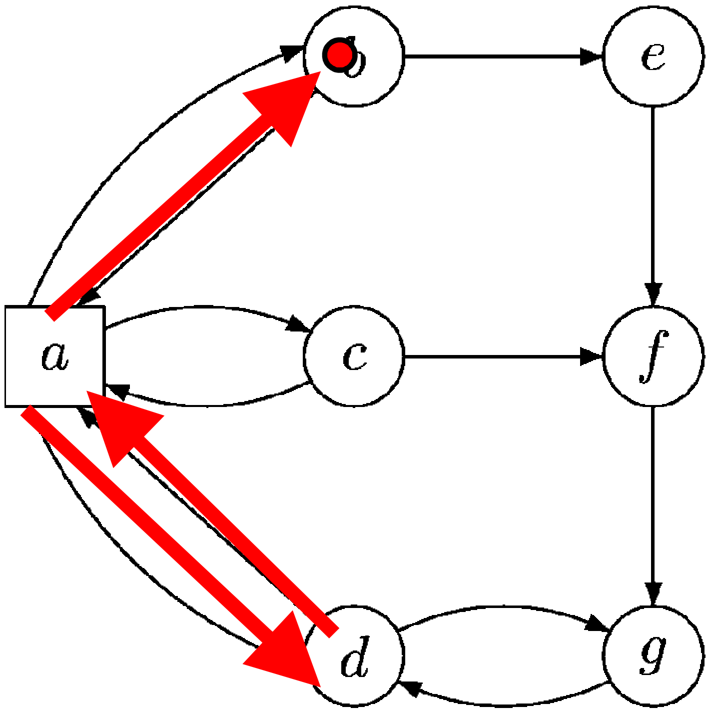
○ Player 1 (maximizer)

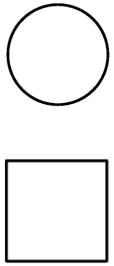
□ Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games

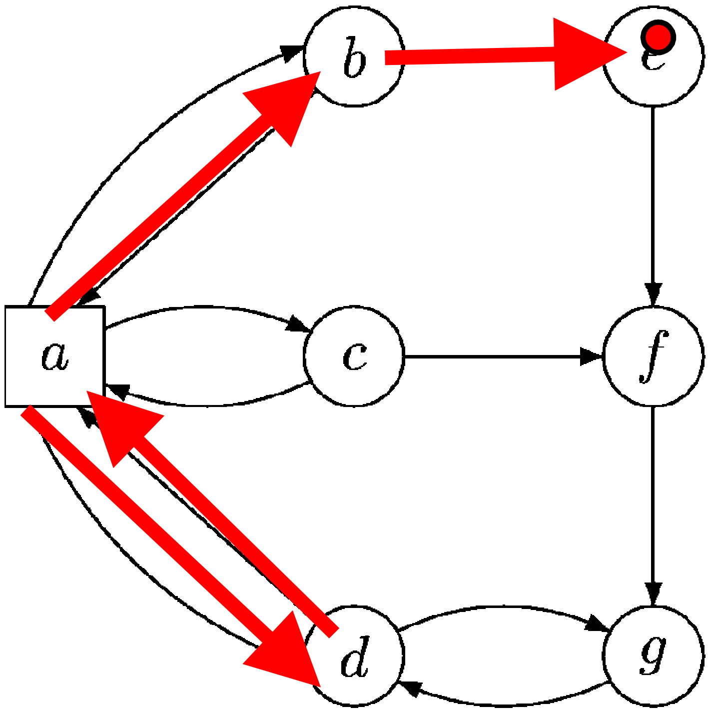


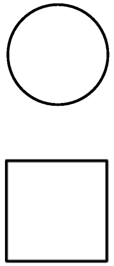


- Turn-based
- Infinite duration

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

Two-player games

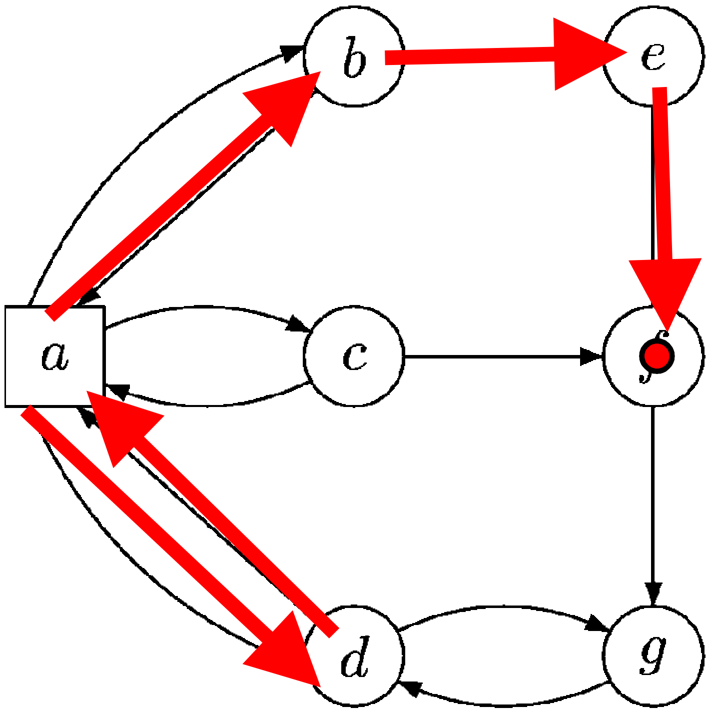


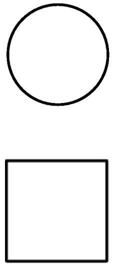


- Turn-based
- Infinite duration

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

Two-player games

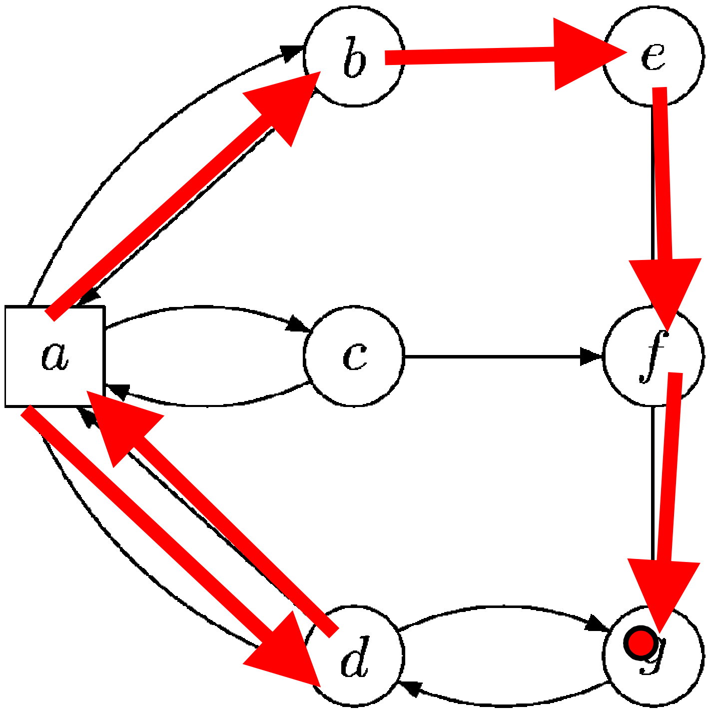




- Turn-based
- Infinite duration

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

Two-player games



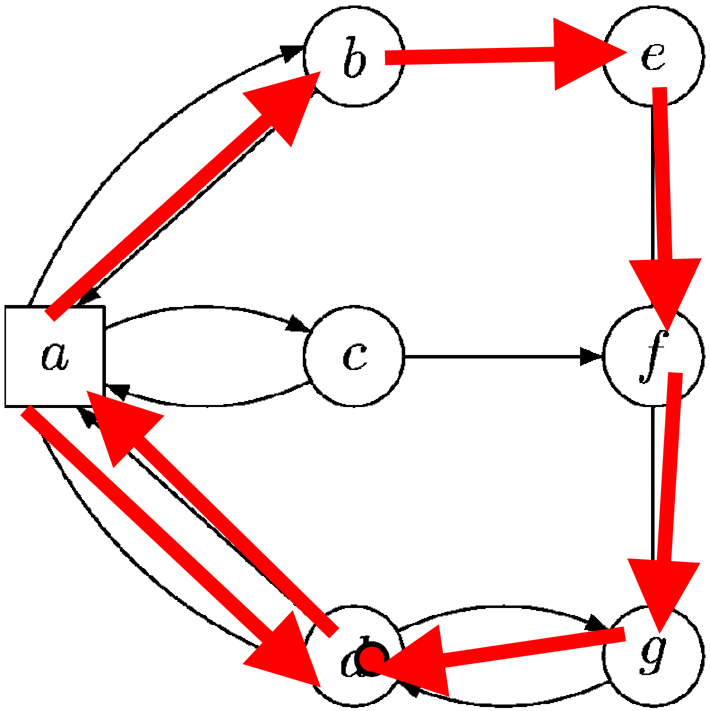
Player 1 (maximizer)

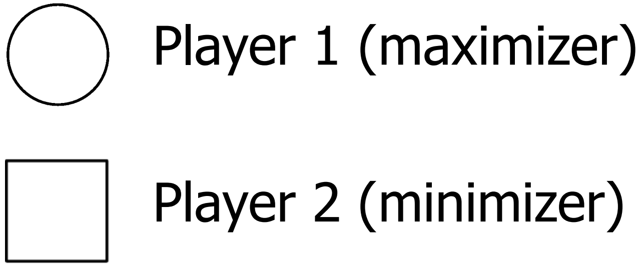
Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games

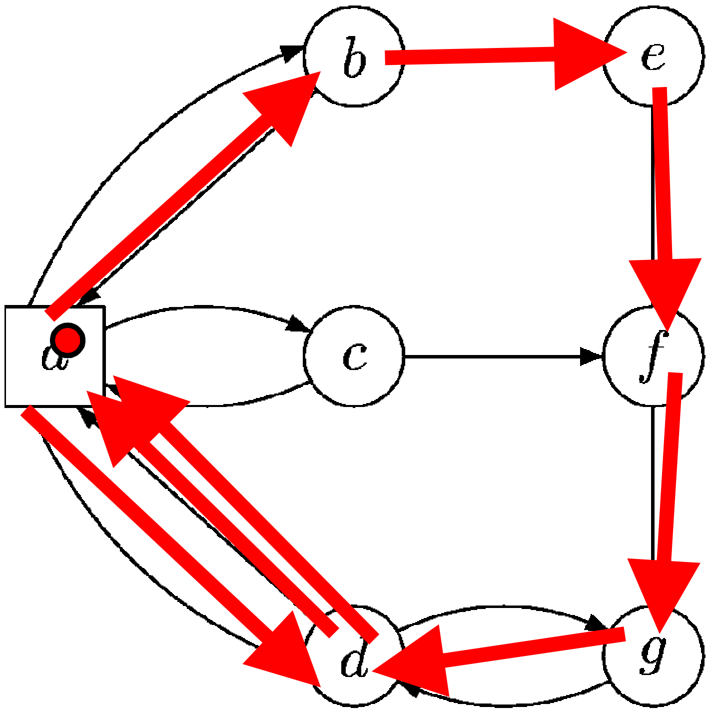


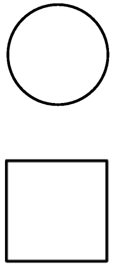


- Turn-based
- Infinite duration

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

Two-player games

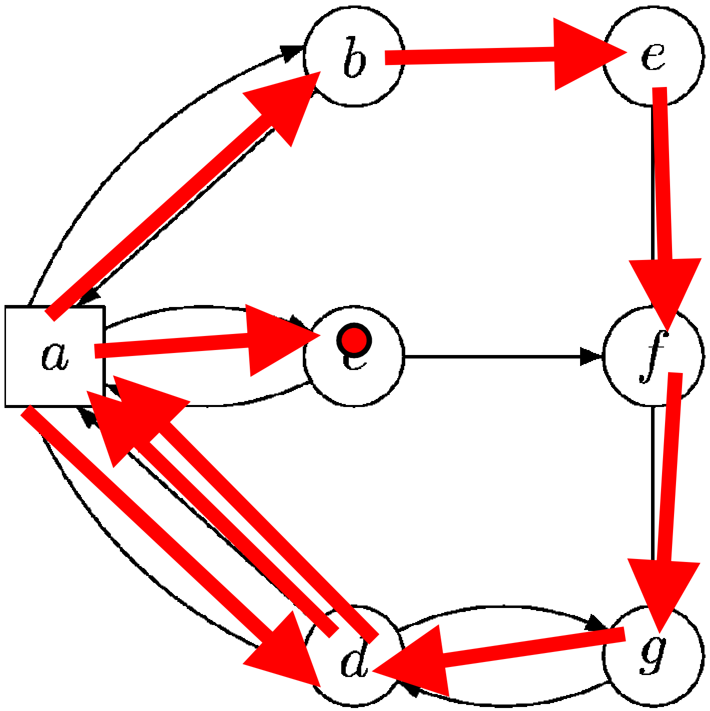




- Turn-based
- Infinite duration

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

Two-player games



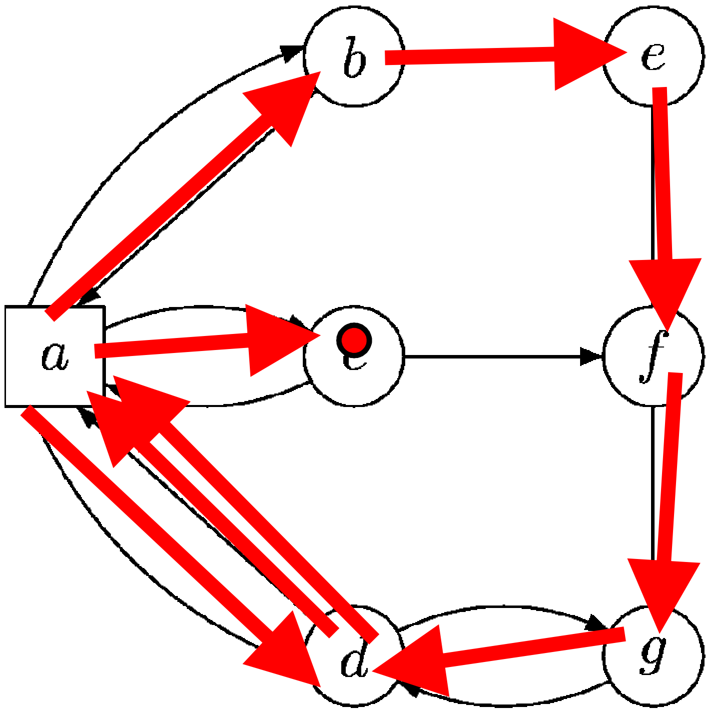
Player 1 (maximizer)

Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games



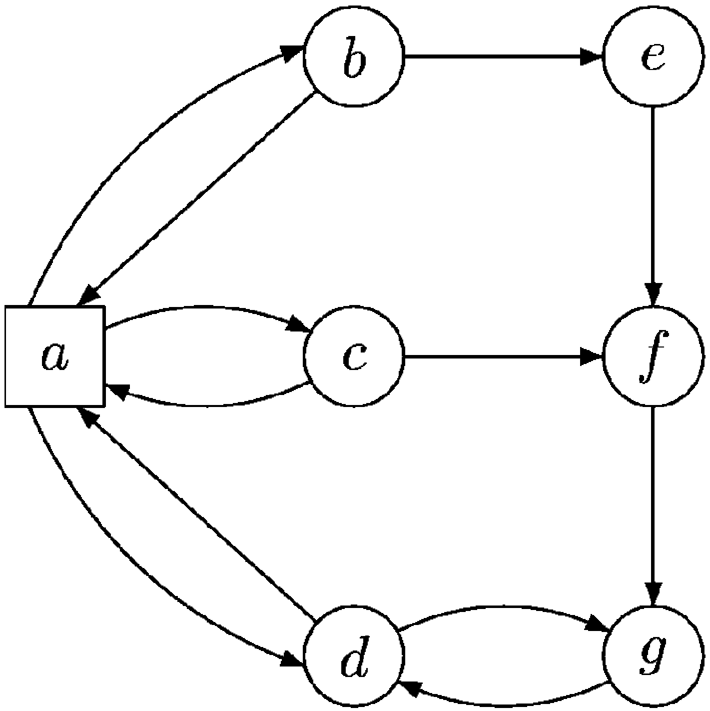
◯ Player 1 (maximizer)

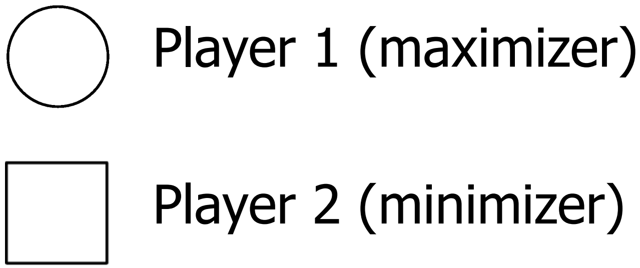
□ Player 2 (minimizer)

- Turn-based
- Infinite duration

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

Two-player games





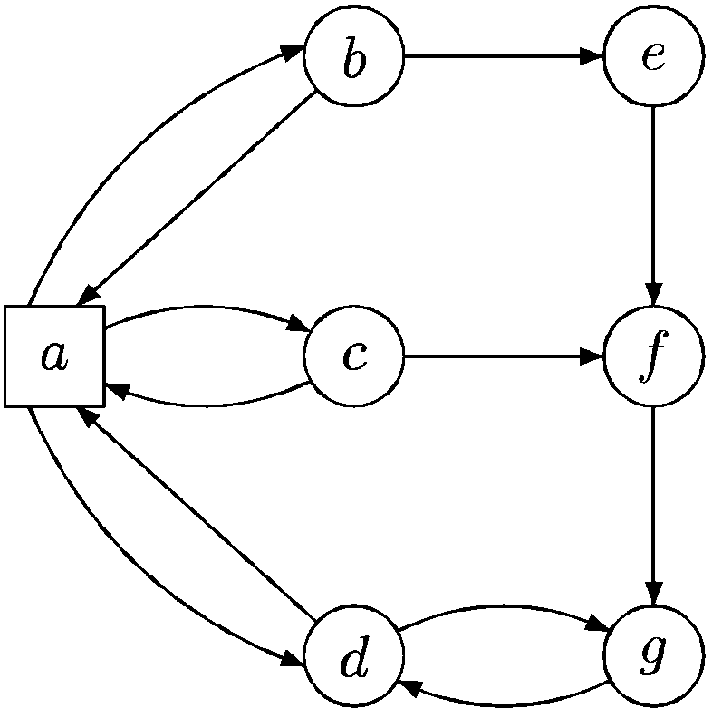
- Turn-based
- Infinite duration

Strategies = recipe to extend the play prefix

Player 1: $\sigma : Q^* \cdot Q_\circ \to Q$

Player 2: $\pi : Q^* \cdot Q_\Box \to Q$

Two-player games



○ Player 1 (maximizer)

□ Player 2 (minimizer)

- Turn-based
- Infinite duration

Strategies = recipe to extend the play prefix

Player 1: $\sigma : Q^* \cdot Q_\circ \to Q$

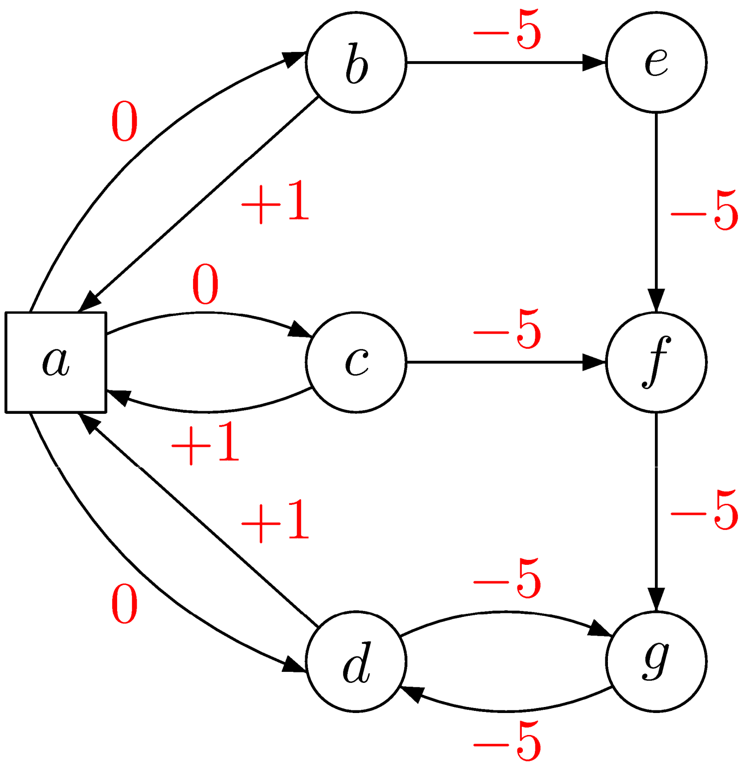
Player 2: $\pi : Q^* \cdot Q_\Box \to Q$

} outcome of two strategies is a play

$\text{outcome}_q^{\sigma,\pi}$

Mean-payoff games

Mean-payoff games



Mean-payoff game:

positive and negative weights

(encoded in binary) $w : E \to \mathbb{Z}$

Decision problem:

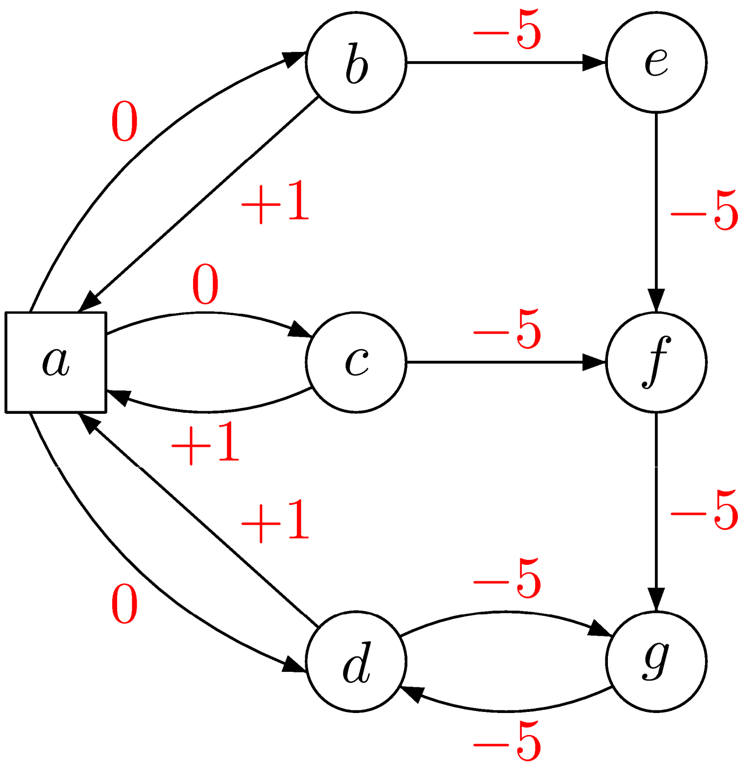
Decide if there exists a player-1 strategy to ensure mean-payoff value ≥ 0

$$\exists \sigma \cdot \forall \pi : \mathsf{MP}(\mathsf{outcome}_q^{\sigma,\pi}) \geq 0$$

Value problem:

$$\sup_\sigma \inf_\pi \mathsf{MP}(\mathsf{outcome}_q^{\sigma,\pi})$$

Mean-payoff games



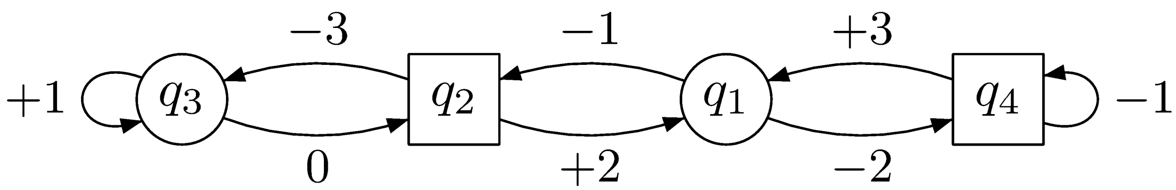
Key ingredients:

- identify memory requirement: infinite vs. finite vs. memoryless
- solve 1-player games (i.e., graphs)

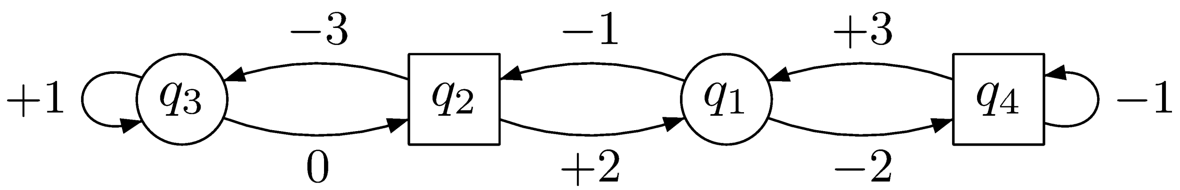
Key arguments for memoryless proof:

