Generalized Energy and Mean-Payoff Games

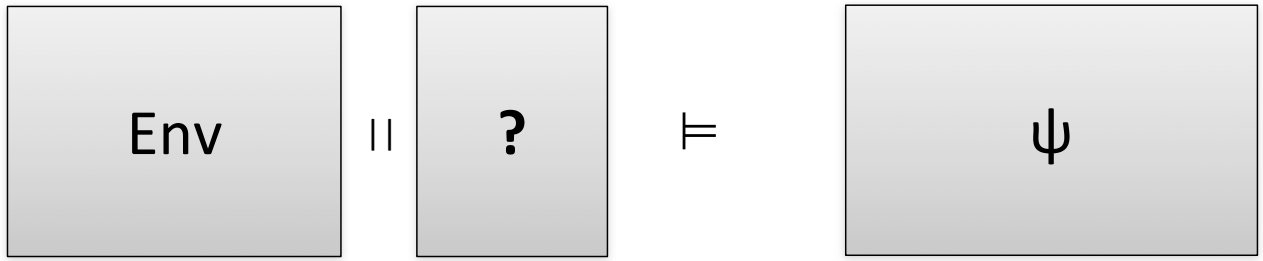
Jean-François Raskin
Computer Science
Université Libre de Bruxelles

Based on joint works with K. Chatterjee[1], L. Doyen[2], T. Henzinger[1], M. Randour[3]

(1) IST Austria - (2) LSV, Cachan - (3) CS, UMons

Games for Synthesis (of Reactive Systems)

☞ support the design process with **automatic synthesis**



- Sys is constructed by an algorithm
- Sys is **correct** by construction
- Underlying theory: **2-player zero-sum games**
- Env is **adversarial** (worst-case assumption)

Correct Sys = Winning strategy

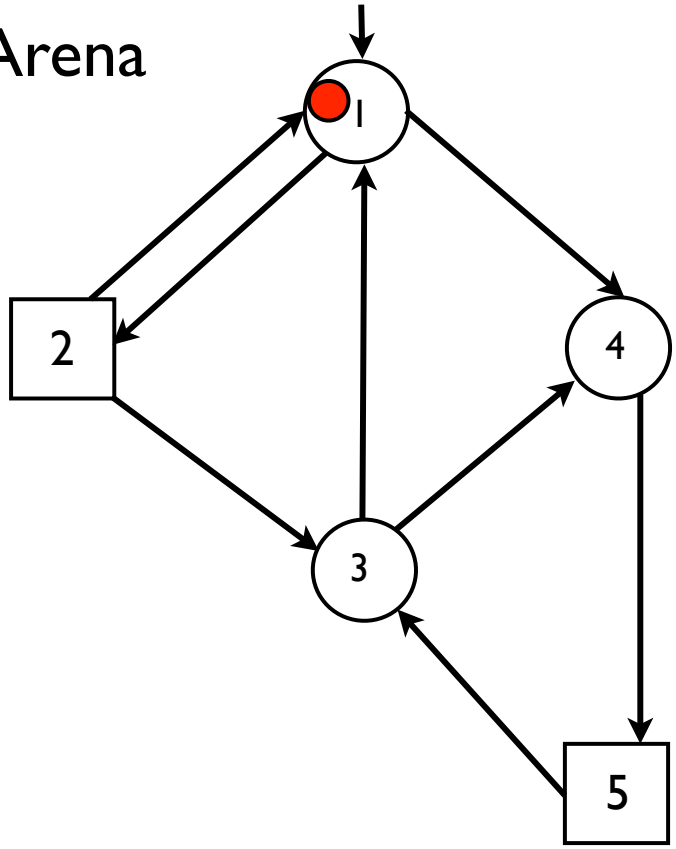
Plan of the talk

- **2-player zero-sum games on graphs**
  - Arenas and winning conditions
  - Strategies
  - Typical questions we want to answer
- **Multi-energy and mean-payoff parity games**
  - Memory requirements
  - Complexity
  - Trading memory for randomness

2-player zero sum games on graphs

Games played on graphs

Arena



Start with pebble on initial vertex

Repeat for ∞-many rounds:

➡ The player that owns the current vertex moves the pebble to an adjacent vertex

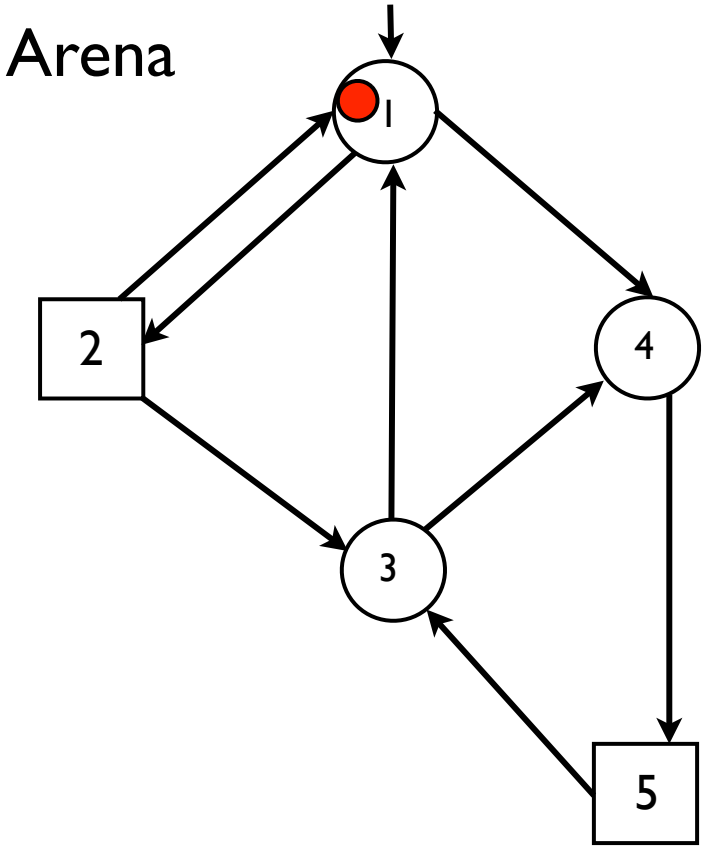
Interaction → **infinite path** = a **play**

Winning condition $W \subseteq V^\omega$ for Player 1
=set of plays
$V^\omega \setminus W$ is winning for Player 2 (zero-sum).

Outcome=Play=1 - 4 - 5 - 3 - 1 - 2 - ...

(Player 1) **Winning condition**: cross "2" and "5" ∞-often

Games played on graphs



Start with pebble on initial vertex

Repeat for ∞-many rounds:

➡ The player that owns the current vertex moves the pebble to an adjacent vertex

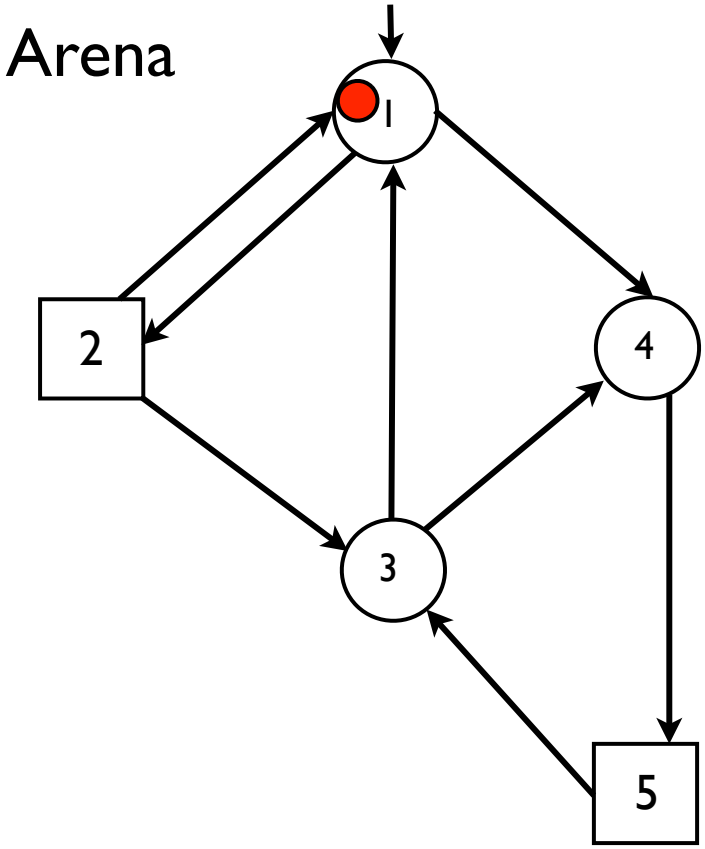
Interaction → **infinite path** = a **play**

Winning condition $W \subseteq V^\omega$ for Player 1
=set of plays
$V^\omega \setminus W$ is winning for Player 2 (zero-sum).

Outcome=Play=1 - 4 - 5 - 3 - 1 - 2 - ...

(Player 1) **Winning condition**: cross "2" and "5" ∞-often

Games played on graphs



Start with pebble on initial vertex

Repeat for ∞-many rounds:

➡ The player that owns the current vertex moves the pebble to an adjacent vertex

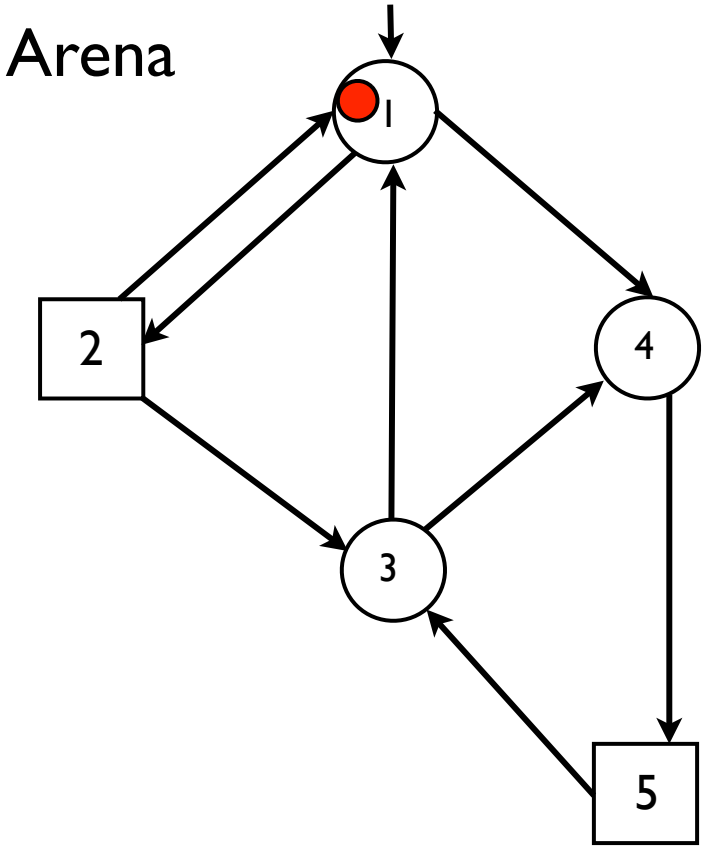
Interaction → **infinite path** = a **play**

Winning condition $W \subseteq V^\omega$ for Player 1
=set of plays
$V^\omega \setminus W$ is winning for Player 2 (zero-sum).

Outcome=Play=1 - 4 - 5 - 3 - 1 - 2 - ...

(Player 1) **Winning condition**: cross "2" and "5" ∞-often

Games played on graphs



Start with pebble on initial vertex

Repeat for ∞-many rounds:

➡ The player that owns the current vertex moves the pebble to an adjacent vertex

Interaction → **infinite path** = a **play**

Winning condition $W \subseteq V^\omega$ for Player 1
=set of plays
$V^\omega \setminus W$ is winning for Player 2 (zero-sum).

Outcome=Play=1 - 4 - 5 - 3 - 1 - 2 - ...

Typical type of winning condition: ω-**regular** sets.

Game

- Arena: bi-partite graph with initial vertex
- A winning condition: a set of plays - often regular

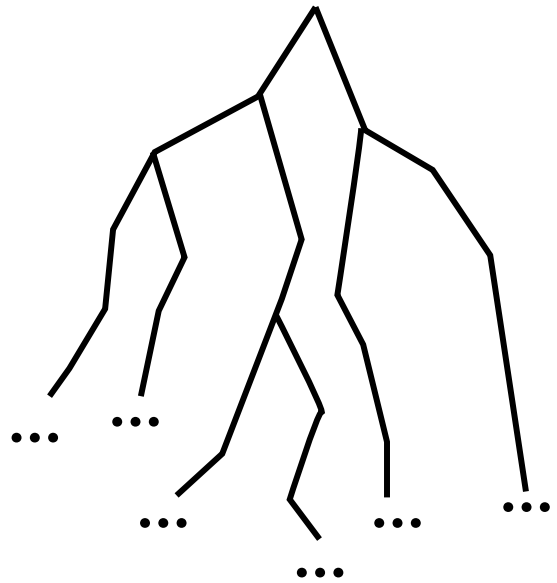
Game

- Arena: bi-partite graph with initial vertex
- A winning condition: a set of plays - often regular

- How do Players play ?