- backward induction
- shuffle of plays
- nested memoryless objectives

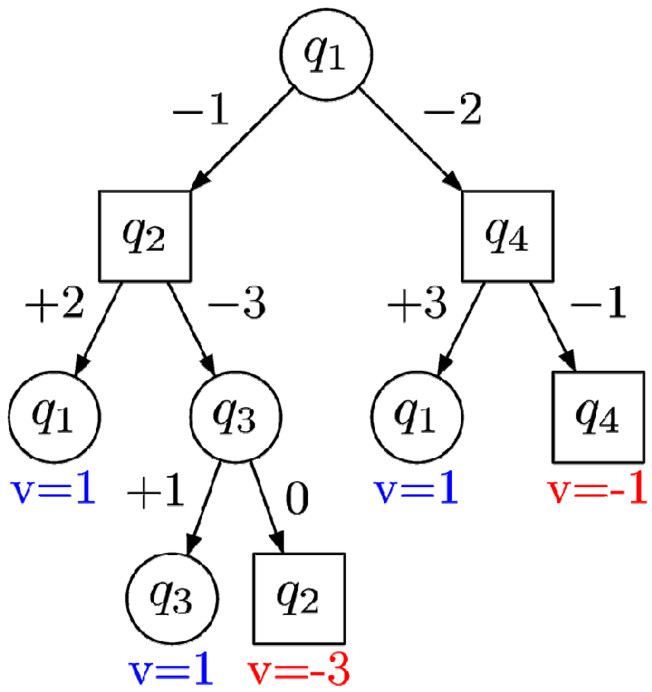
Reduction to Reachability Games



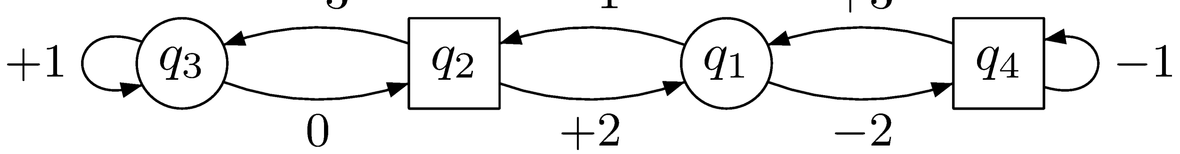
Reduction to Reachability Games



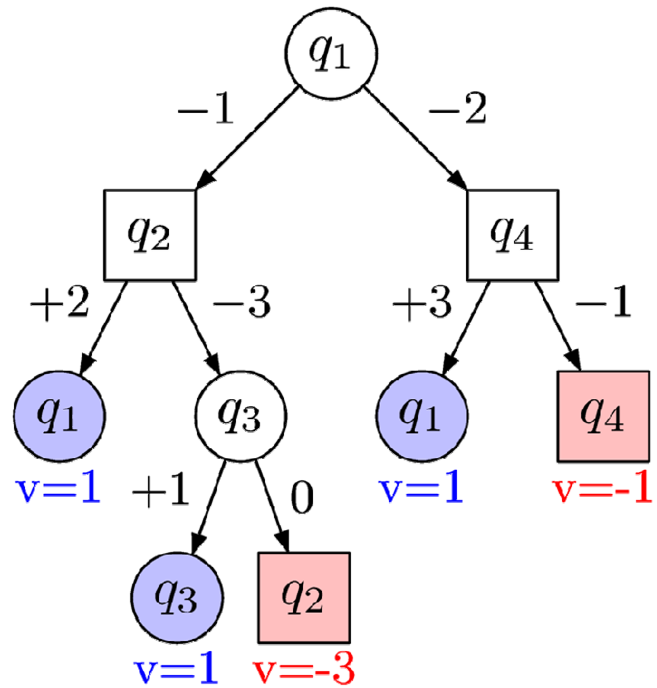
Reachability objective:
positive cycles (v ≥ 0)



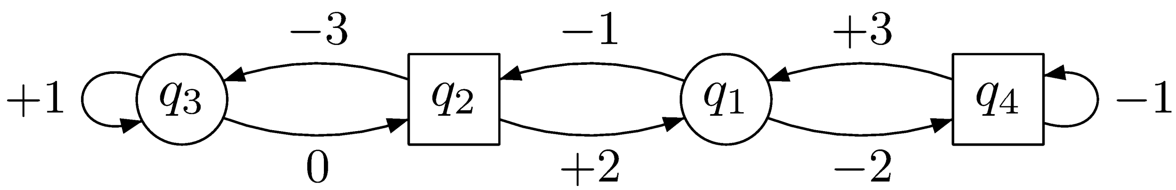
Reduction to Reachability Games



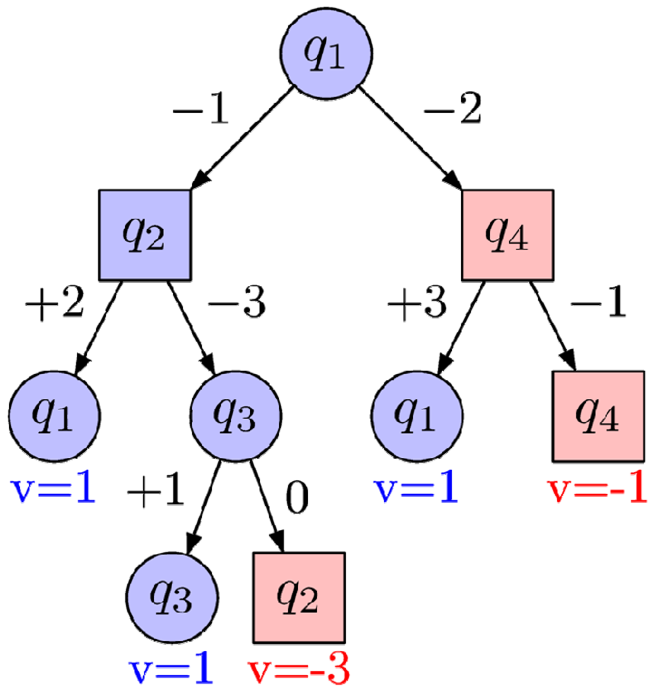
Reachability objective:
positive cycles (v ≥ 0)



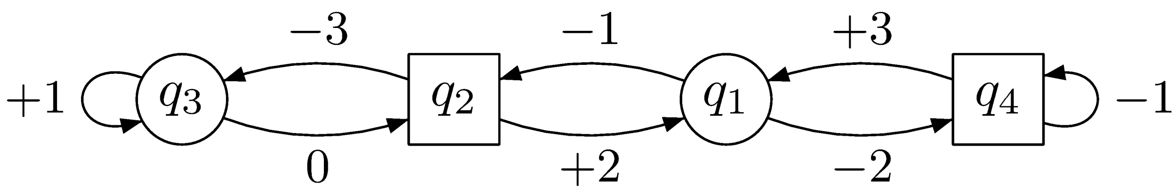
Reduction to Reachability Games



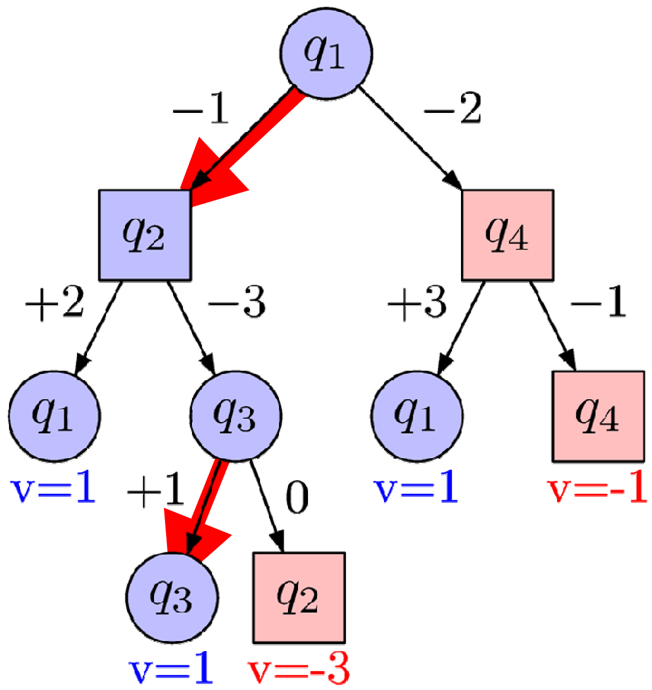
Reachability objective:
positive cycles (v ≥ 0)



Reduction to Reachability Games



Reachability objective:
positive cycles (v ≥ 0)

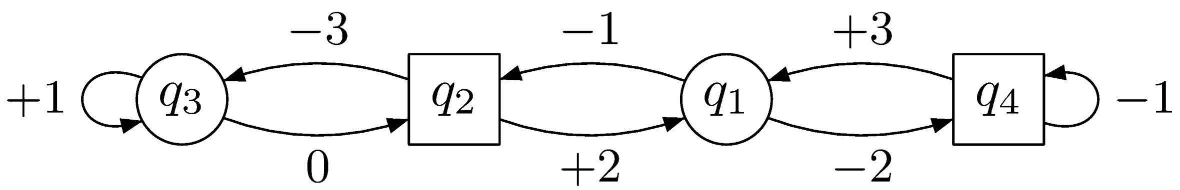


If player 1 wins → only positive cycles are formed → mean-payoff value ≥ 0

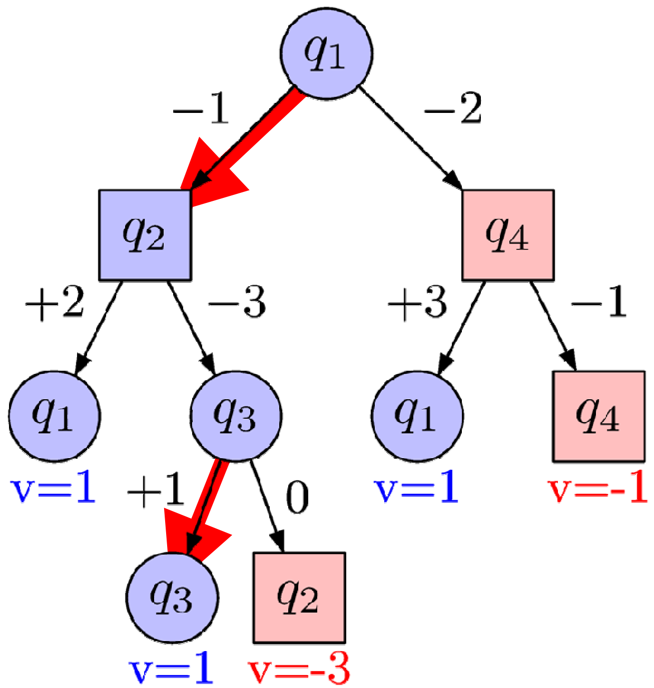
If player 2 wins → only negative cycles are formed → mean-payoff value < 0

(Note: limsup vs. liminf does not matter)

Reduction to Reachability Games



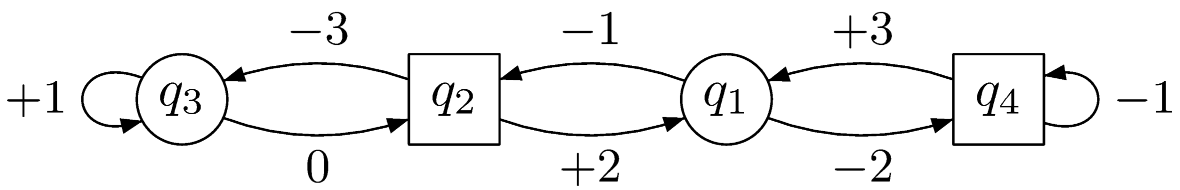
Reachability objective:
positive cycles (v ≥ 0)



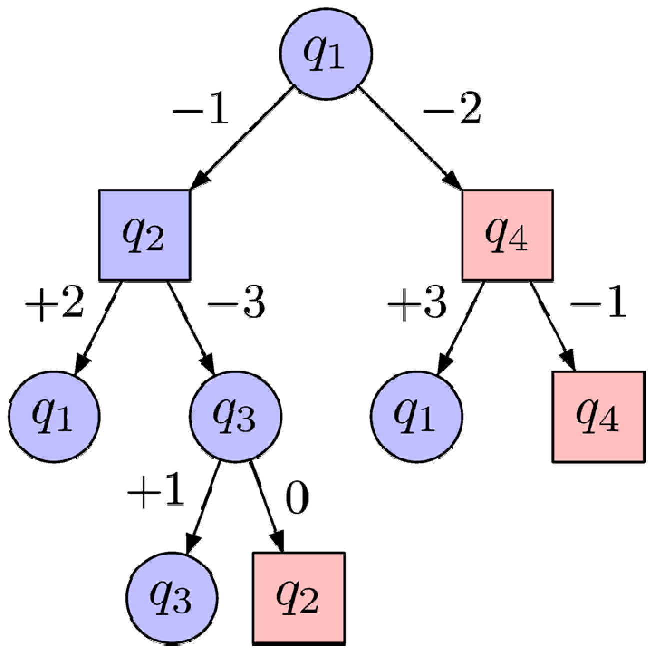
Mean-payoff game ⇔ Ensuring positive cycles

Memoryless strategy transfers to finite-memory mean-payoff winning strategy

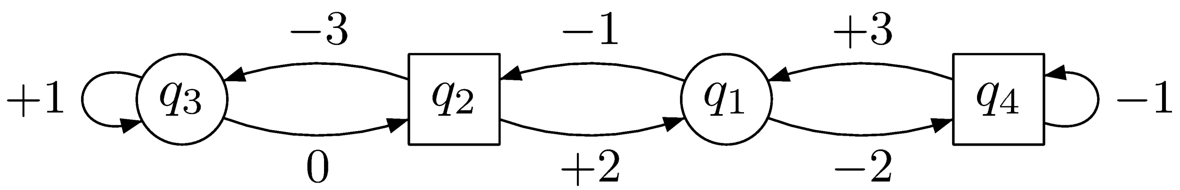
Strategy Synthesis



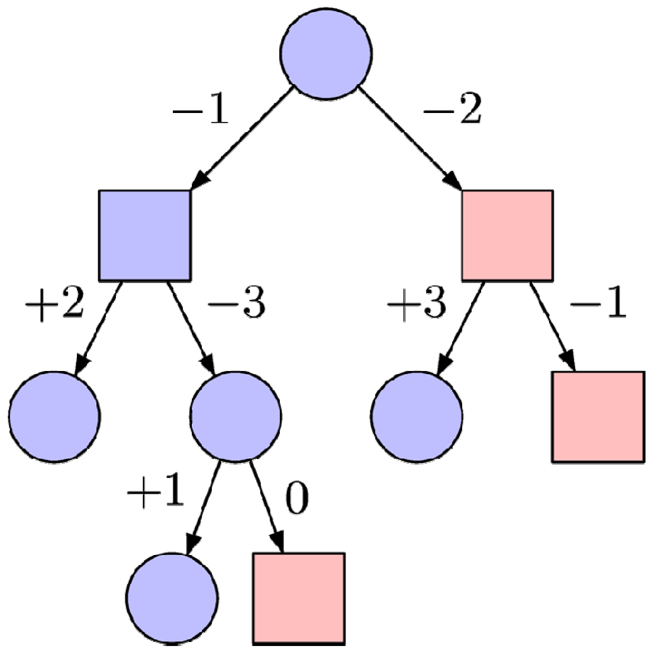
Memoryless mean-payoff winning strategy ?