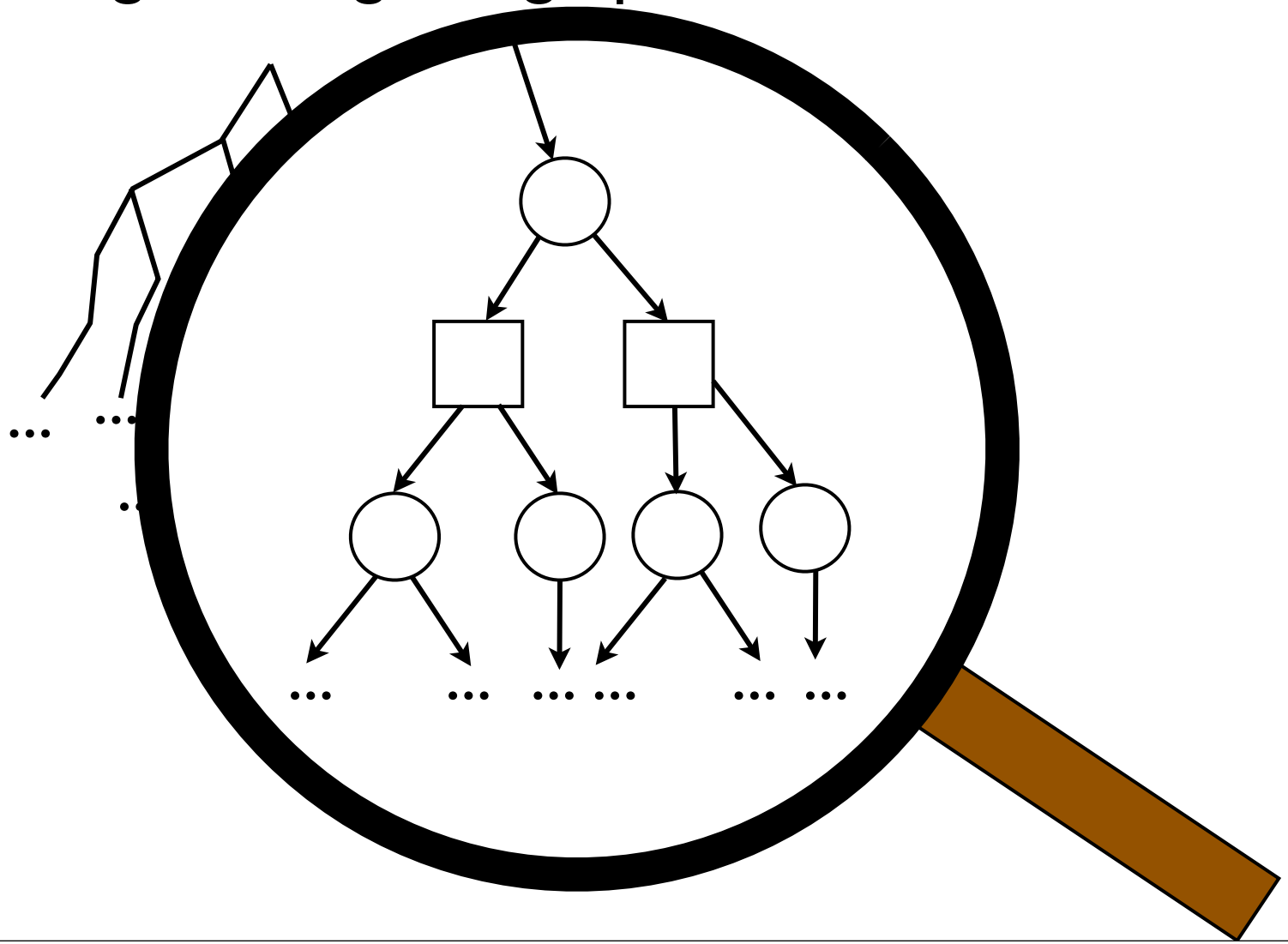
Strategies

Unfolding of the game graph



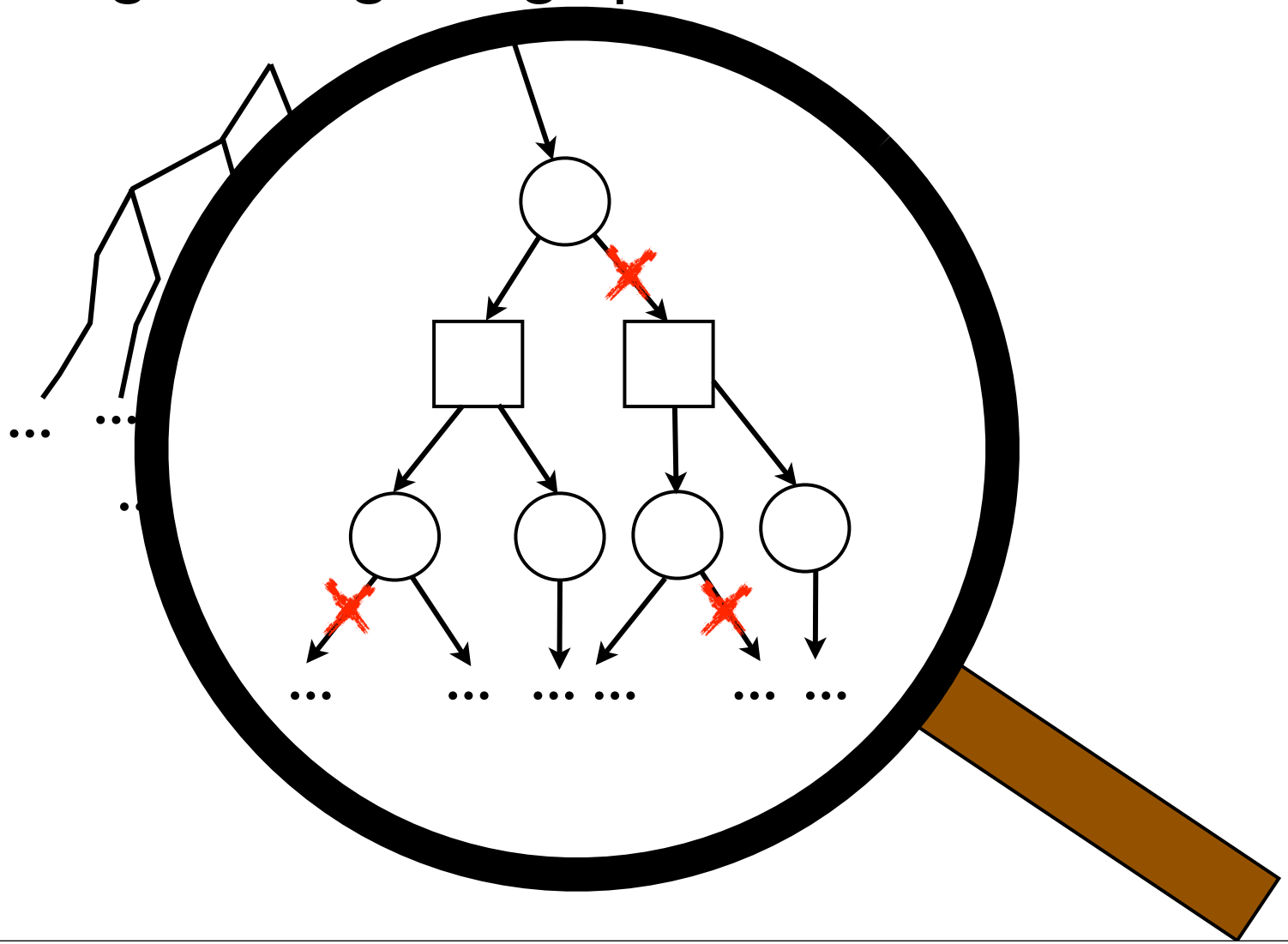
Strategies

Unfolding of the game graph

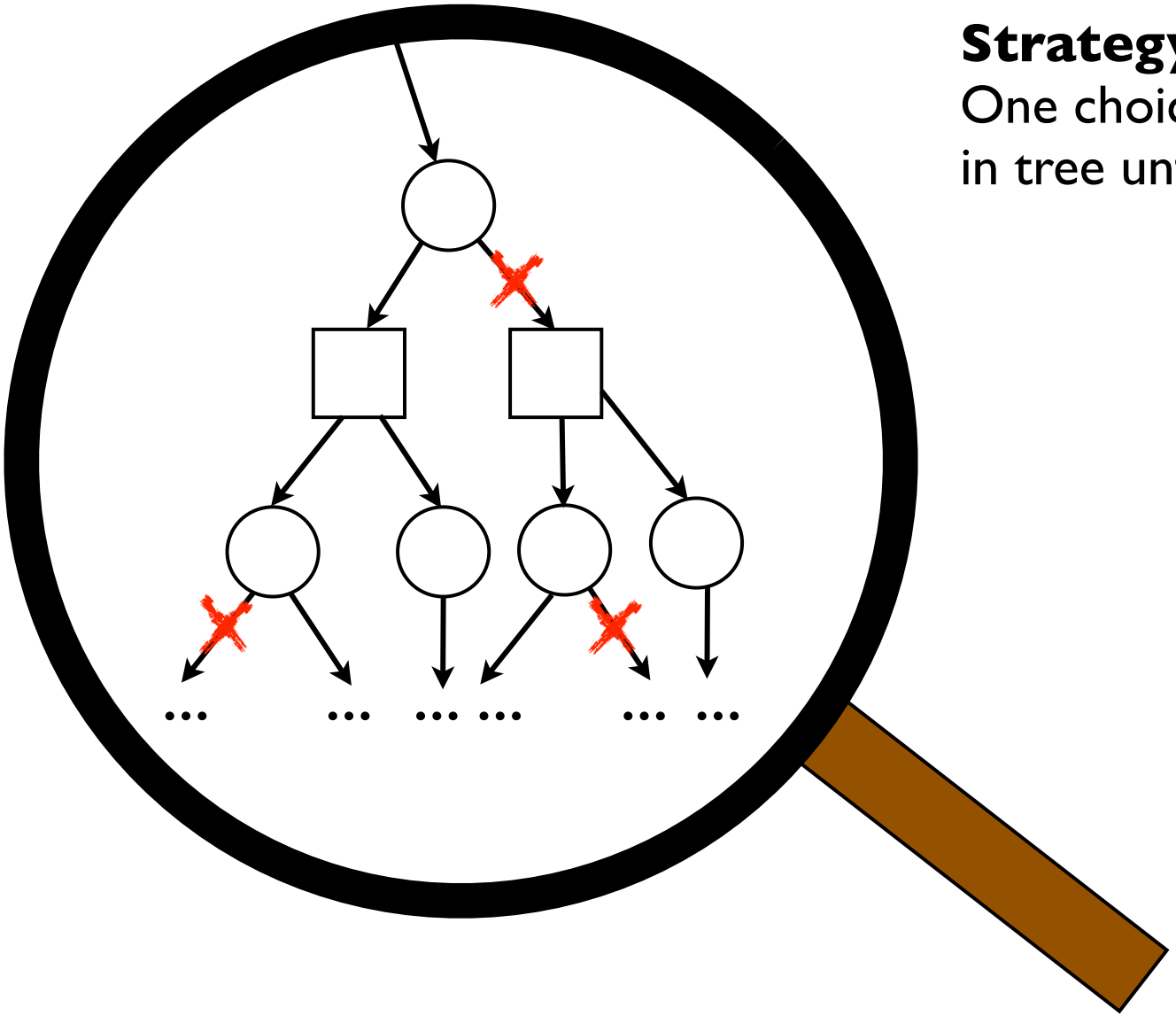


Strategies

Unfolding of the game graph

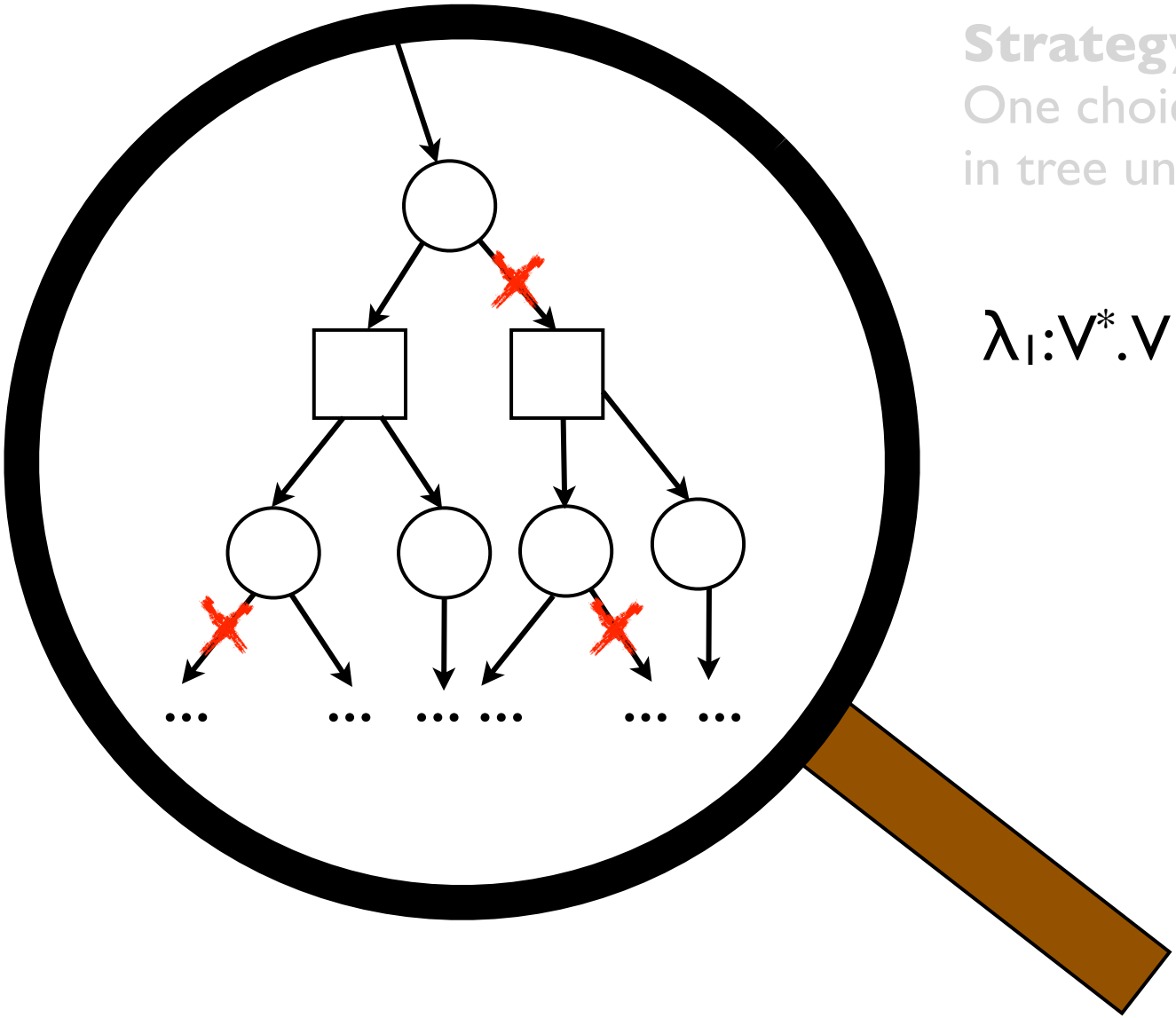


Strategies



Strategy for Player 1 =
One choice in each node of Player 1 in tree unfolding

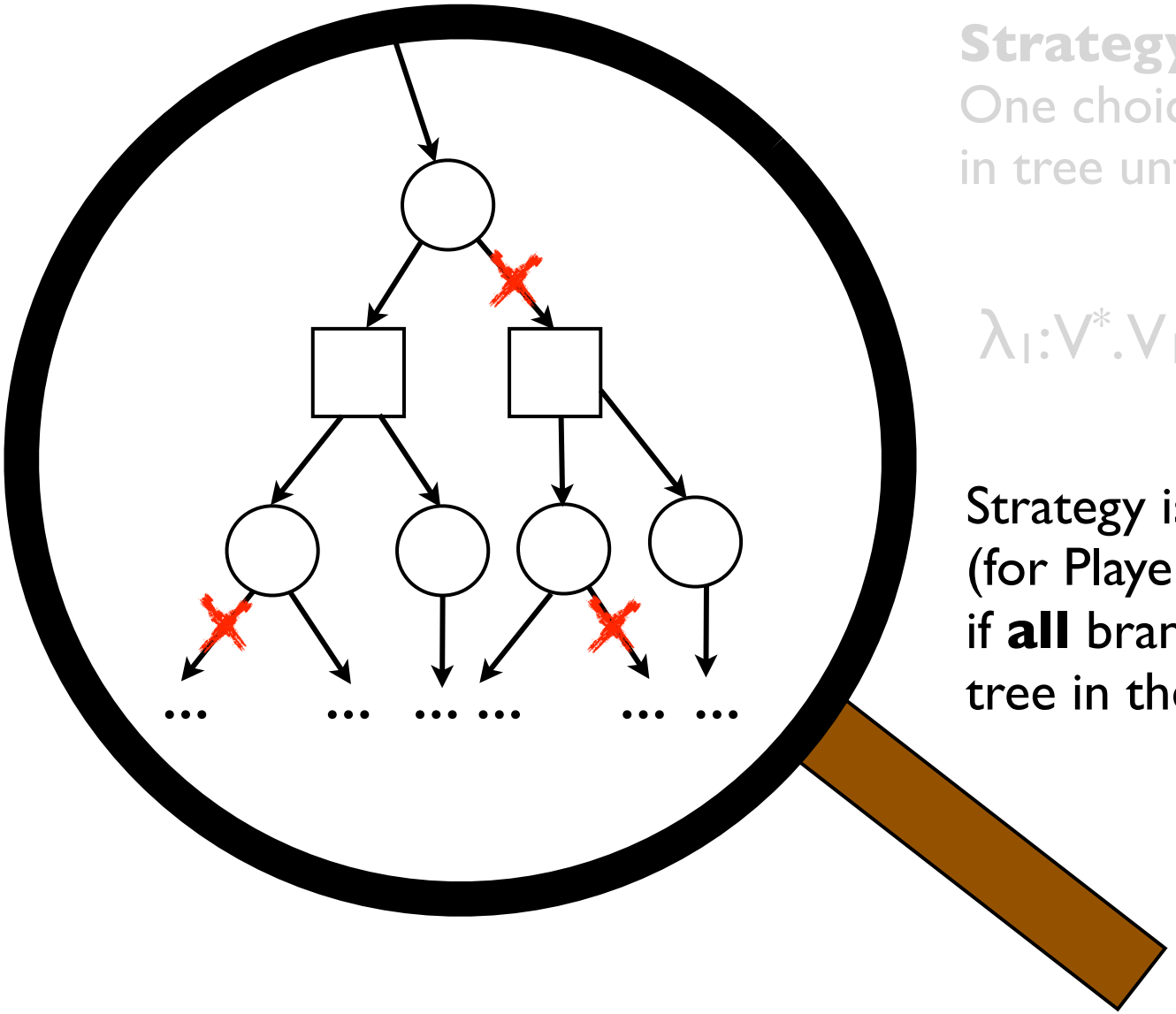
Strategies



Strategy for Player I =
One choice in each node of Player I in tree unfolding

$\lambda_I : V^*.V_I \rightarrow \text{edge}$

Strategies



Strategy for Player 1 =
One choice in each node of Player 1 in tree unfolding

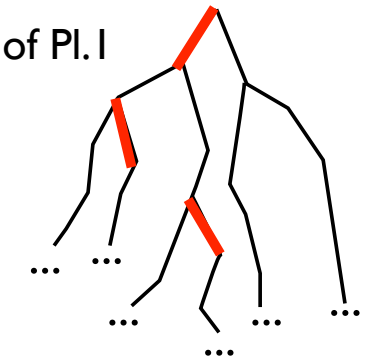
$\lambda_1 : V^*.V_1 \rightarrow \text{edge}$

Strategy is **winning**
(for Player 1),
if **all** branches of the resulting tree in the winning condition

Types of strategies

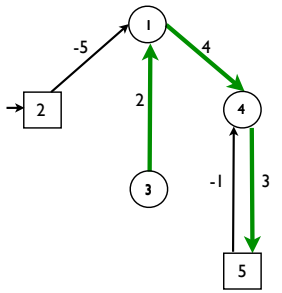
(Player 1) strategy:

$\lambda_1 : V^*.V_1 \rightarrow$ edge.
Σ_1=set of strategies of Pl.1



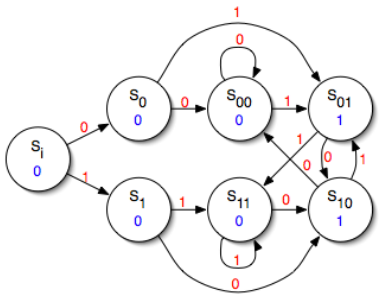
Memoryless strategy:

$\lambda_{1,\mathbf{m}} : V_1 \rightarrow$ edge.
$\Sigma_{1,\mathbf{m}}$=set of memoryless strategies of Player1



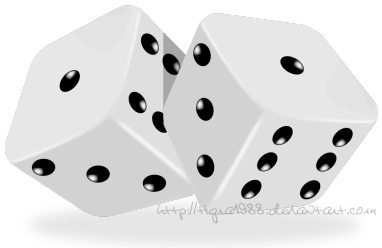
Finite-memory strategy:

$\lambda_{1,\mathbf{f}} : V^*.V_1 \rightarrow$ edge but regular (Moore machine)
$\Sigma_{1,\mathbf{f}}$=set of finite memory strategies of Player 1



Randomized strategy:

$\lambda_{1,\mathbf{m}} : V^*.V_1 \rightarrow$ **Dist**(edge).
$\Sigma_{1,\mathbf{m}}$=set of randomized strategies of Player1



Typical questions

- **Types of objectives** (winning conditions) that ensures that one player wins or the other **- determinacy** ?
  General result: Borel determinacy [Martin73].
- **Types of strategies** necessary to win ?
  - How much memory is needed ?
  - Do we need randomization ?
    (this is necessary for example when considering concurrent moves or imperfect information)
- **Complexity** of deciding the winner ...
- **Efficient algorithms** to construct winning strategies ...

Need for richer models

From

2-player games	Sys **versus** Env
Zero sum games	Env is **antagonist**
Boolean objectives	Winning **or** losing
Perfect information	Sys observes Env **perfectly**

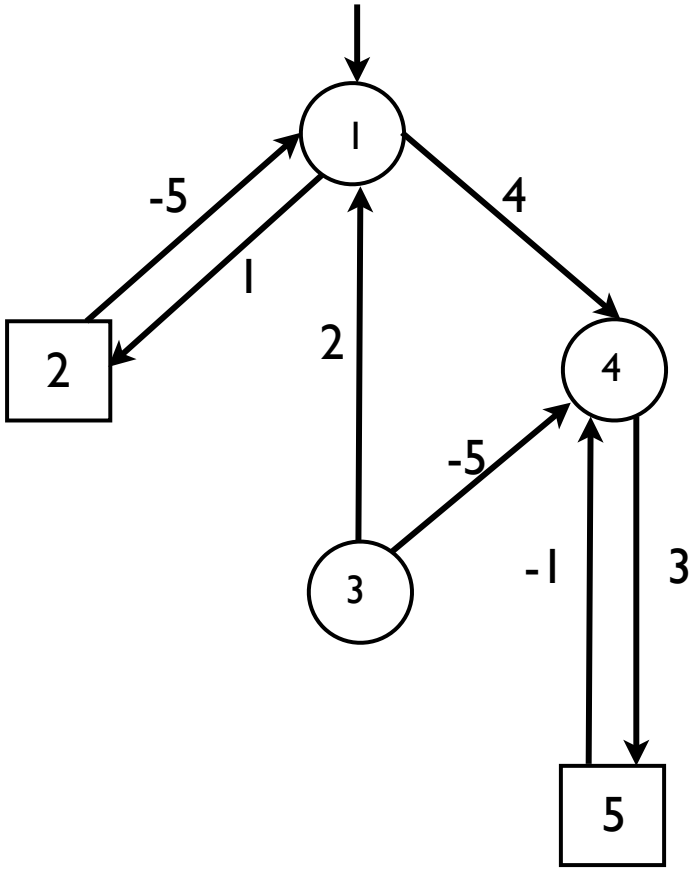
To

N-player games	Modern systems are made of **components**
Non-zero sum	Components have **their own objectives**
Quantitative objectives	Need to **compare** different solutions
Imperfect information	Components have a **partial view** of the system

Plan of the talk

- 2-player zero-sum games on graphs
  - Arena and winning condition
  - Strategies
  - Typical questions we want to answer
- **Multi-energy and mean-payoff (parity) games**
  - Memory requirements
  - Complexity
  - Trading memory for randomness

Mean-Payoff Games [EM79]



: positions of **maximizer=system**

: positions of **minimizer=environment**

Edges are labelled with rewards

(1,4) (4,5) (5,4) ... (4,5) (5,4) .... =play
4 3 -1 3 -1 ....

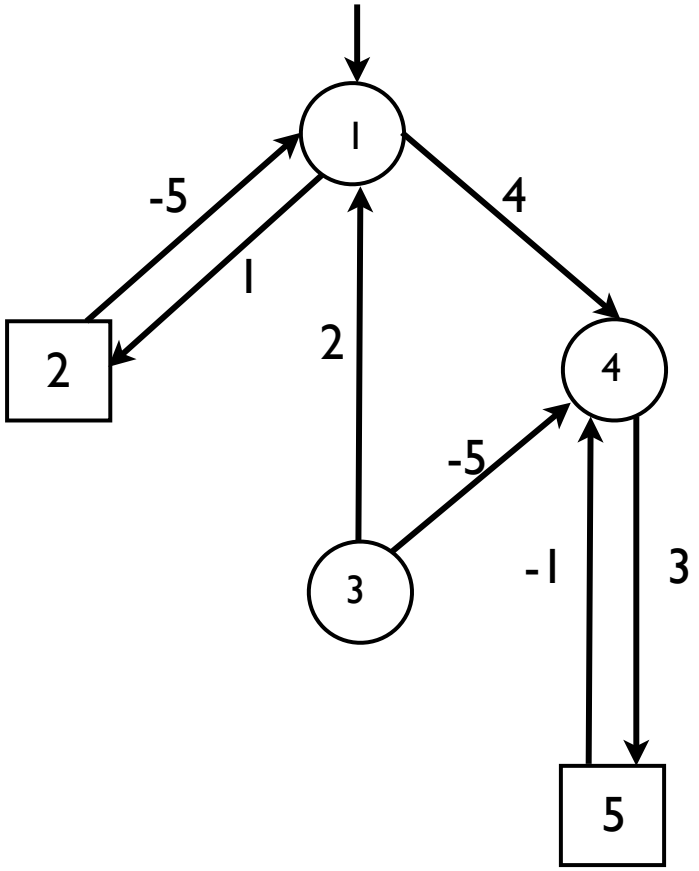
= **Lim Sup**$_{n\to+\infty}$ $\Sigma_{i=1,i=n}$ r_i / n = 1.

= **MP**((1,4) (4,5) (5,4) ... (4,5) (5,4) ....)