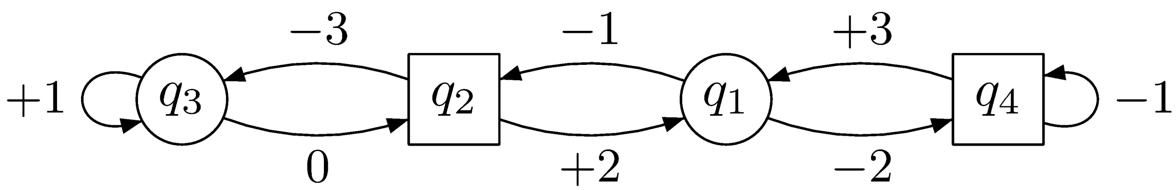
Strategy Synthesis

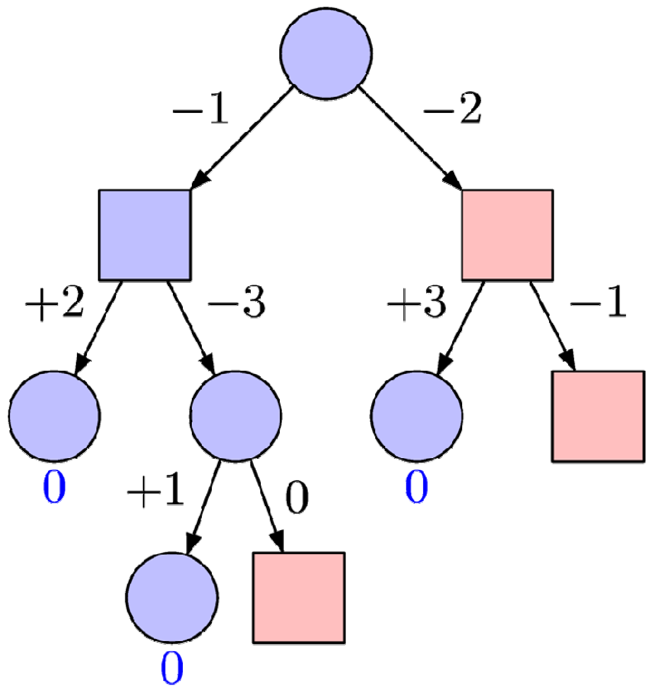


Memoryless mean-payoff winning strategy ?



Strategy Synthesis

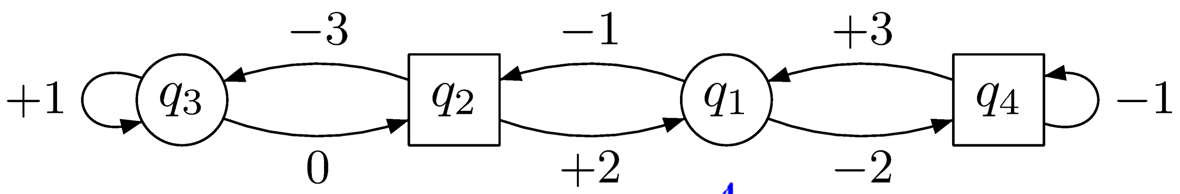


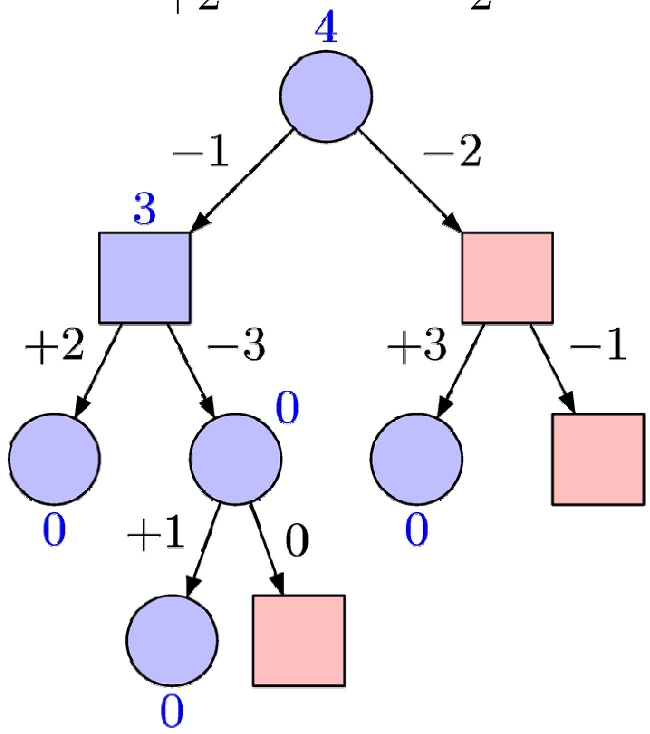


Memoryless mean-payoff winning strategy ?

Progress measure: minimum initial credit to stay always positive

Strategy Synthesis

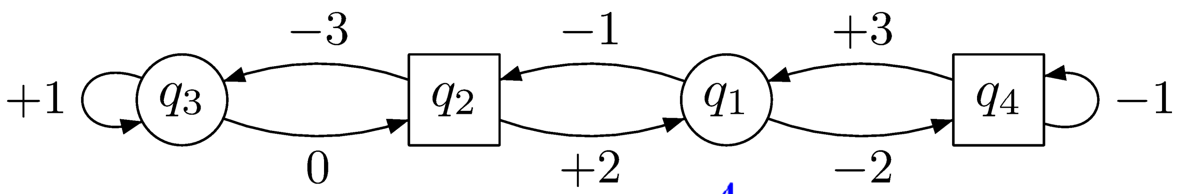


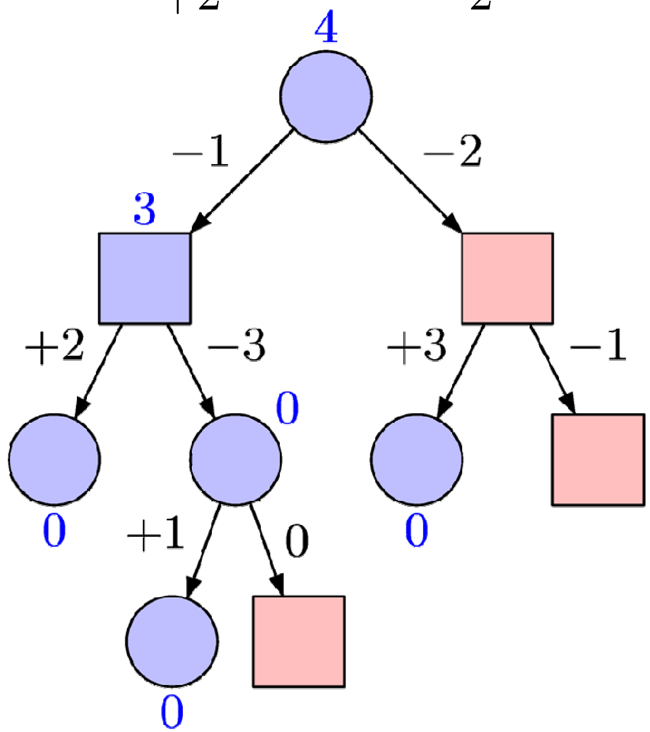


Memoryless mean-payoff winning strategy ?

Progress measure: minimum initial credit to stay always positive

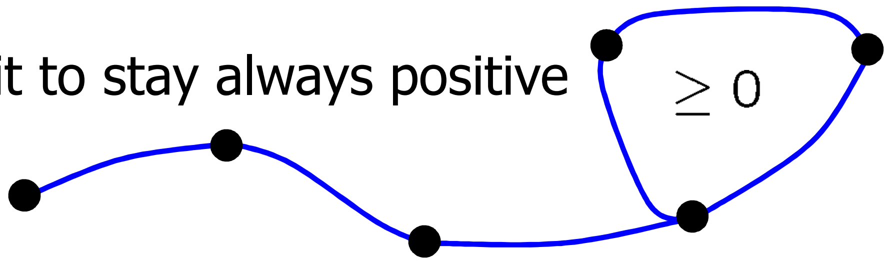
Strategy Synthesis



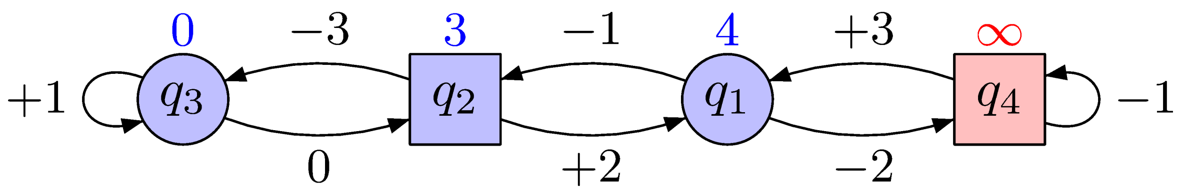


Memoryless mean-payoff winning strategy ?

Progress measure: minimum initial credit to stay always positive



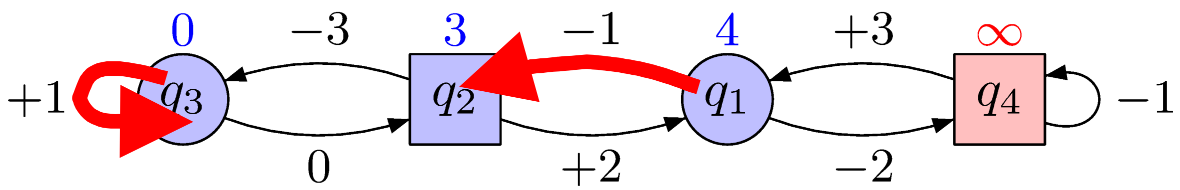
Strategy Synthesis



Memoryless mean-payoff winning strategy ?

Progress measure: minimum initial credit to stay always positive

Strategy Synthesis



Memoryless mean-payoff winning strategy ?

Choose successor to stay above minimum credit

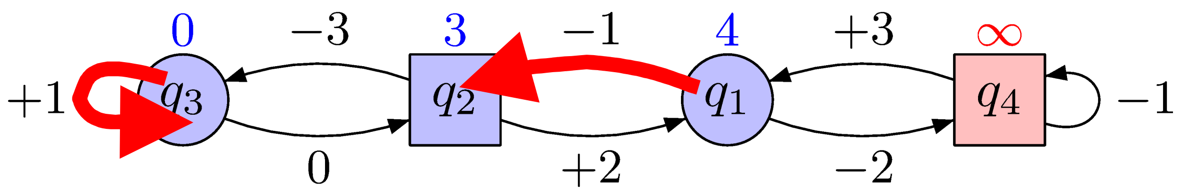
$\mu : Q \to \mathbb{N}$ minimum credit such that

$$\exists \sigma \cdot \forall \pi \cdot \forall n : \mu(q_0) + \sum_{i=0}^{n} \mathsf{w}(q_i, q_{i+1}) \geq 0$$

Progress measure: minimum initial credit to stay always positive

In $q \in Q_\circ$ choose q' such that $\mu(q) + \mathsf{w}(q, q') \geq \mu(q')$

Strategy Synthesis



Memoryless mea...
winning st...

"Energy Game"
(stay always positive,
for some initial credit)

...cessor to stay above minimum credit

$\mu : Q \to \mathbb{N}$ minimum credit such that

$$\exists \sigma \cdot \forall \pi \cdot \forall n : \mu(q_0) + \sum_{i=0}^{n} \mathrm{w}(q_i, q_{i+1}) \geq 0$$

Progress measure: minimum initial credit to stay always positive

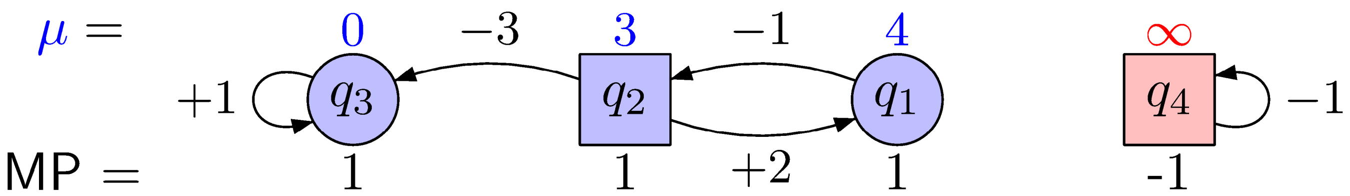
In $q \in Q_\circ$ choose q' such that $\mu(q) + \mathrm{w}(q, q') \geq \mu(q')$

Memoryless proofs

Key arguments for memoryless proof:

- backward induction
- shuffle of plays
- nested memoryless objectives

Energy Games

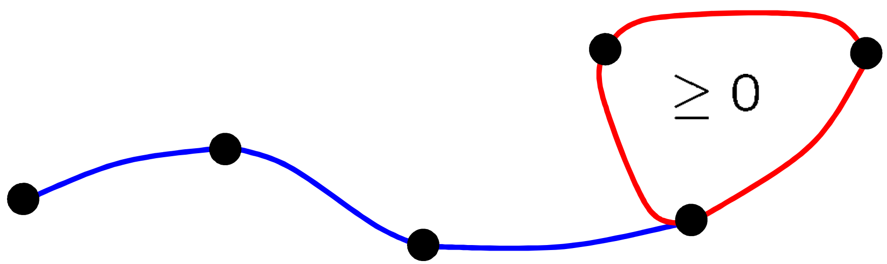


Energy: min-value of the prefix.
(if positive cycle; otherwise ∞)

$$\mu = -\min_{n \in \mathbb{N}} \sum_{i=0}^{n-1} \mathsf{w}_i$$

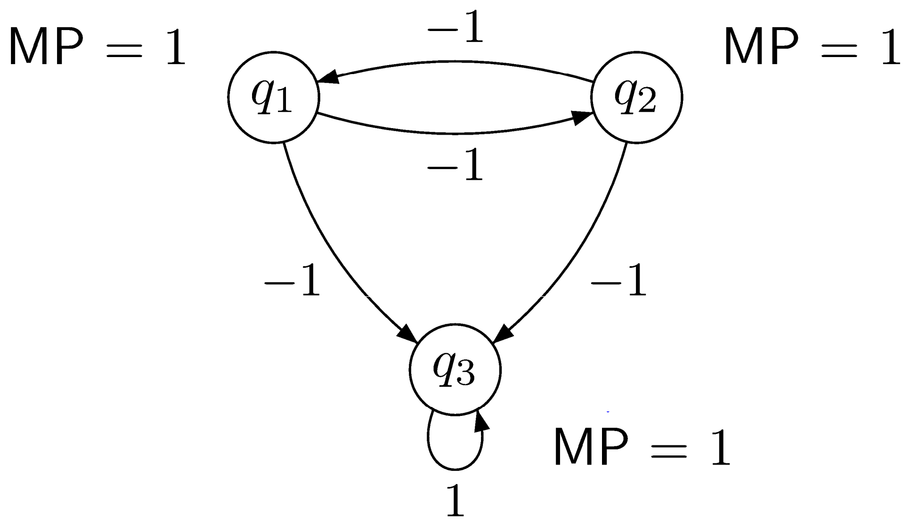
Mean-payoff: average-value of the cycle.

$$\mathsf{MP} = \lim_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{w}_i$$



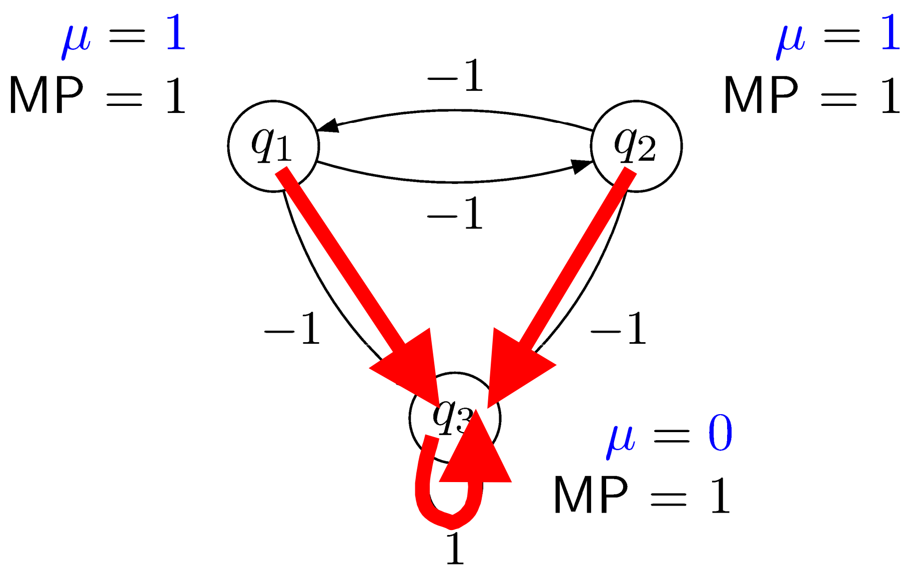
NP ∩ coNP

Energy Games



Winning strategy ?

Energy Games

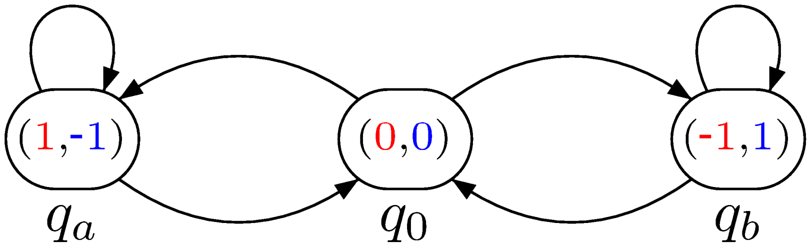


Winning strategy ?

Follow the minimum initial credit !

Multi-dimension games

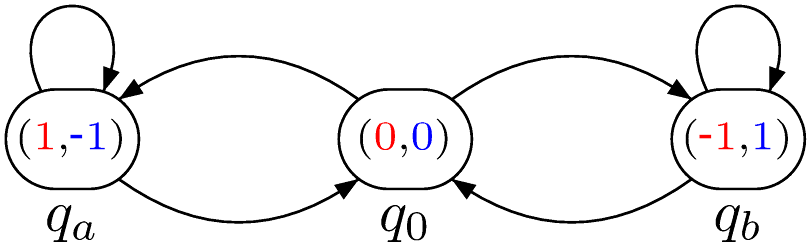
Multi-dimension games



Multiple resources $\mathsf{w} : Q \to \mathbb{Z}^d$

- Energy: initial credit to stay always above (0,0)
- Mean-payoff: $\mathsf{MP}(\mathsf{w}_1) \geq 0 \wedge \mathsf{MP}(\mathsf{w}_2) \geq 0$

Multi-dimension games



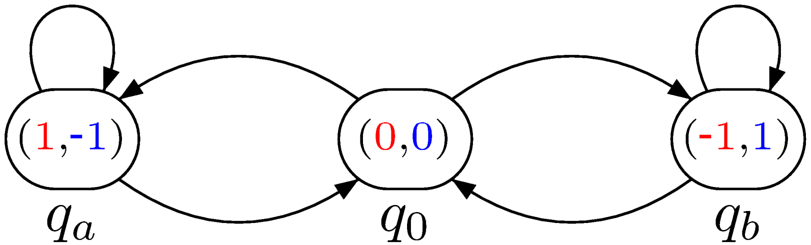
Multiple resources $w : Q \to \mathbb{Z}^d$

- Energy: initial credit to stay always above (0,0)
- Mean-payoff: $\mathrm{MP}(w_1) \geq 0 \wedge \mathrm{MP}(w_2) \geq 0$

same ?

same as positive cycles ?

Multi-dimension games



Multiple resources $w : Q \to \mathbb{Z}^d$

- Energy: initial credit to stay always above (0,0)
- Mean-payoff: $\mathrm{MP}(w_1) \geq 0 \wedge \mathrm{MP}(w_2) \geq 0$

same ?

same as positive cycles ?

If player 1 can ensure positive simple cycles, then energy and mean-payoff are satisfied.

Not the converse !

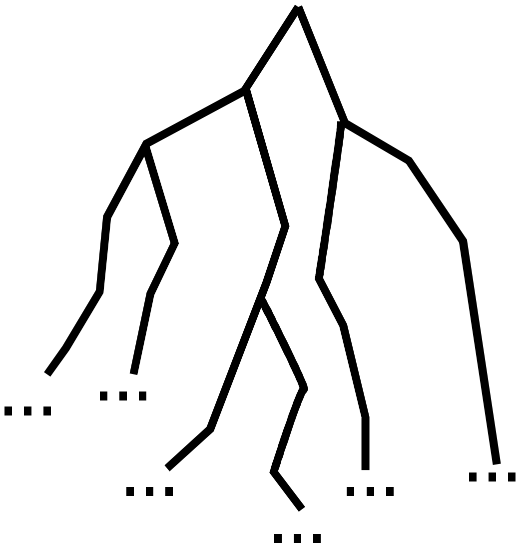
Multi-dimension games

If player 1 has initial credit to stay always positive (Energy)
then finite-memory strategies are sufficient

Multi-dimension games

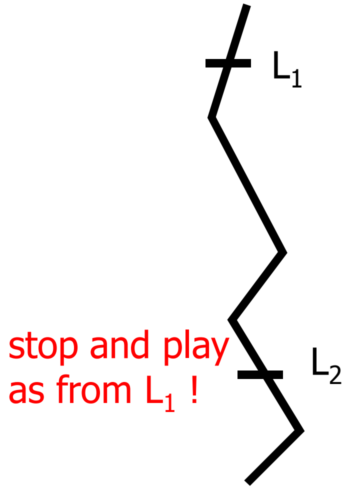
If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient

Let σ_1 be winning



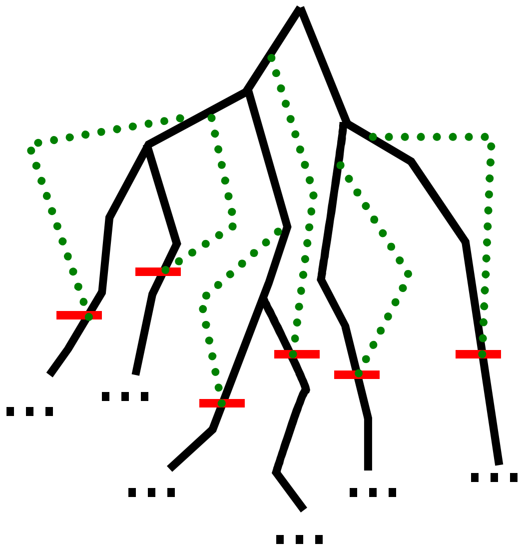
$(\mathbb{N}^d,\leq)$ is well-quasi ordered

On each branch



With $L_1 \leq L_2$

Then σ'_1 is winning and finite memory



wqo + Koenig's lemma

Multi-energy games

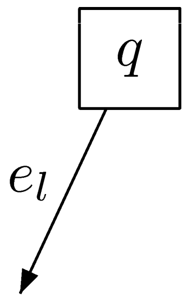
If player 1 has initial credit to stay always positive (Energy)
then finite-memory strategies are sufficient

For player 2 ?

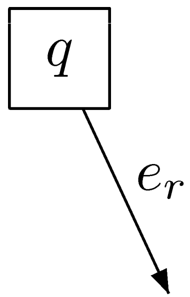
Multi-energy games

For player 2, memoryless strategies are sufficient

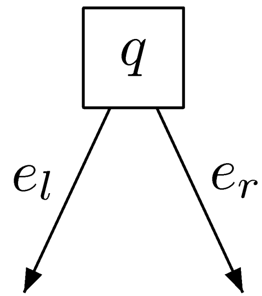
- induction on player-2 states
- if ∃ initial credit against all memoryless strategies, then ∃ initial credit against all arbitrary strategies.



'left' game



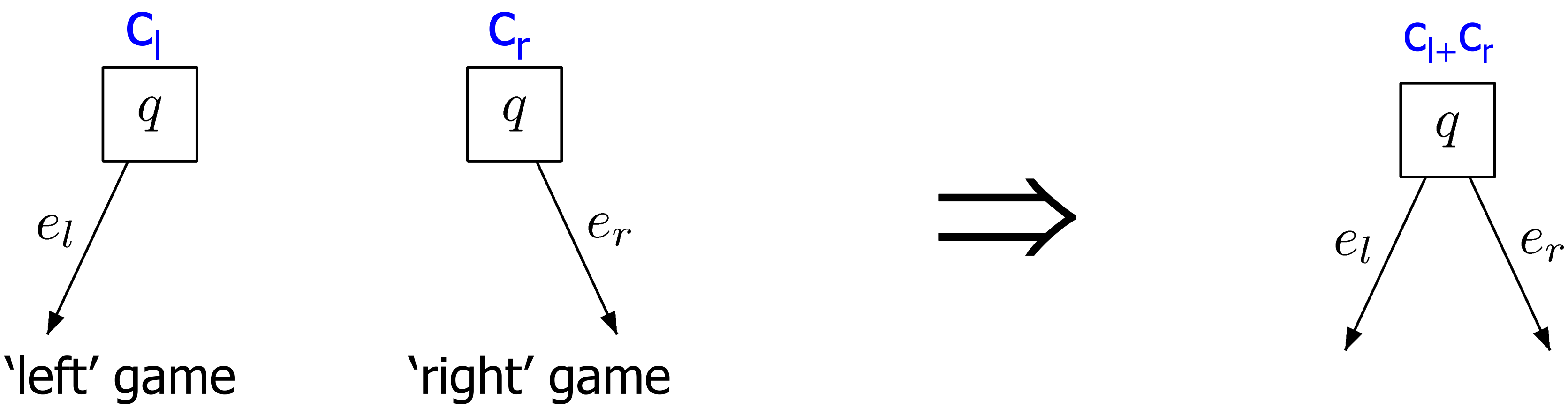
'right' game



Multi-energy games

For player 2, memoryless strategies are sufficient

- induction on player-2 states
- if ∃ initial credit against all memoryless strategies, then ∃ initial credit against all arbitrary strategies.