Win={ play | MP(play)≥**c**}

Note: **not** ω-regular.

Mean-Payoff Games [EM79]



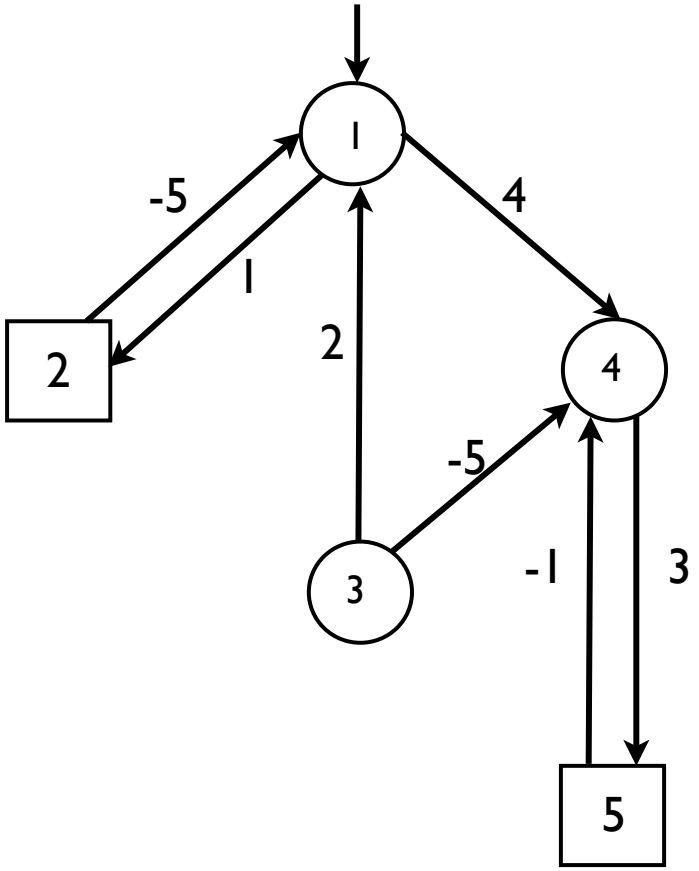
Outcome(s,λ_1,λ_2)=infinite sequence of edges when players play λ_1 and λ_2 from s.

The value of the game in s under λ_1 and λ_2 is

MP(**Outcome**(s,λ_1,λ_2)).

The decision problem for mean-payoff games asks
given an initial state **s** if Maximizer has **a strategy** to ensure a mean-payoff greater or equal to a integer value **v** from **s** no matter what Minimizer plays.

Energy Games [CdAHS03,BFLMS08]



: positions of **maximizer**

: positions of **minimizer**

Edges are labelled with energy consumptions or energy gains.

Strategies are defined as for MPGs.

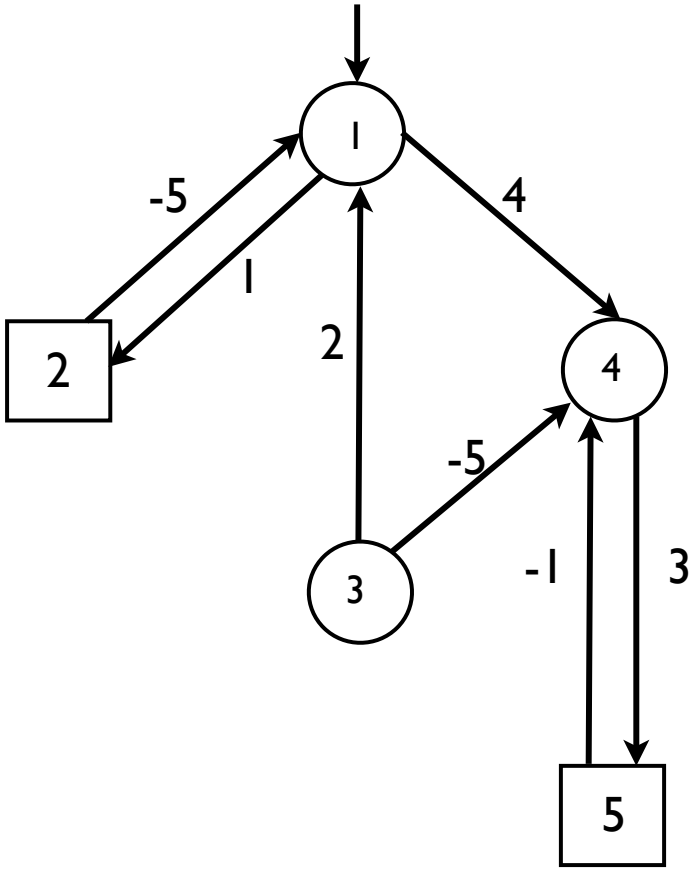
Initial energy level : 7

Play : (1,2) (2,1) (1,4) (4,5) (5,4) (4,5) (5,4) ...

EL : 7 8 3 7 10 9 10 9 ...

$\models$ □ **EL** ≥ 0

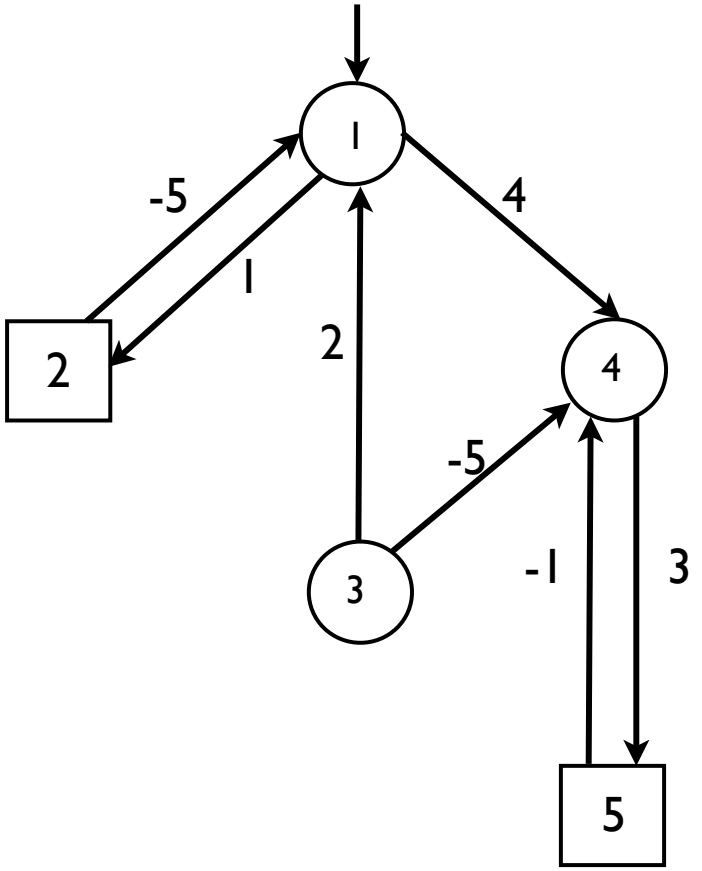
Energy Games [CdAHS03,BFLMS08]



The **decision problem for energy games** asks given an initial state **s** if there exist

- ✶ an **initial energy level**
- ✶ and λ_1 **for Maximizer** to maintain a positive energy level at all time ($\Box$ **EL**≥ 0), no matter what Minimizer plays.

Energy Games [CdAHS03,BFLMS08]

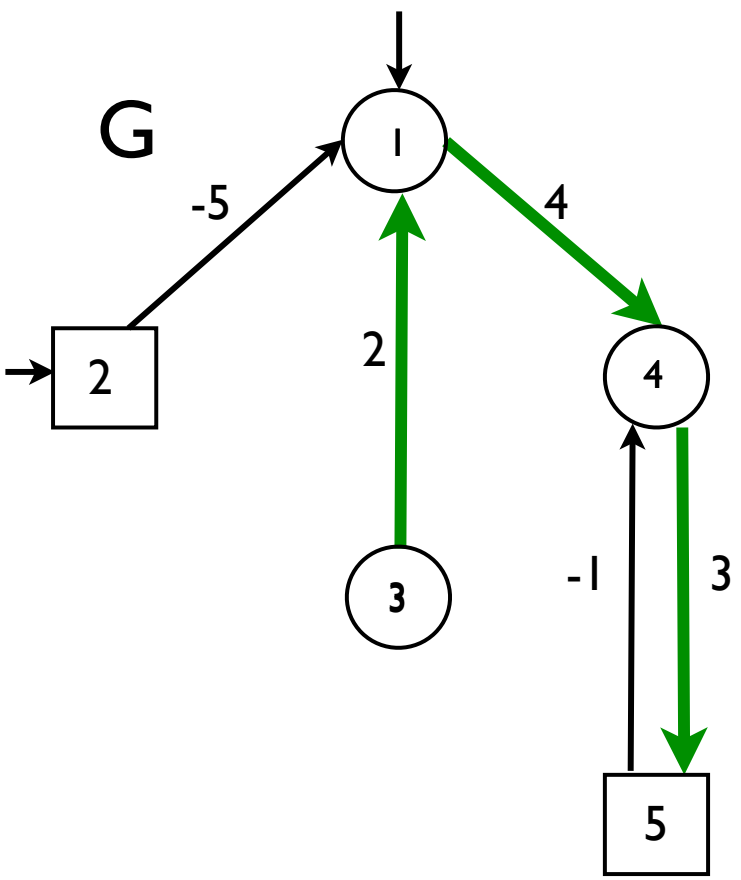


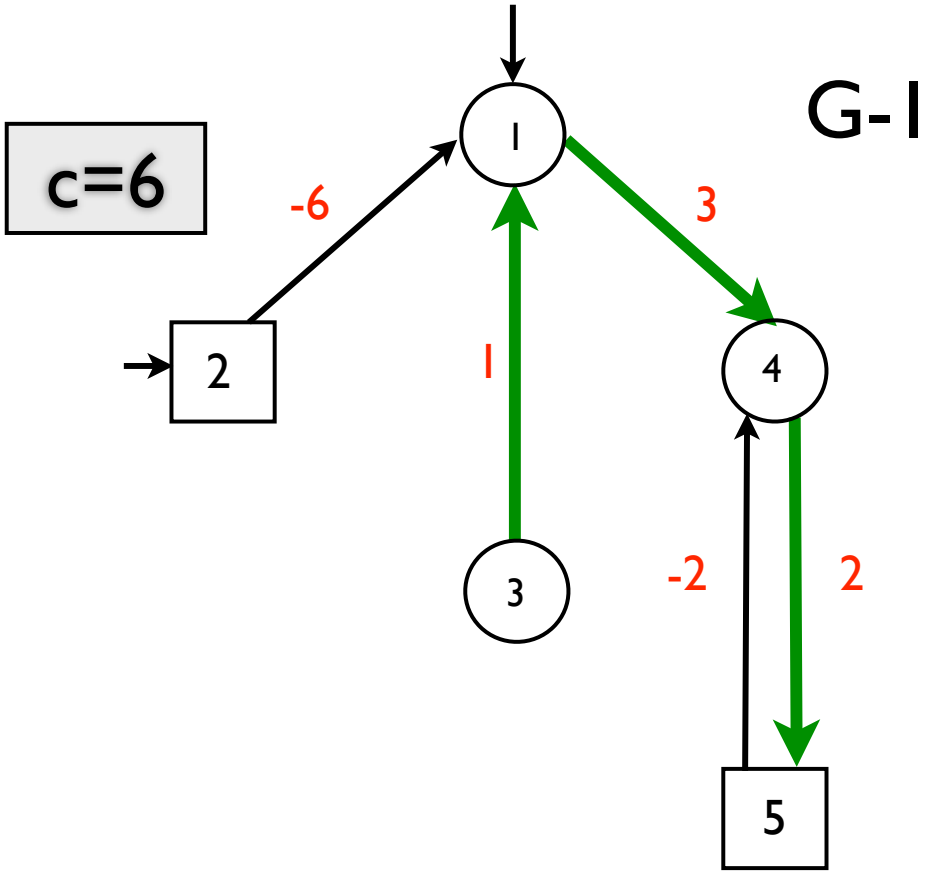
The **decision problem for energy games** asks given an initial state **s** if there exist

- ✶ an **initial energy level**
- ✶ and λ_1 **for Maximizer** to maintain a positive energy level at all time ($\Box$ **EL**≥ 0), no matter what Minimizer plays.

Thm. Maximizer enforces value at least v in MPG G
iff Maximizer has a winning strategy in the ENG G-v.

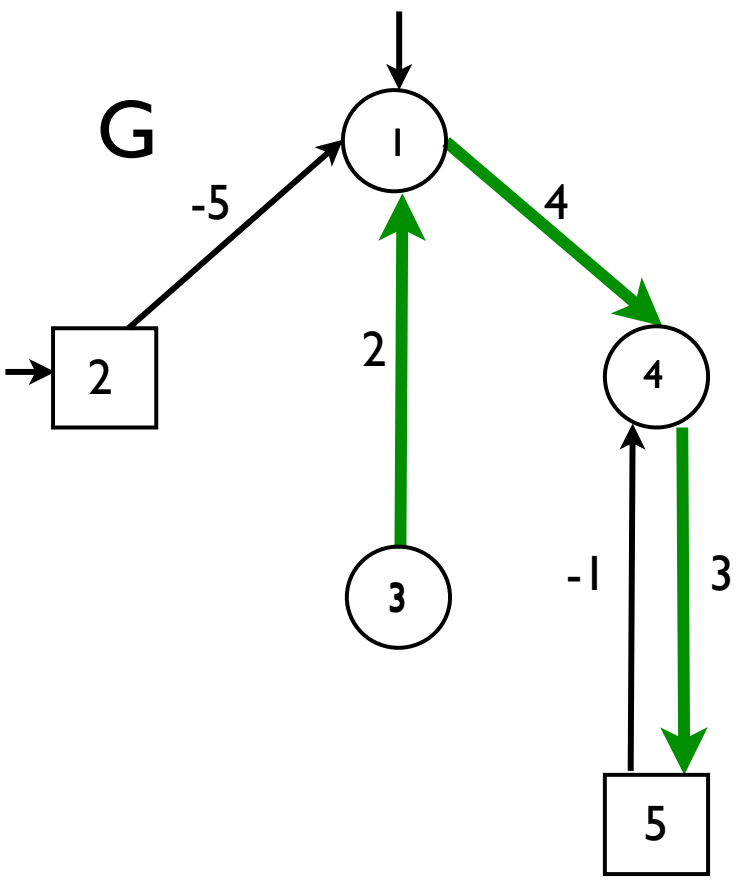
MPGs and EGs [BFLMS08]

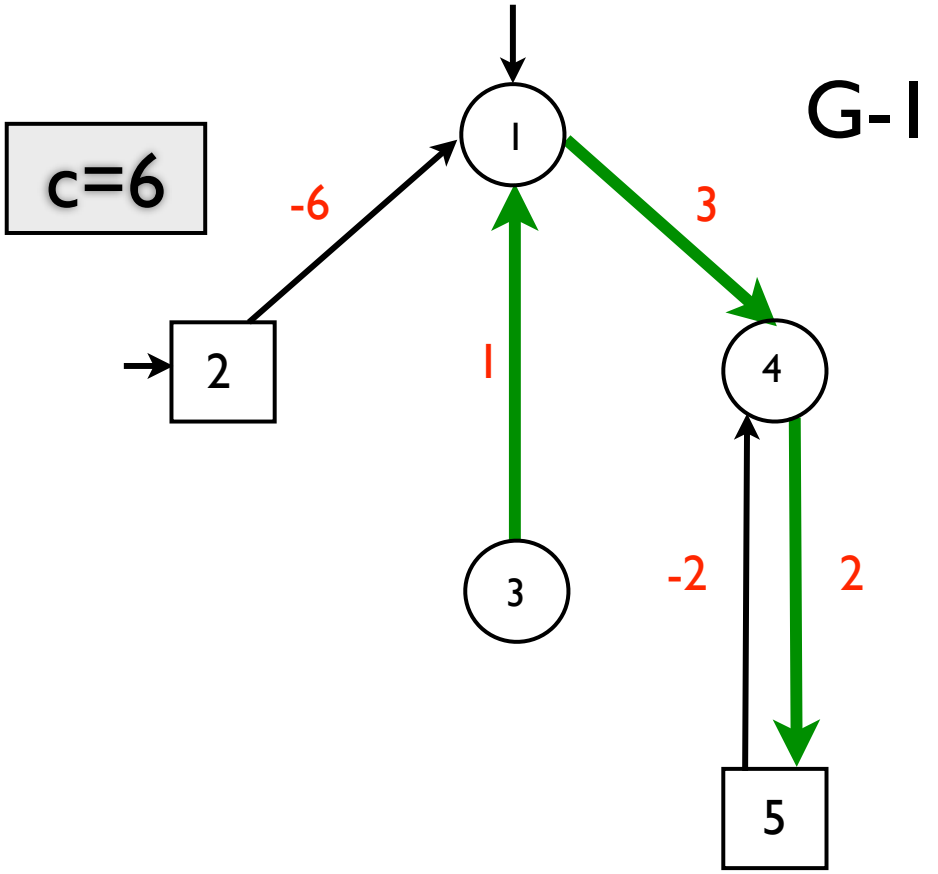




Maximizer ensure value 1 in MPG **iff** Maximizer win EG G-1.

MPGs and EGs [BFLMS08]





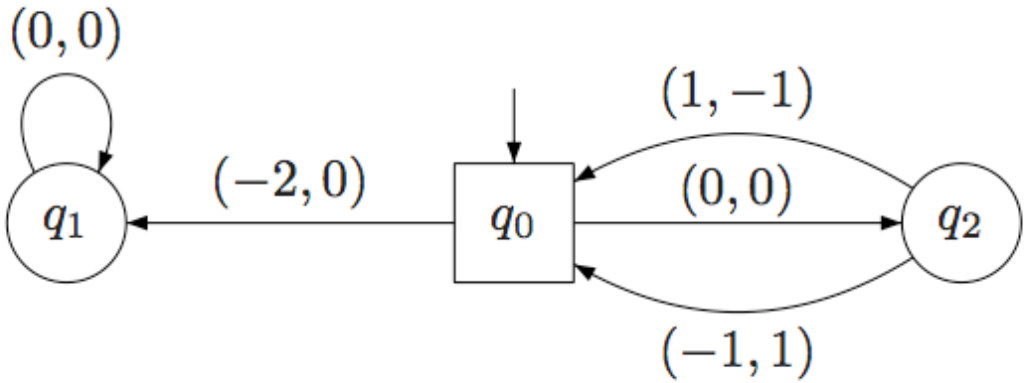
MPGs and EGs are **memoryless** determined and in **NP∩coNP**

Generalization - Motivations

We study here **multi-dimensional** extensions of energy and mean-payoff games:

- Useful for modeling control problems for embedded systems with **multiple** quantitative aspects;
- We study "under **which sets of strategies** are those games determined ?"
- We settle open **complexity** questions.

Generalized Energy Games (gEGs)



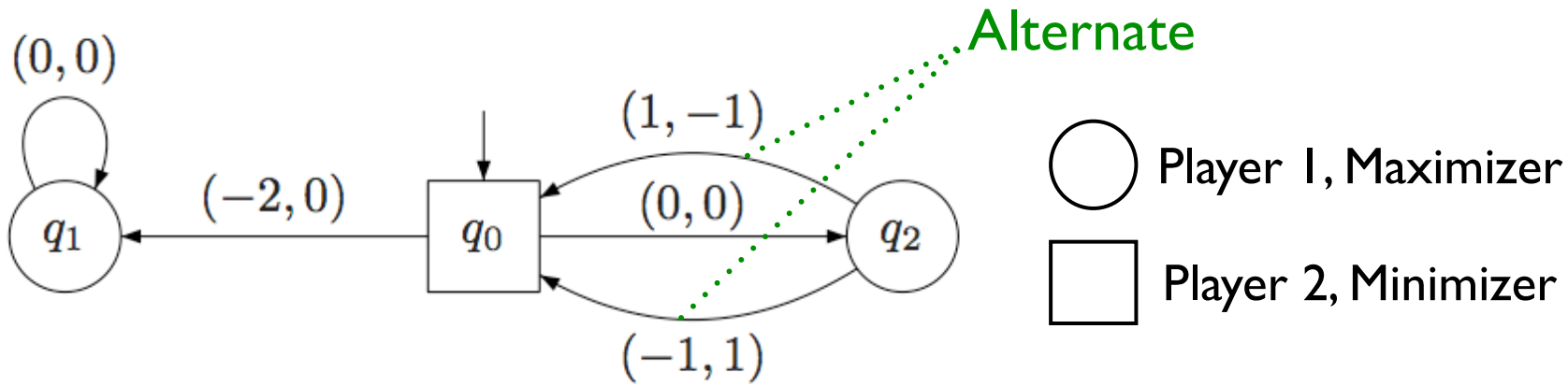
Player 1, Maximizer

Player 2, Minimizer

? $\exists(C_1,C_2)\in\mathbb{N}^2$ and λ_1 s. t. **Outcome**$(q_0,\lambda_1,(C_1,C_2)) \models \square$ $\mathbf{EL_1}\geq 0 \wedge \mathbf{EL_2}\geq 0$.

Memory

Generalized Energy Games (gEGs)



? $\exists(C_1,C_2)\in\mathbb{N}^2$ and λ_1 s. t. **Outcome**$(q_0,\lambda_1,(C_1,C_2)) \models \Box$ $\mathbf{EL_1}\geq 0 \wedge \mathbf{EL_2}\geq 0$.

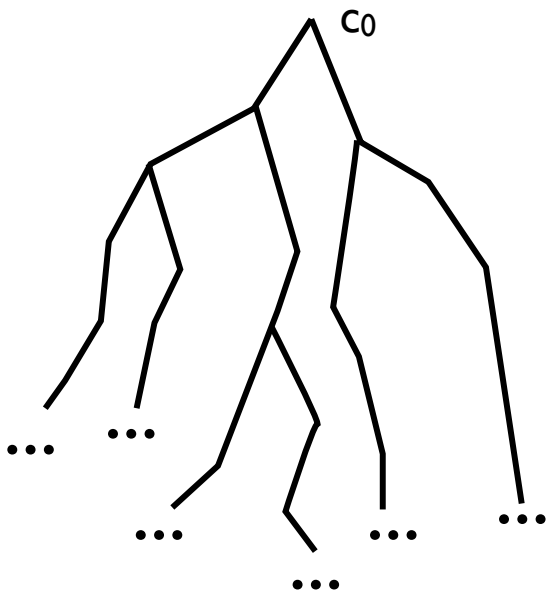
- For any $(C_1,C_2)\geq(2,1)$, Player 1 has a winning strategy.
- Player 1 needs memory ! How much ?

Player 1 - Finite Memory Strategies

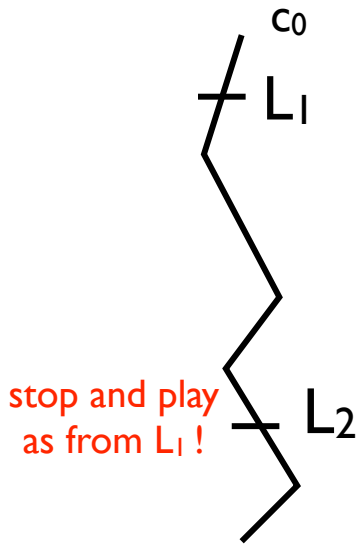
Lemma. Finite memory strategies are sufficient for Player 1 to win in gEGs.

Proof. First, remember that $(\mathbb{N}^k, \leq)$ is well-quasi ordered.

Let λ_1 be winning

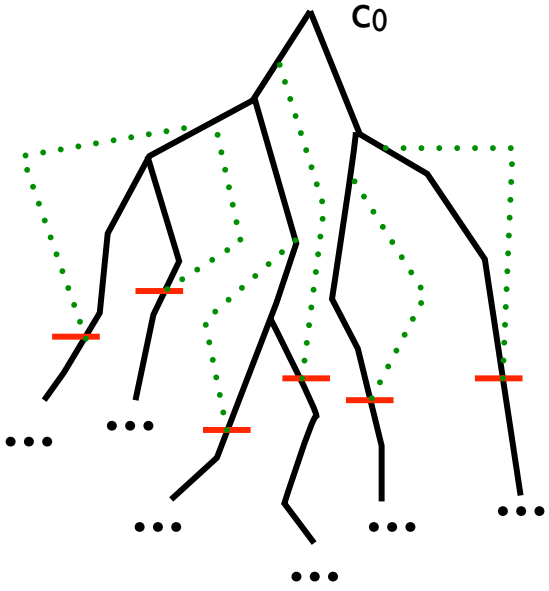


On each branch



With $L_1 \leq L_2$

Then λ'_1 is winning and finite memory



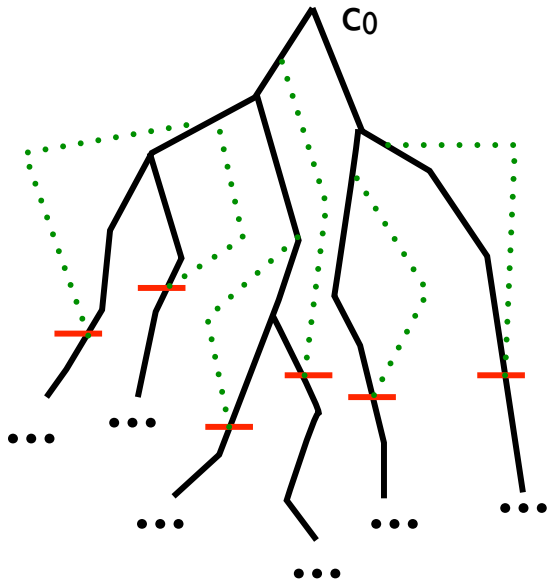
wqo+Koenig's lemma

Player 1 - Finite Memory Strategies

Lemma. Finite memory strategies are sufficient for Player 1 to win in gEGs.

Proof. First, remember that $(\mathbb{N}^k, \leq)$ is well-quasi ordered.