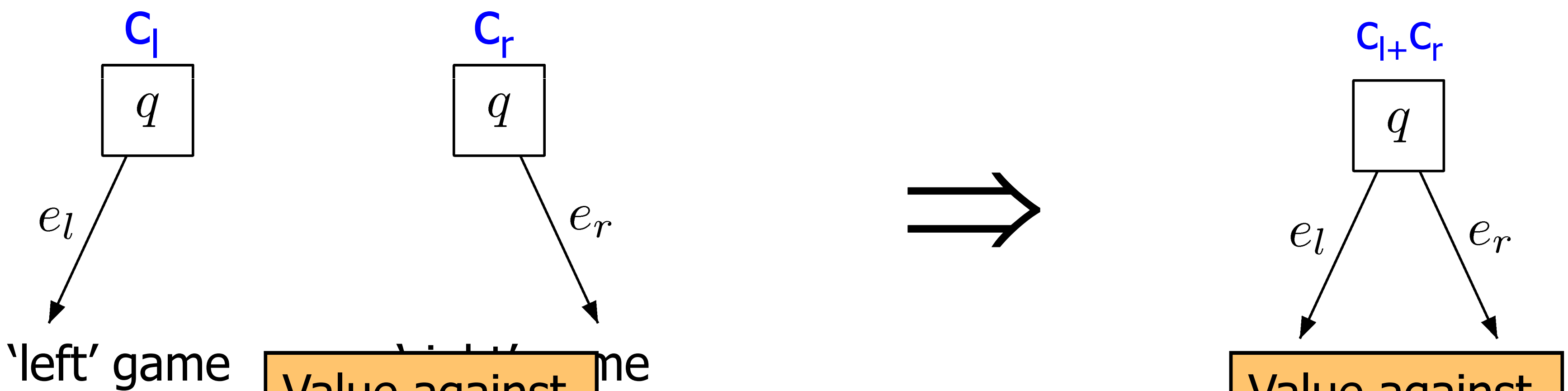
Play is a shuffle of left-game play and right-game play

Energy is sum of them

Multi-energy games

For player 2, memoryless strategies are sufficient

- induction on player-2 states
- if ∃ initial credit against all memoryless strategies, then ∃ initial credit against all arbitrary strategies.



Value against memoryless strategies

Play is a shuffle of
and right-game pl

Energy is sum of them

Value against arbitrary strategies

In general, we need $\min(\text{val}(\rho_L), \text{val}(\rho_R)) \leq \text{val}(\text{shuffle}(\rho_L, \rho_R))$

Memoryless proofs

Key arguments for memoryless proof:

- backward induction
- shuffle of plays
- nested memoryless objectives

Multi-energy games

If player 1 has initial credit to stay always positive (Energy)
then finite-memory strategies are sufficient

For player 2, memoryless strategies are sufficient

coNP ?

Multi-energy games

If player 1 has initial credit to stay always positive (Energy)
then finite-memory strategies are sufficient

For player 2, memoryless strategies are sufficient

coNP ?

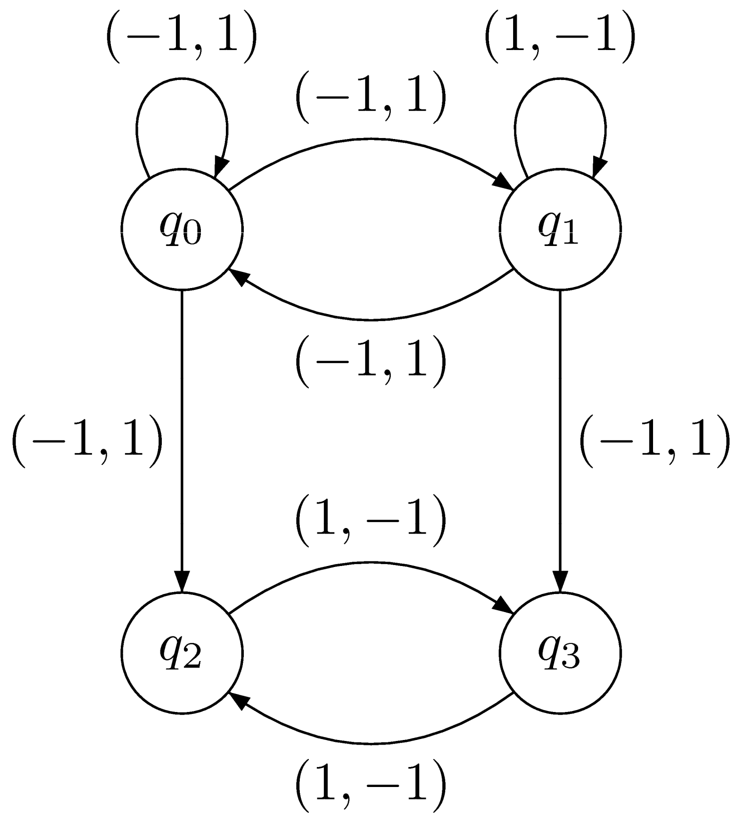
not necessarily simple cycle!

- guess a memoryless strategy π for Player 2
- Construct G_π
- check in polynomial time that G_π contains no cycle with nonnegative effect in all dimensions

Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

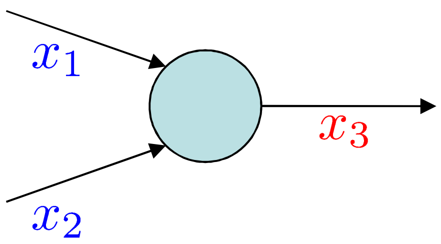
- Flow constraints using LP
- Divide and conquer algorithm



Multi-weighted energy games

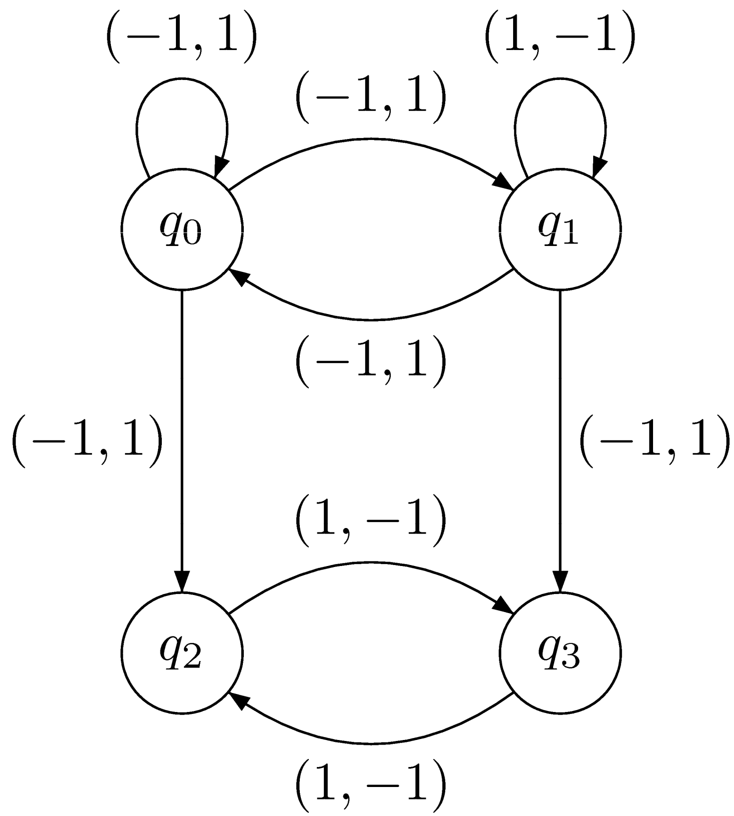
Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP



$$x_1 + x_2 = x_3$$

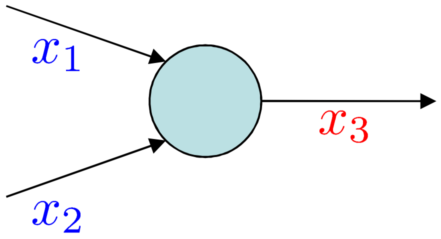
$$\sum_i x_i \cdot w_i \geq 0$$



Multi-weighted energy games

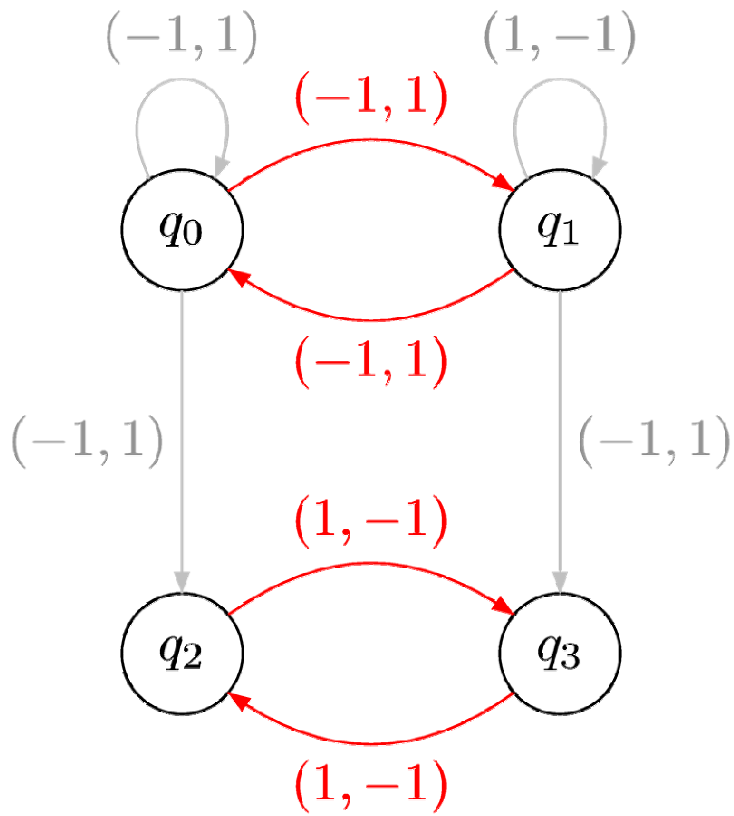
Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP



$$x_1 + x_2 = x_3$$

$$\sum_i x_i \cdot w_i = 0$$



Not connected !

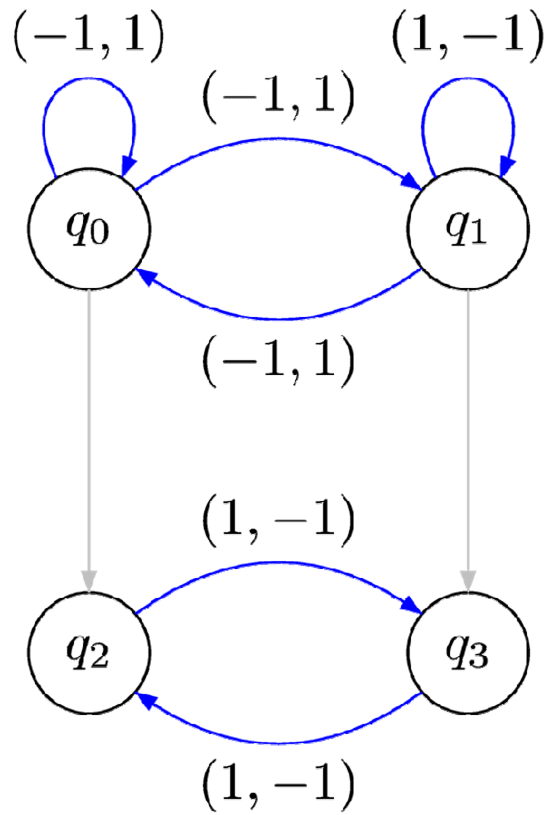
Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP
- Divide and conquer algorithm

Mark the edges that belong to some (pseudo) solution.

Solve the connected subgraphs.



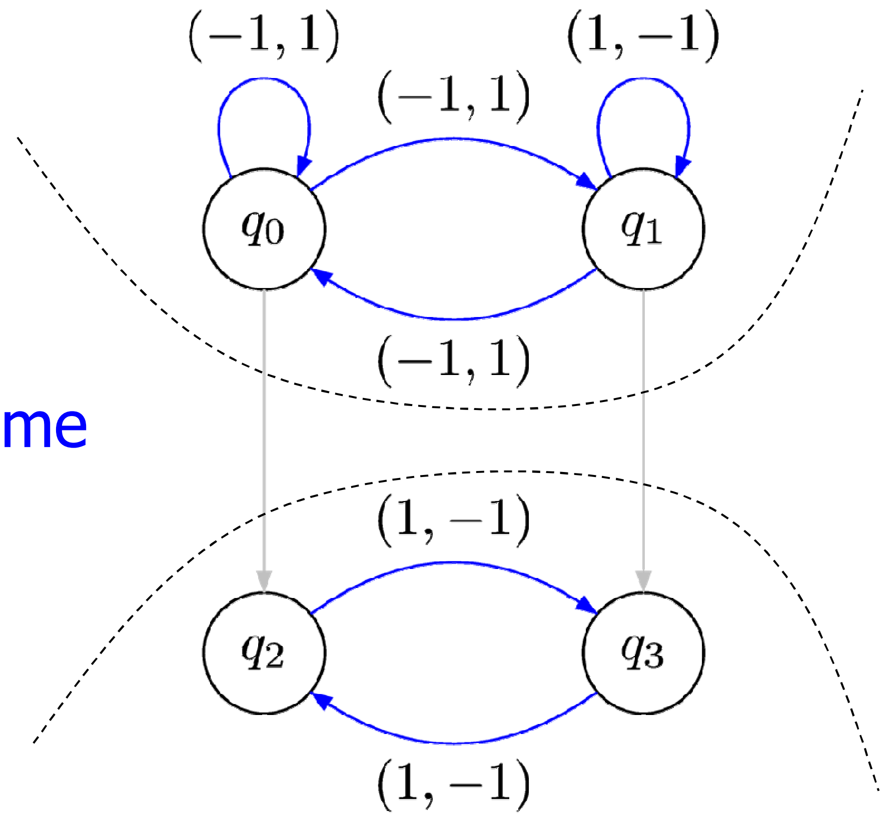
Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP
- Divide and conquer algorithm

Mark the edges that belong to some (pseudo) solution.