Then λ'_1 is winning and finite memory



wqo+Koenig's lemma

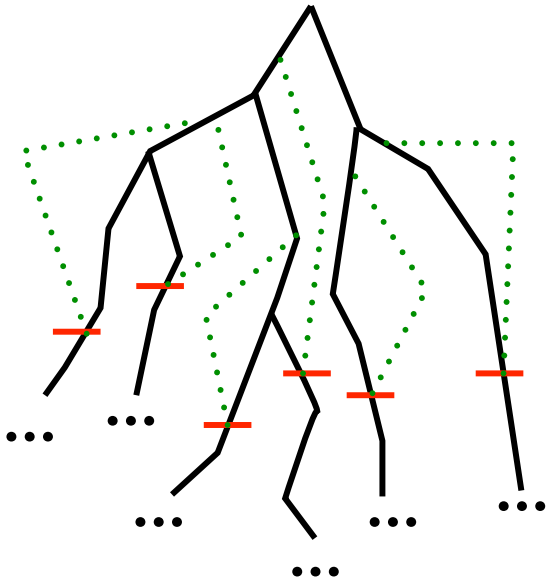
Finite tree=winning strategy:

① play according to the choices made in tree

② in leaf, go to ancestor with lower energy

Finite memory → Exponential memory

Then λ'_1 is winning and finite memory

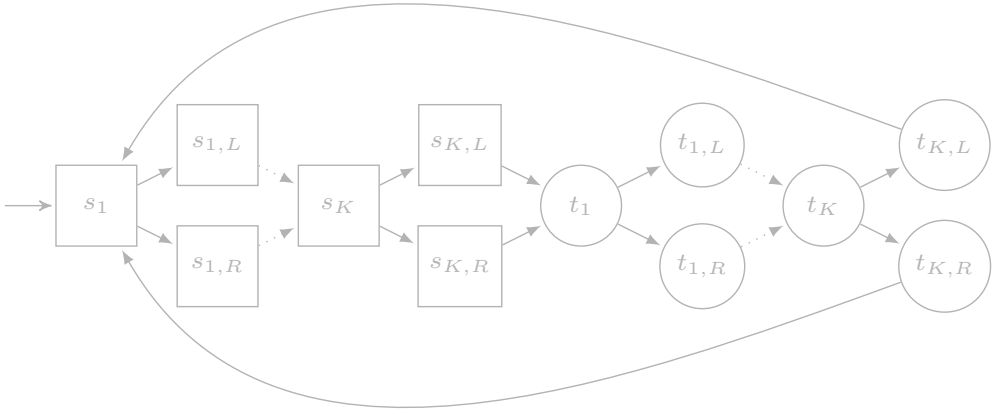


wqo+Koenig's lemma

① Exponential memory is sufficient.

- Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

② Exponential memory is needed

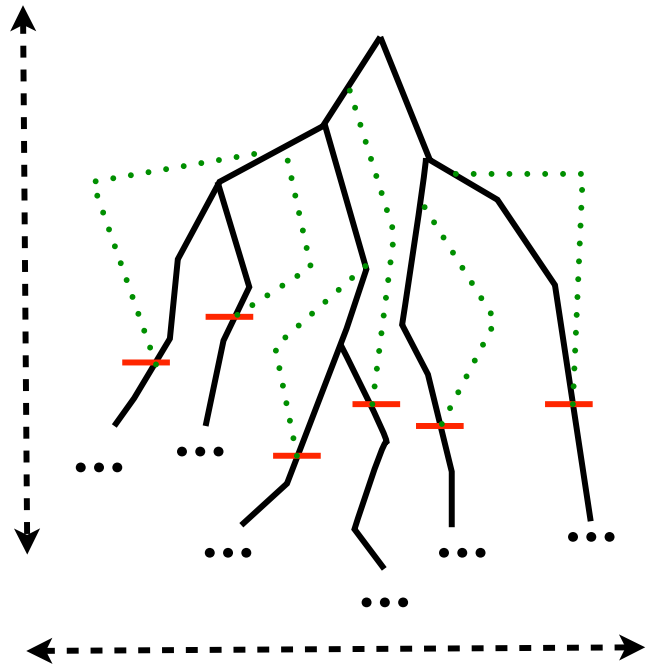


③ Leads to symbolic and incremental algorithms

Finite ponential memory

=**S**elf-**C**overing **T**ree [BJK10]

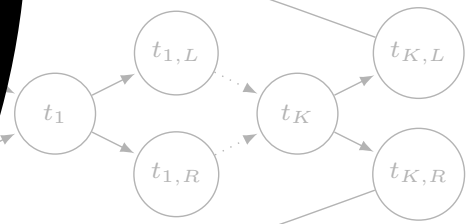
1Exp
[BJK10]
Arbitrary
weights:
2Exp



3Exp

icient.

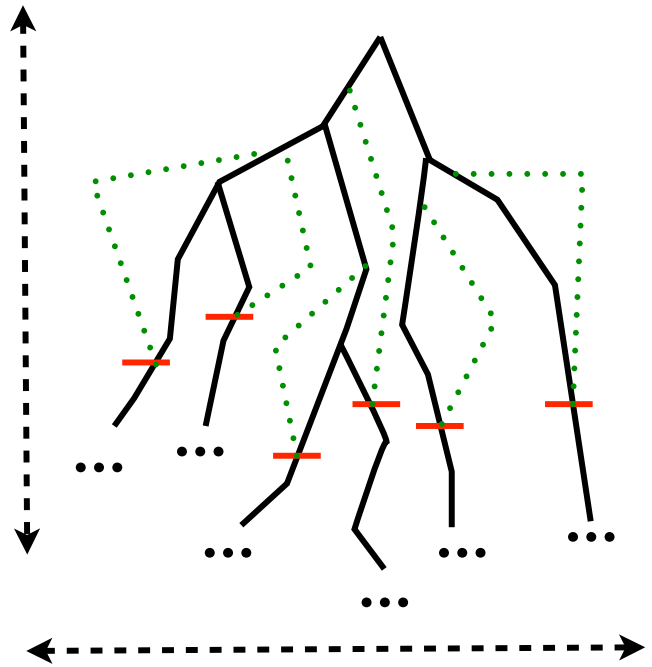
nics à la Rackoff
s of [BJK10]



hms

Finite ponential memory

=**S**elf-**C**overing **T**ree [BJK10]



1Exp [BJK10] Arbitrary weights: ~~2Exp~~

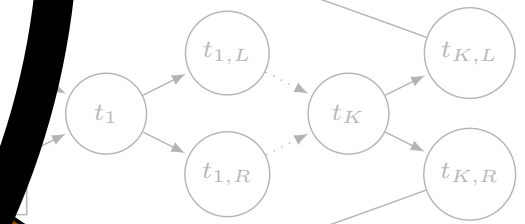
1Exp

~~3Exp~~ 1Exp

Depth: single exponential - encoding of arbitrary weights into {-1,0,1} does not add choices to the adversary.
Width: only energy level important (DAG).

ficient.

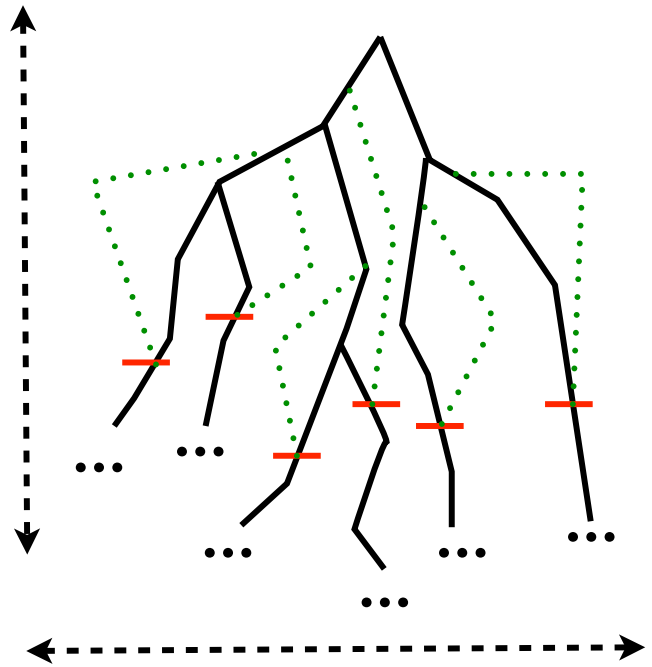
nics à la Rackoff
s of [BJK10]



hms

Finite ponential memory

=**S**elf-**C**overing **T**ree [BJK10]



1Exp [BJK10] Arbitrary weights: ~~2Exp~~

1Exp

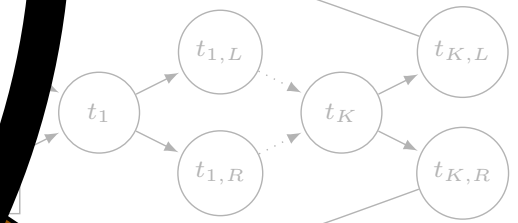
~~3Exp~~ 1Exp

Depth: single exponential - encoding of arbitrary weights into {-1,0,1} does not add choices to the adversary.
Width: only energy level important (DAG).

Works also with parity

ficient.

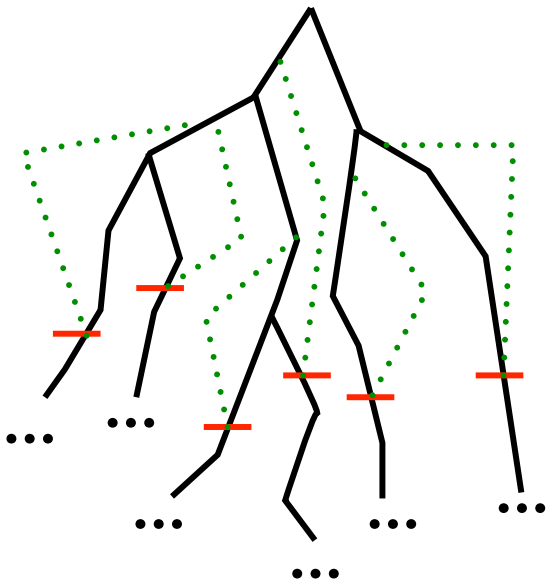
nics à la Rackoff
s of [BJK10]



hms

Finite memory → Exponential memory

Then λ'_1 is winning and finite memory

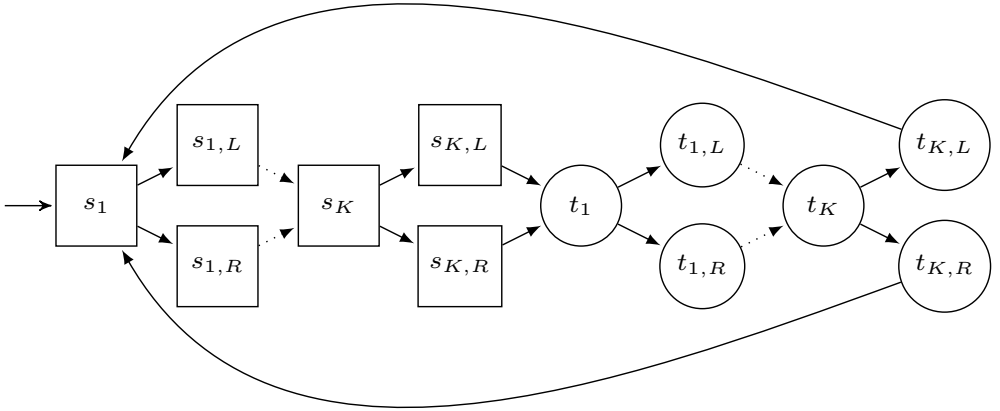


wqo+Koenig's lemma

① Exponential memory is sufficient.

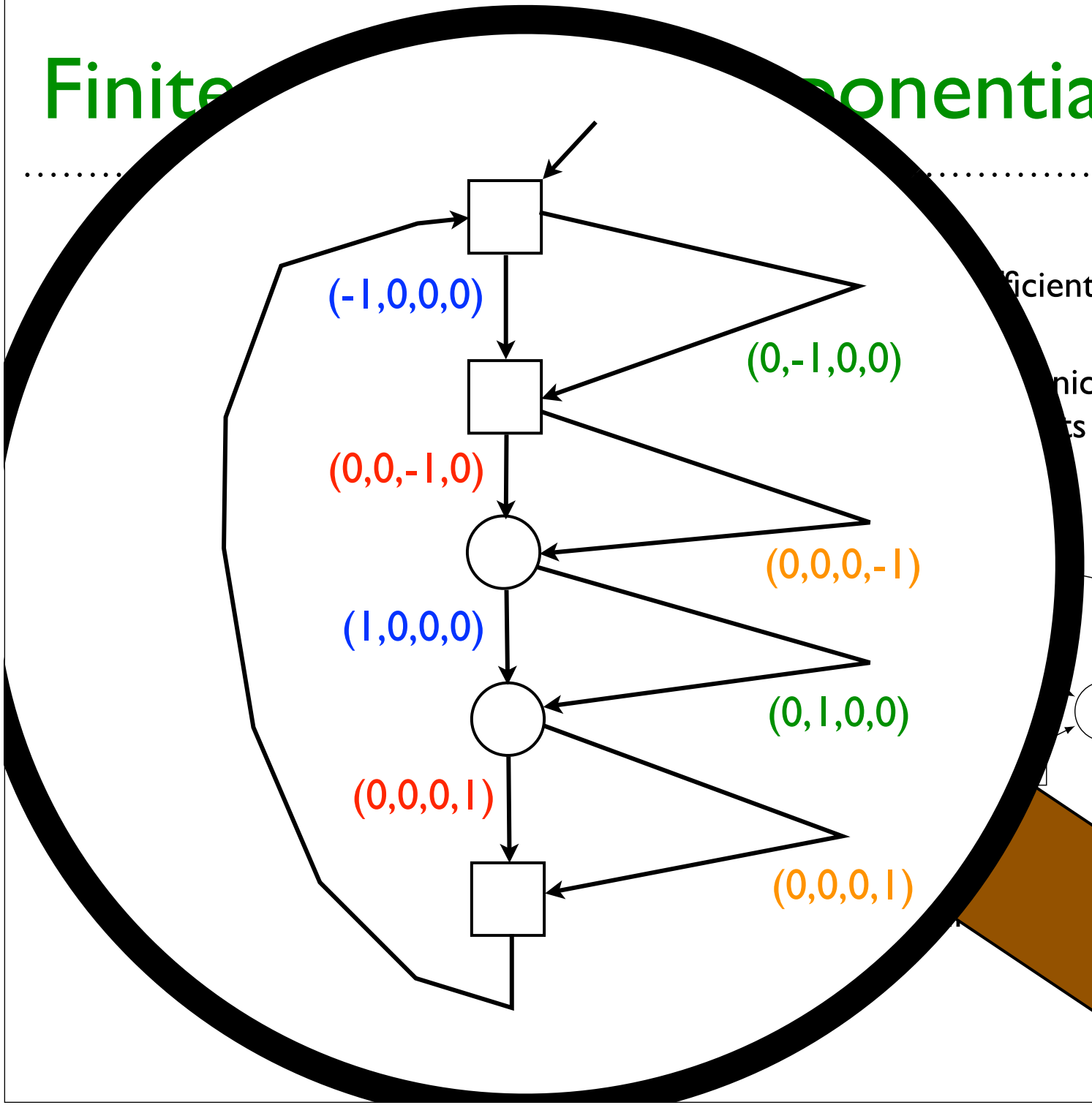
- Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

② Exponential memory is needed



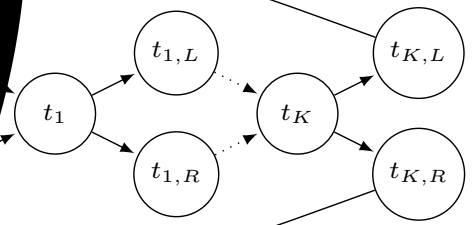
③ Leads to symbolic and incremental algorithms

Finite ponential memory



ficient.

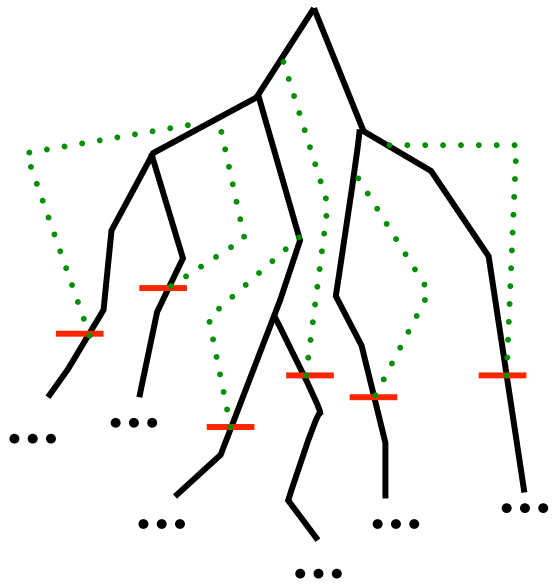
nics à la Rackoff
ts of [BJK10]



thms

Finite memory → Exponential memory

Then λ'_1 is winning and finite memory

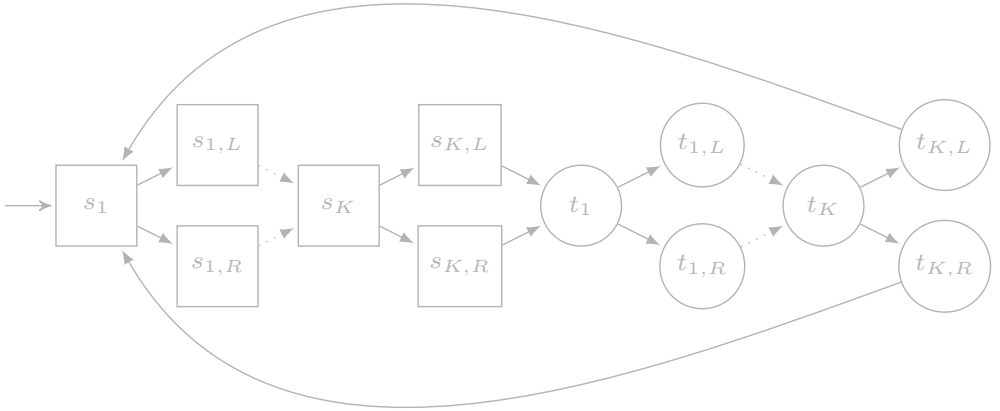


wqo+Koenig's lemma

① Exponential memory is sufficient.

- Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

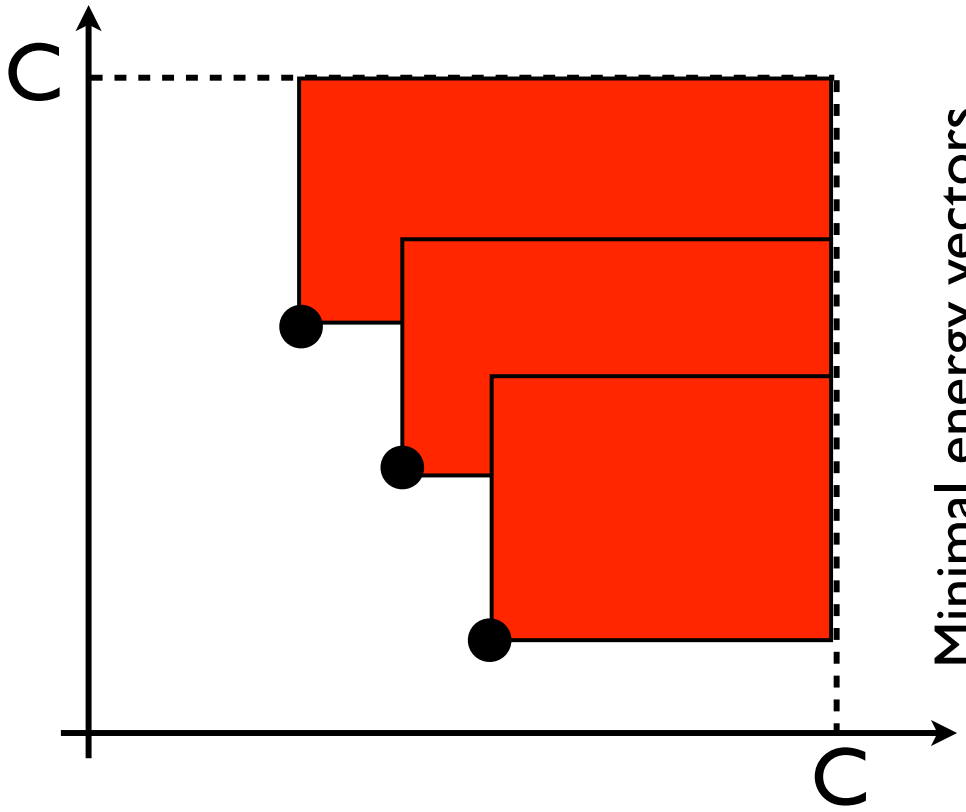
② Exponential memory is needed



③ Leads to symbolic and incremental algorithms

Finite ponential memory

C=max. constant appearing in the SCT

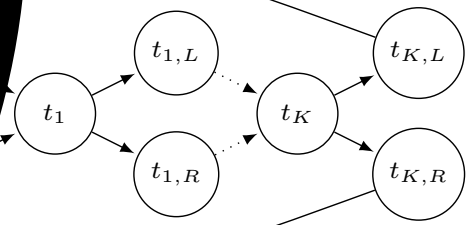


Incremental and **symbolic** algorithm

ficient.

nics à la Rackoff

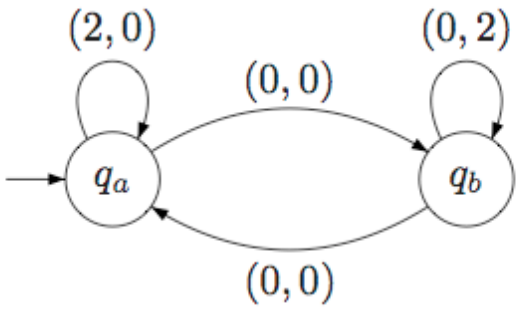
ts of [BJK10]



thms

gMPGs - Infinite Memory

To play optimally gMPGs, infinite memory is necessary



Consider the strategy that alternates visits to q_a and q_b such that after the n^{th} alternation, the self-loop on the visited state q ($q \in \{q_a, q_b\}$) is taken **so many times** that the average frequency of q gets larger than (n-1)/n in the current finite prefix of the play. This is always possible and achieves threshold (2, 2) for Lim Sup MP.

Player 2 - Memoryless Strategies

gEGs are particular cases of "zero-games" played on **vector addition systems extended with states** (VASS).

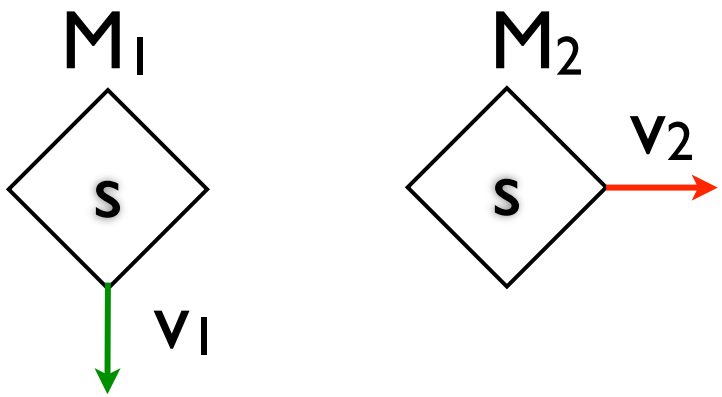
For such games, [**B**razdil**J**ancar**K**uncera10] establishes that **memoryless** strategies are sufficient for Player 2.

Lemma[BJK10]. Memoryless strategies are sufficient for Player 2 to win in "zero-games" played on VASS.

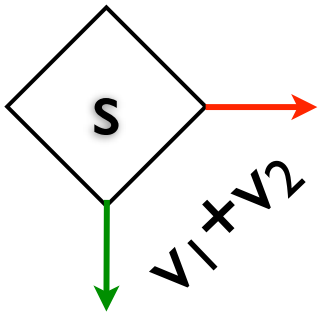
Player 2 - Memoryless - Why ?

Intuition: Player 2 cannot play better if he knows the energy levels.

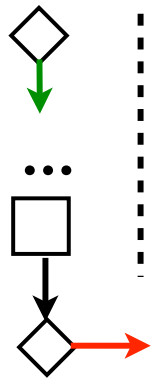
If



Then



Strategy for Player 1
alternates
when Player 2 changes in s



Play like in M_1

Play like in M_2

...

Player 2 - Memoryless Strategies

gEGs are particular cases of "zero-games" played on **vector addition systems extended with states** (VASS).

For such games, [**B**razdil**J**ancar**K**uncera10] establishes that **memoryless** strategies are sufficient for Player 2.

Lemma[BJK10]. Memoryless strategies are sufficient for Player 2 to win in "zero-games" played on VASS.

Theorem[BJK10]. Zero-game played on vector addition systems extended with state can be solved in **ExpSpace**.

Corollary. The unknown initial credit problem in gEGs is in **ExpSpace**.

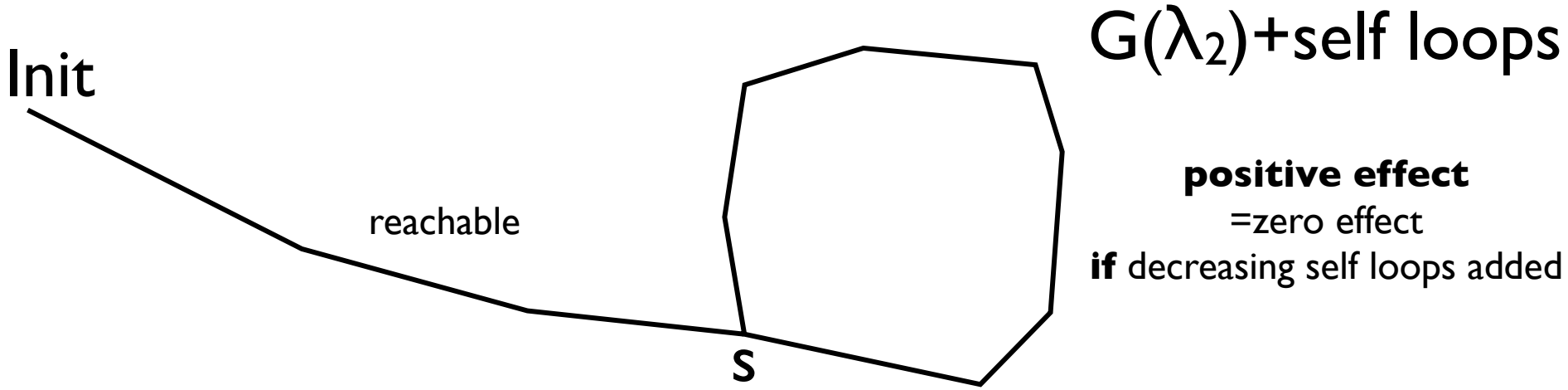
Complexity

Complexity of gEGs

- Memoryless strategies are sufficient for Player 2 to win a gEG G.
- Let $\lambda_2 \in \Sigma_{2,\mathbf{m}}$, $G(\lambda_2)$ is a multi-weighted graph.
- λ_2 is losing **iff** $G(\lambda_2)$ contains a reachable cycle (not necessarily simple) with positive effect on all dimensions.

Complexity of gEGs

- Memoryless strategies are sufficient for Player 2 to win a gEG G.
- Let $\lambda_2 \in \Sigma_{2,\mathbf{m}}$, $G(\lambda_2)$ is a multi-weighted graph.
- λ_2 is losing **iff** $G(\lambda_2)$ contains a reachable cycle (not necessarily simple) with positive effect on all dimensions.