Solve the connected subgraphs.



Multi-dimension games

If player 1 has initial credit to stay always positive (Energy)
then finite-memory strategies are sufficient

For player 2, memoryless strategies are sufficient

Equivalent with mean-payoff games (under finite-memory):

If player 1 wins → positive cycles are formed → mean-payoff value ≥ 0

Otherwise, for all finite-memory strategy of player 1 (with memory M), player 2 can repeat a negative cycle (in G x M)

Multi-dimension games

Player 1	Energy	$\underline{\text{MP}}$ - liminf	$\overline{\text{MP}}$ - limsup
Finite memory	coNP-complete Player 2 memoryless		
Infinite memory			

Multi-dimension games

<table>
<tr><th>Player 1</th><th>Energy</th><th>$\underline{\mathsf{MP}}$ - liminf</th><th>$\overline{\mathsf{MP}}$ - limsup</th></tr>
<tr><td>Finite memory</td><td colspan="3">coNP-complete
Player 2 memoryless</td></tr>
<tr><td>Infinite memory</td><td></td><td>coNP-complete
Pl. 2 memoryless</td><td></td></tr>
</table>

- Player 2 memoryless (shuffle argument)
- Graph problem in PTIME (LP argument)

$$\min(\mathsf{MP}(\rho_L), \mathsf{MP}(\rho_R)) \leq \mathsf{MP}(\text{shuffle}(\rho_L, \rho_R))$$

- True for $\underline{\mathsf{MP}}$
- False for $\overline{\mathsf{MP}}$

Multi-mean-payoff games

The winning region R of player 1 has the following characterization:

Player 1 wins $\bigwedge_i \overline{\mathsf{MP}}(w_i) \geq 0$ from every state in R

if and only if player 1 wins each $\overline{\mathsf{MP}}(w_i) \geq 0$ from every state in R

(without leaving R)

Proof idea: $\Box\Diamond(1 \wedge 2) \equiv \Box\Diamond 1 \wedge \Box\Diamond 2$

Multi-mean-payoff games

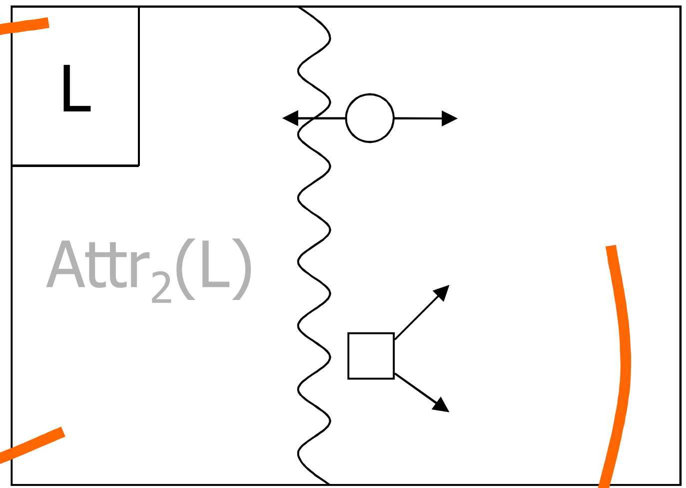
The winning region R of player 1 has the following characterization:

Player 1 wins $\bigwedge_i \overline{\text{MP}}(w_i) \geq 0$ from every state in R

if and only if player 1 wins each $\overline{\text{MP}}(w_i) \geq 0$ from every state in R

(without leaving R)

Proof idea: $\Box\Diamond(1 \wedge 2) \equiv \Box\Diamond 1 \wedge \Box\Diamond 2$



Losing for player 1 for single objective

Winning for player 2, with memoryless strategy

By induction, player 2 is memoryless in the subgame

Memoryless proofs

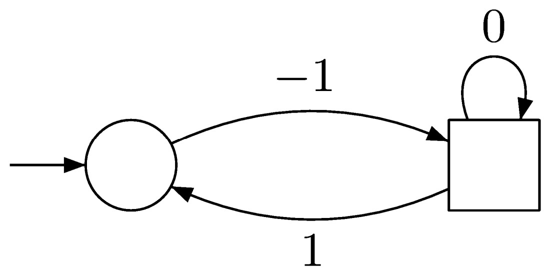
Key arguments for memoryless proof:

- backward induction
- shuffle of plays
- nested memoryless objectives

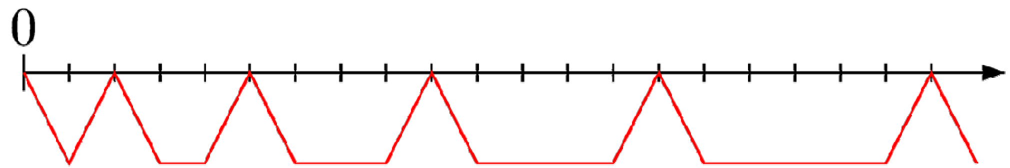
Multi-dimension games

Player 1	Energy	MP - liminf	MP - limsup
Finite memory	coNP-complete Player 2 memoryless		
Infinite memory		coNP-complete Pl. 2 memoryless	NP $\cap$ coNP Pl. 2 memoryless

Window games

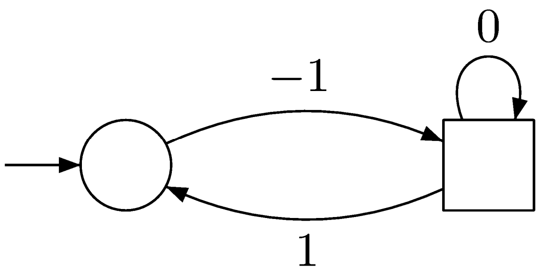


Issues with mean-payoff



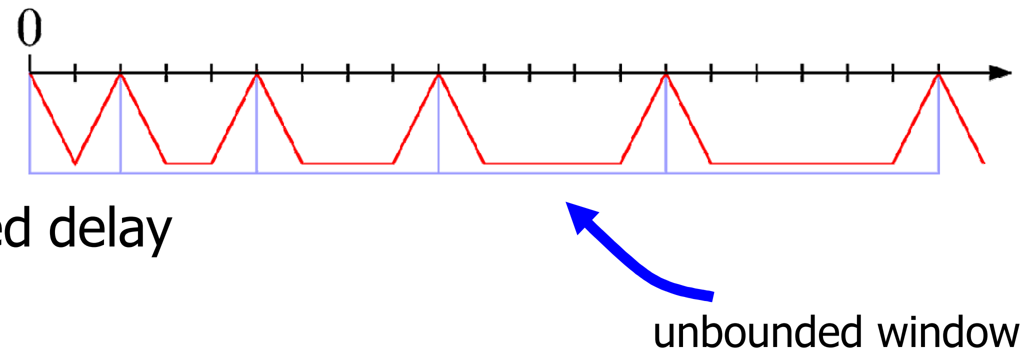
- limsup vs. liminf
- limit-behaviour, unbounded delay
- complexity

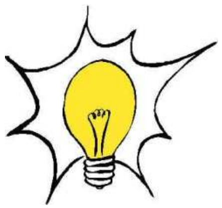
Window games



Issues with mean-payoff

- limsup vs. liminf
- limit-behaviour, unbounded delay
- complexity





Sliding window of size at most B

At every step, MP ≥ 0 within the window

Window games

Window objective:

from some point on, at every step, MP ≥ 0 within window of B steps

prefix-independent

bounded delay

Implies the mean-payoff condition

Window games

Window objective:

from some point on, at every step, MP ≥ 0 within window of B steps

prefix-independent

bounded delay

Implies the mean-payoff condition

Complexity, Algorithm ?

- like coBüchi objective $\Diamond\Box(\Sigma^{\leq B} \geq 0)$

$O(V^2 \cdot E \cdot B \cdot \log W)$

min-max cost (for ≤B steps)

stable set (safety)

attractor & subgame iteration

Window games

Window objective:

from some point on, at every step, MP ≥ 0 within window of B steps

prefix-independent

bounded delay

Implies the mean-payoff condition

Complexity, Algorithm ?

- like coBüchi objective $\Diamond\Box(\Sigma^{\leq B} \geq 0)$ $O(V^2 \cdot E \cdot B \cdot \log W)$

- multi-dimension: EXPTIME-complete

Hyperplane Separation

Multi-dimension mean-payoff (liminf): coNP-complete

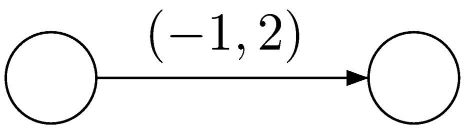
Naive algorithm: exponential in number of states

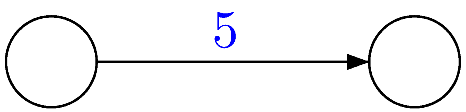
Hyperplane separation: reduction to single-dimension mean-payoff games

$\vec{\lambda} = (1, 3)$

$\mathsf{w} = (-1, 2)$

$\vec{\lambda} \cdot \mathsf{w}^T = 1 \cdot (-1) + 3 \cdot 2 = 5$



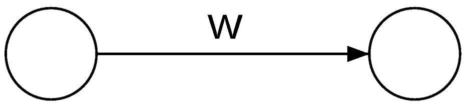


Hyperplane Separation

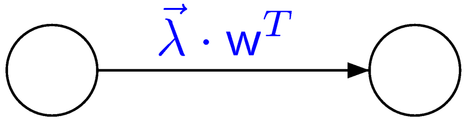
Multi-dimension

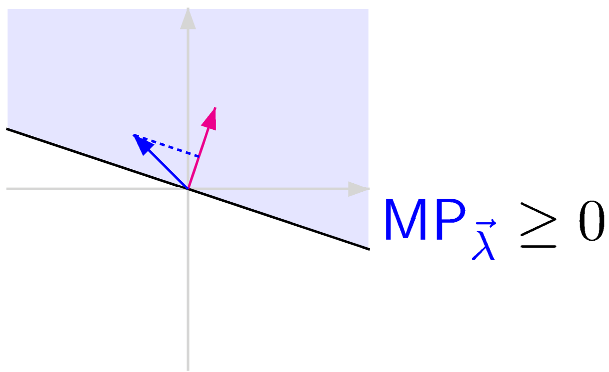
Single dimension

$$\vec{\lambda} = (1, 3)$$

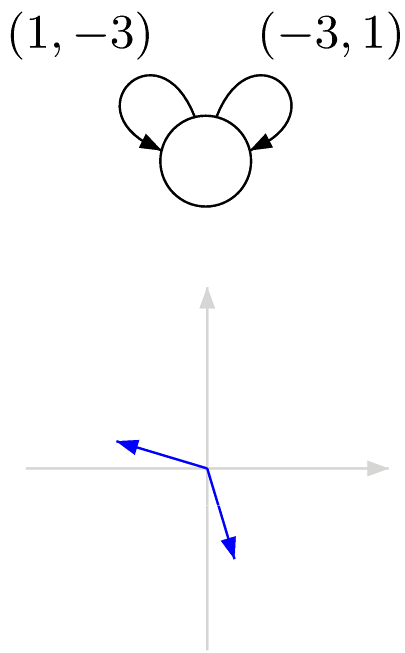


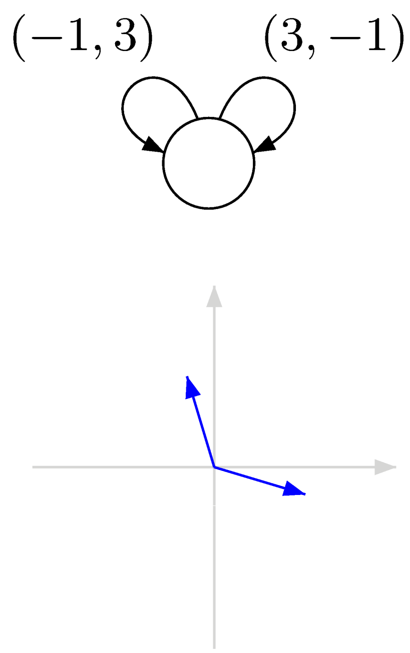
$$\Rightarrow$$



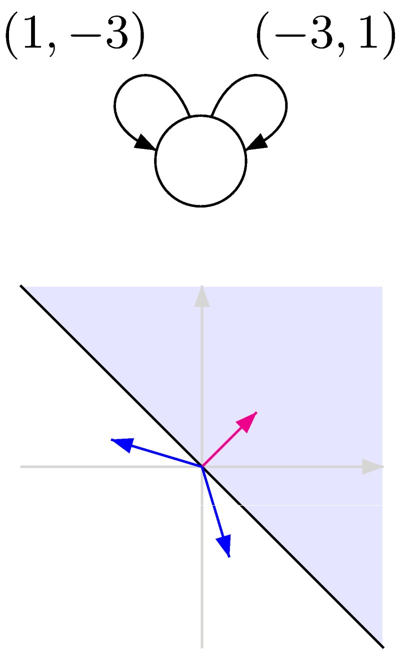


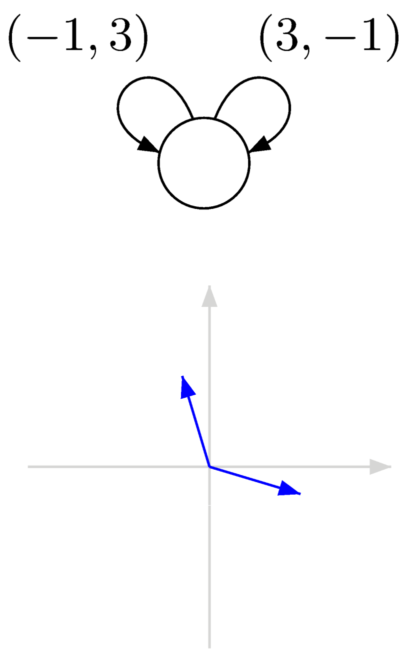
Hyperplane Separation





Hyperplane Separation



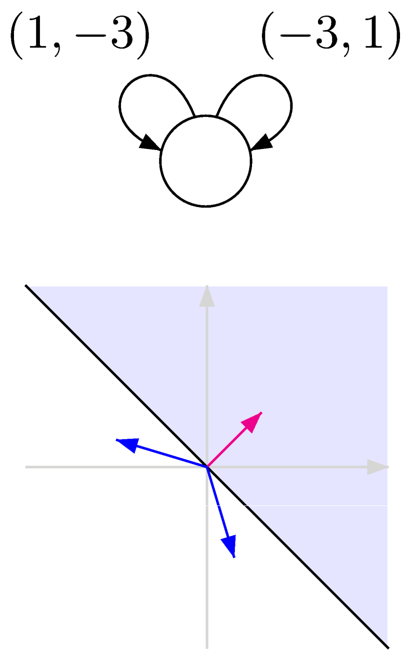


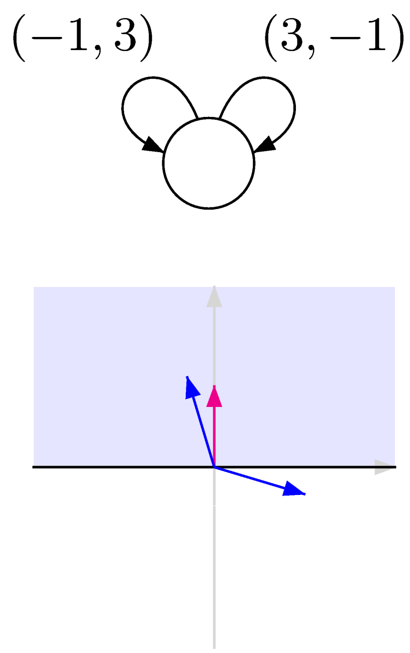
Player 1 cannot ensure $MP_\lambda \geq 0$ for some λ

$\Leftrightarrow$

Player 1 loses the multi-dimension game

Hyperplane Separation



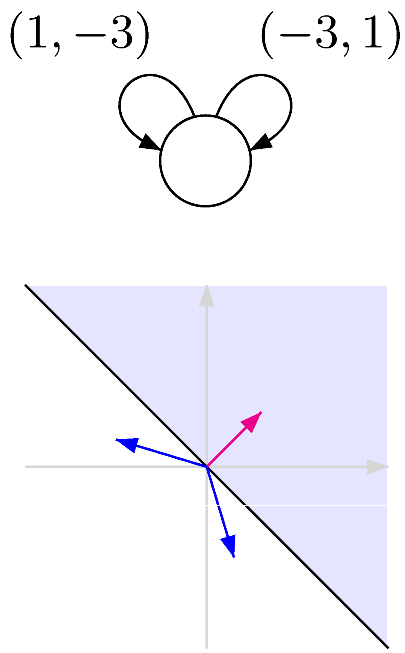


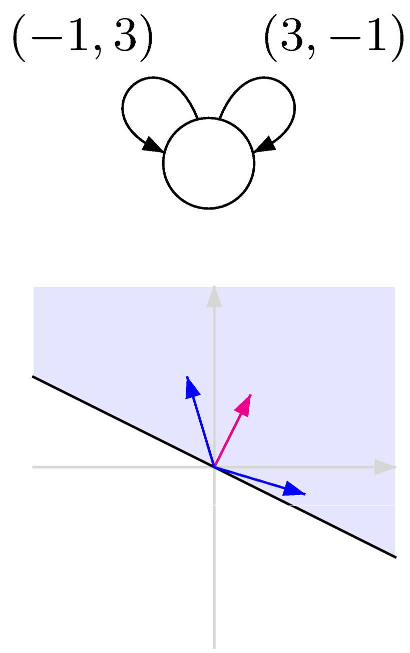
Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Hyperplane Separation



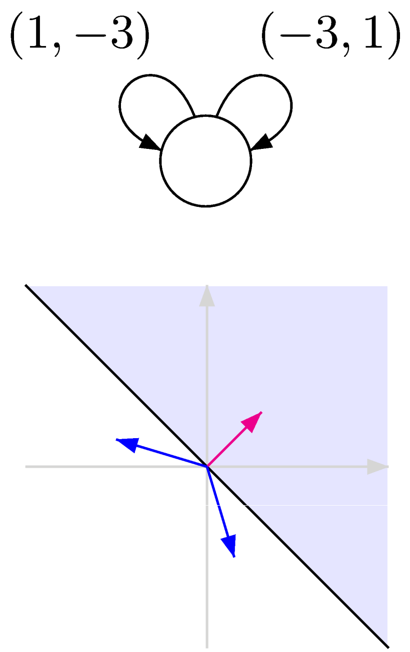


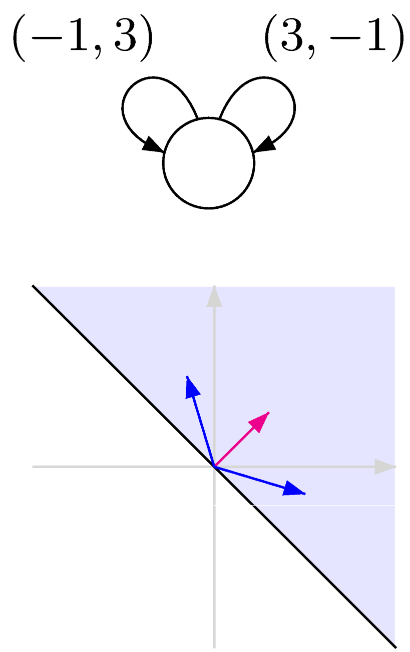
Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Hyperplane Separation



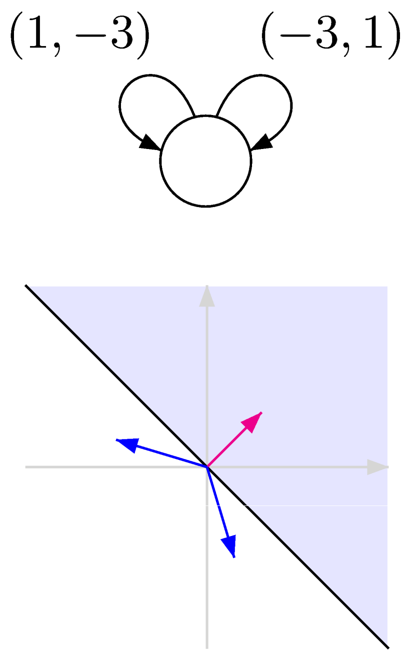


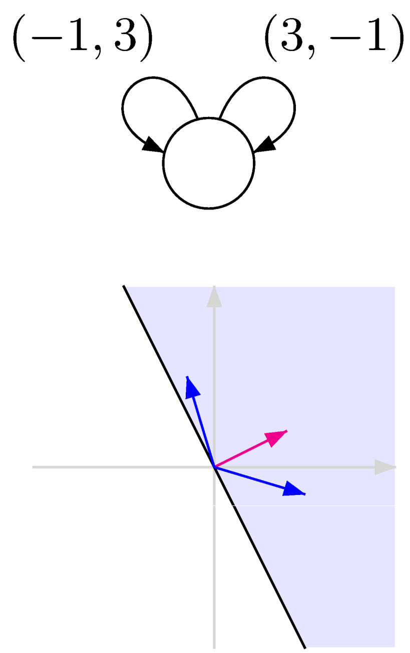
Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Hyperplane Separation



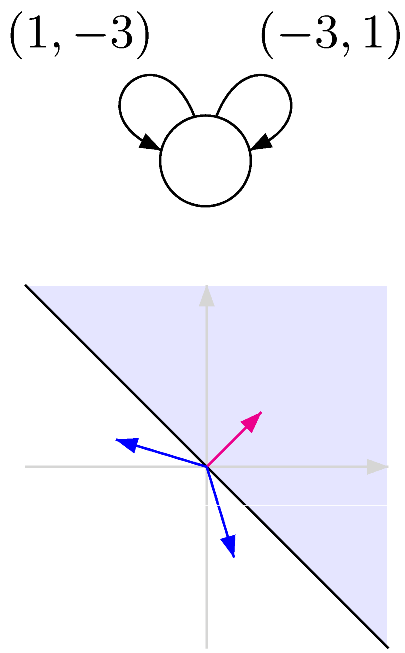


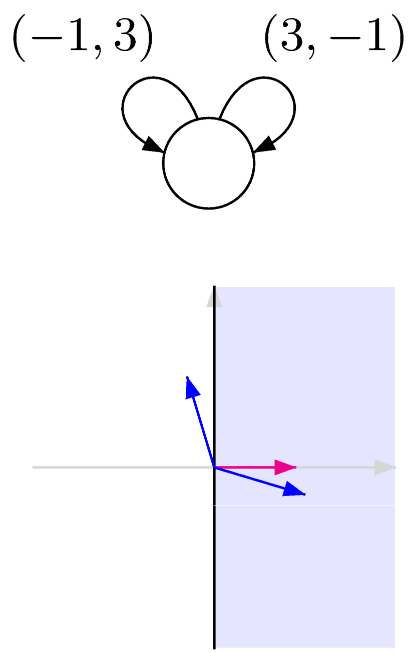
Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Hyperplane Separation





Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Hyperplane Separation

- Multi-dimension mean-payoff (liminf): coNP-complete
- Naive algorithm: exponential in number of states
- Hyperplane separation: reduction to single-dimension mean-payoff games

Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$ $\Leftrightarrow$ Player 1 wins the multi-dimension game

In fact, it is sufficient for player 1 to win for all $\lambda \in \{0,\ldots,\underbrace{(d \cdot n \cdot W)^{d+1}}_{M}\}^d$

Fixpoint algorithm:

- remove states if losing for some λ
- remove attractor (for player 2) of losing states

Solving $O(n \cdot M^d)$ mean-payoff games in $O(n \cdot m \cdot M)$

$O(n^2 \cdot m \cdot M^{d+1})$

Conclusion

Multiple dimensions of mean-payoff games

- Reachability game
- Energy game
- Cycle-forming game

Multi-dimension mean-payoff games

Memoryless proofs

Other directions: parity condition, stochasticity, imperfect information

Credits

- Energy/Mean-Payoff Games is joint work with *Lubos Brim, Jakub Chaloupka, Raffaela Gentilini, Jean-Francois Raskin*.
- Multi-dimension Games is joint work with *Krishnendu Chatterjee, Jean-Francois Raskin, Alexander Rabinovich, Yaron Velner*.
- Window games is joint work with *Krishnendu Chatterjee, Michael Randour, Jean-Francois Raskin*.

Other important contributions:

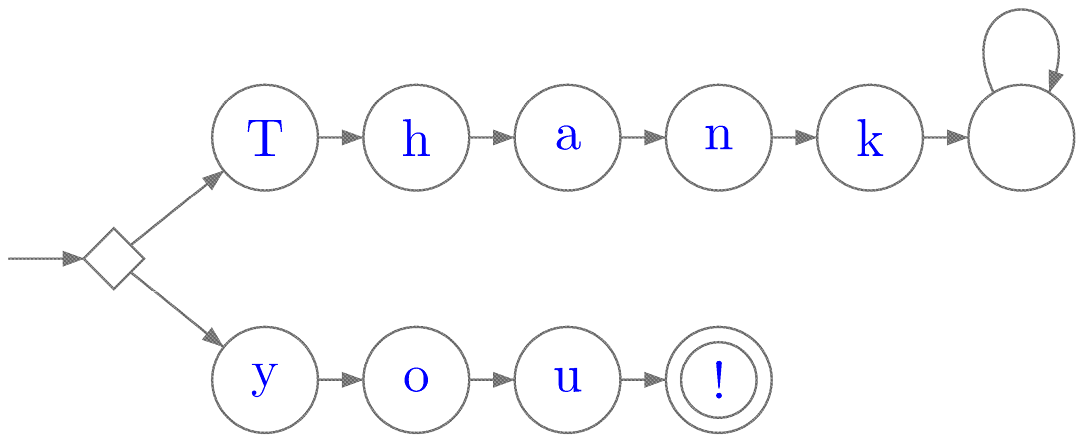
[BFLMS08] Bouyer, Fahrenberg, Larsen, Markey, Srba. ***Infinite Runs in weighted timed automata with energy constraints*****. FORMATS'08.**

[BJK10] Brazdil, Jancar, Kucera. ***Reachability Games on extended vector addition systems with states*****. ICALP'10.**

[CV14] Chatterjee, Velner. ***Hyperplane Separation Technique for Multidimensional Mean-Payoff Games*****. FoSSaCS'14.**

[Kop06] Kopczynski. ***Half-Positional Determinacy of Infinite Games*****. ICALP'06.**

[KS88] Kosaraju, Sullivan. ***Detecting cycles in dynamic graphs in polynomial time.*** **STOC'88.**


T
h
a
n
k
y
o
u
!

The end

Thank you !

Questions ?