Theorem[**K**osaraju,**S**ullivan88]. Given a multi-weighted graph G, it is decidable in deterministic polynomial time if G contains a state s which is reachable from itself with a (not necessarily simple) path with zero effect on all dimensions.

Complexity of gEGs

- Memoryless strategies are sufficient for Player 2 to win a gEG G.
- Let $\lambda_2 \in \Sigma_{2,\mathbf{m}}$, $G(\lambda_2)$ is a multi-weighted graph.
- λ_2 is losing **iff** $G(\lambda_2)$ contains a reachable cycle (not necessarily simple) with positive effect on all dimensions.

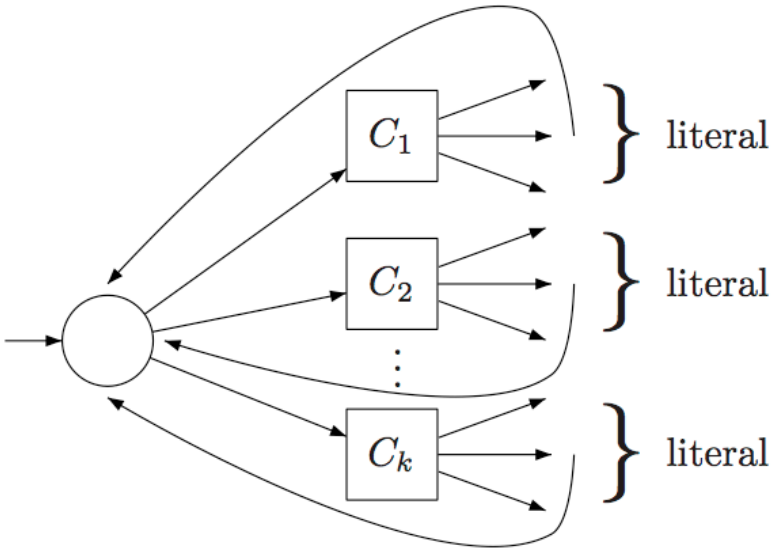
☞ **Lemma**. The unknown initial credit problem in gEGs belongs to coNP.

Complexity of gEGs

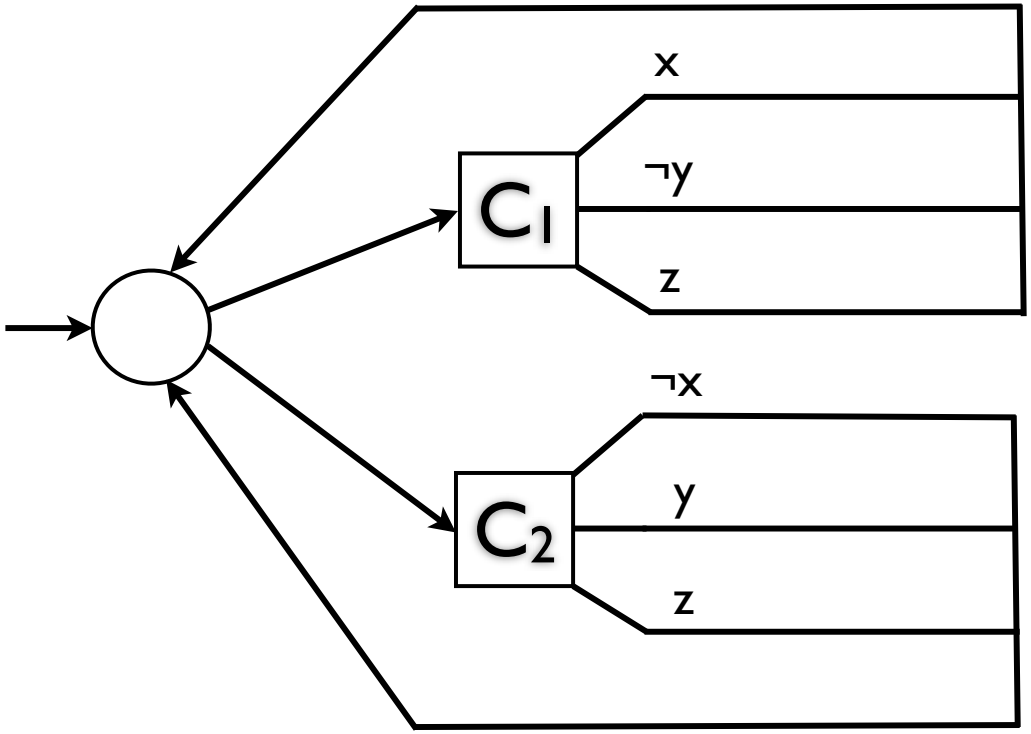
Lemma. The unknown initial credit problem in gEGs is coNP-Hard.

Proof. We show that deciding whether **Player 1 has a winning strategy** is as hard as deciding if a 3CNF formula is **unsatisfiable**.

Let ψ be a 3CNF formula with clauses $C_1, C_2, \ldots, C_k$ over variables $\{x_1, x_2, \ldots, x_n\}$. We construct from ψ the following game structure with weight in $\mathbb{Z}^{2n}$:

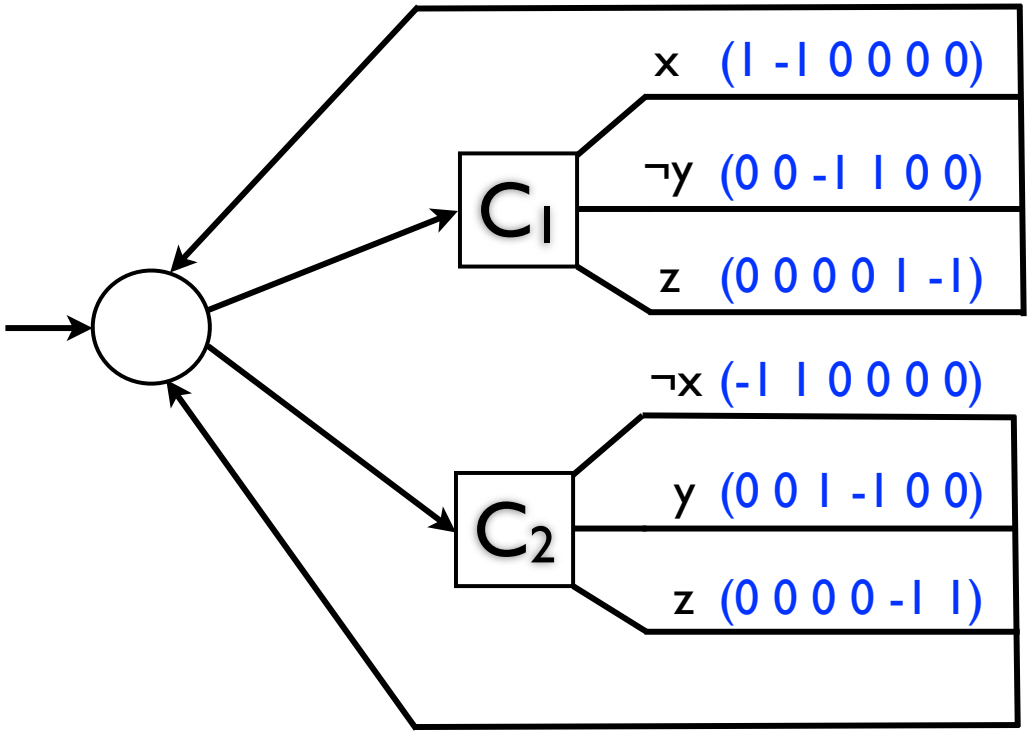


Complexity of gEGs



Ex: $\Phi=(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z)$

Complexity of gEGs



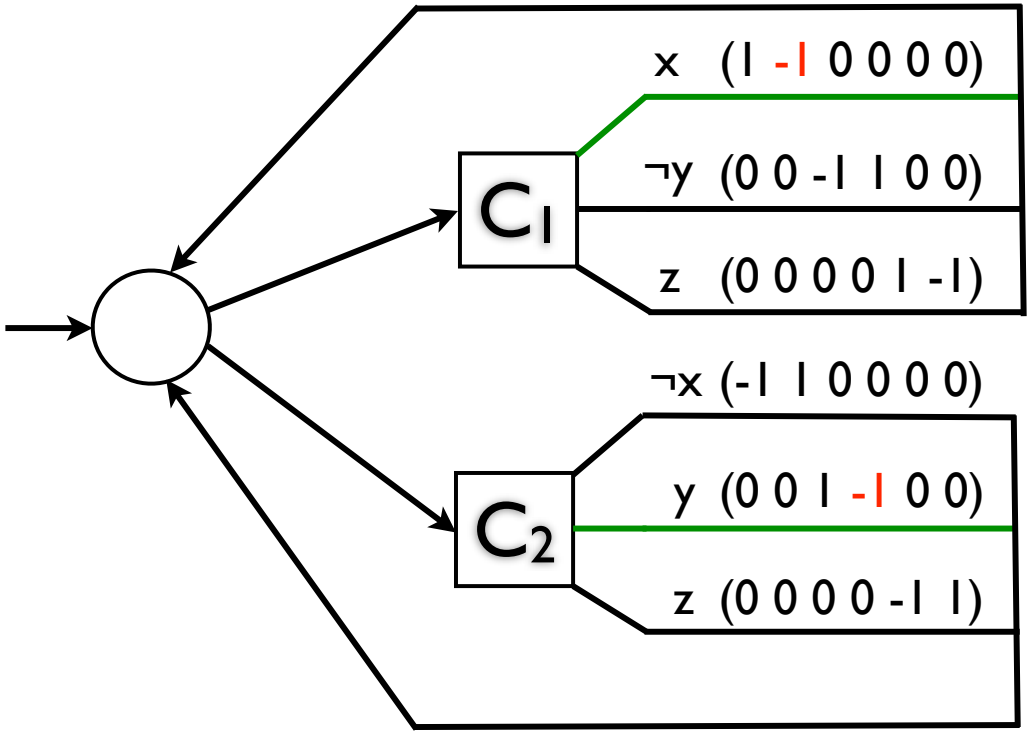
Ex: $\Phi=(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z)$

We define the weight labeling as follows:

-every edge is labeled by $\{0\}^{2n}$ with the exception of edges going from literals back to initial state.
-for a literal y and an edge back to the initial state, the weight vector contains:

- 1 in the dimension of y
- -1 in the dimension of the complement of y
- 0 otherwise

Complexity of gEGs



Ex: $\Phi=(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z)$

$v(x)=1, v(y)=1, v(z)=0$

$v \models \Phi$

$\lambda_2(C_1)=x$

$\lambda_2(C_2)=y$

Φ is satisfiable implies Player 2 wins

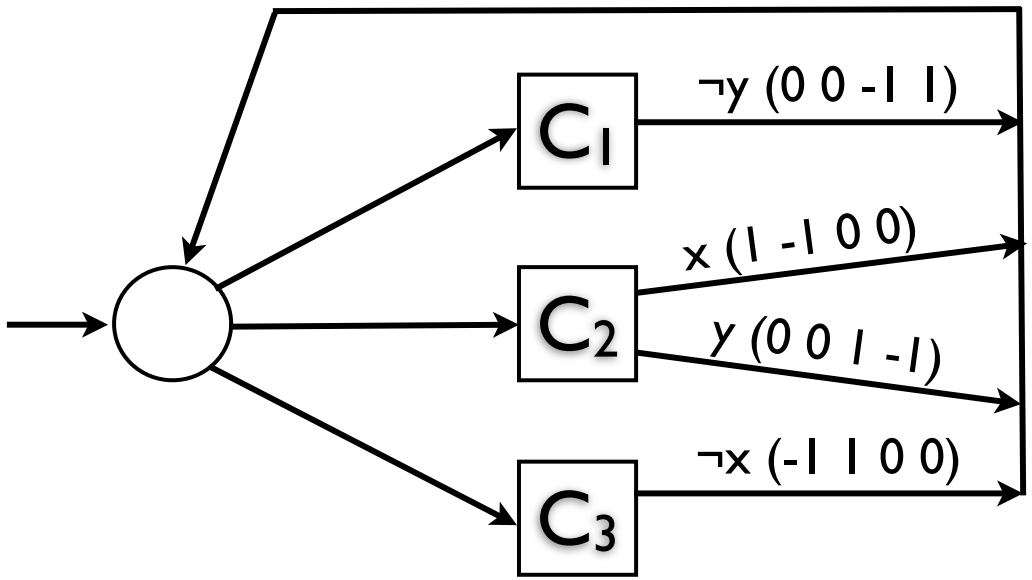
Let v be s.t. $v \models \Phi$. We construct λ_2 as follows: in each clause C_i, λ_2 chooses l_{ij} s. t. $v \models l_{ij}$. Now, take any λ_1 and consider the play consistent with λ_1 and λ_2. There must exists C_i that appears ∞-often along this play: the dimension that correspond to $\neg l_{ij}$ is decreased ∞-often without ever being increased ! There is no initial credit that can help Player 1 !

Complexity of gEGs

Φ is unsatisfiable implies Player 1 is winning
or equivalently **Φ is unsatisfiable implies Player 2 is not winning**

As Φ is unsatisfiable, when Player 2 chooses one literal per clause (we know that he can play optimally without memory), it has to choose two literals that are **complementary**. Let assume that the choice of Player 2 are complementary for clauses C_i and C_j. In that case, the winning strategy for Player 1 is to alternate between C_i and C_j. This strategy is winning for a initial credit of 1 in all dimension.

$$\Phi = (\neg y) \wedge (x \vee y) \wedge (\neg x)$$

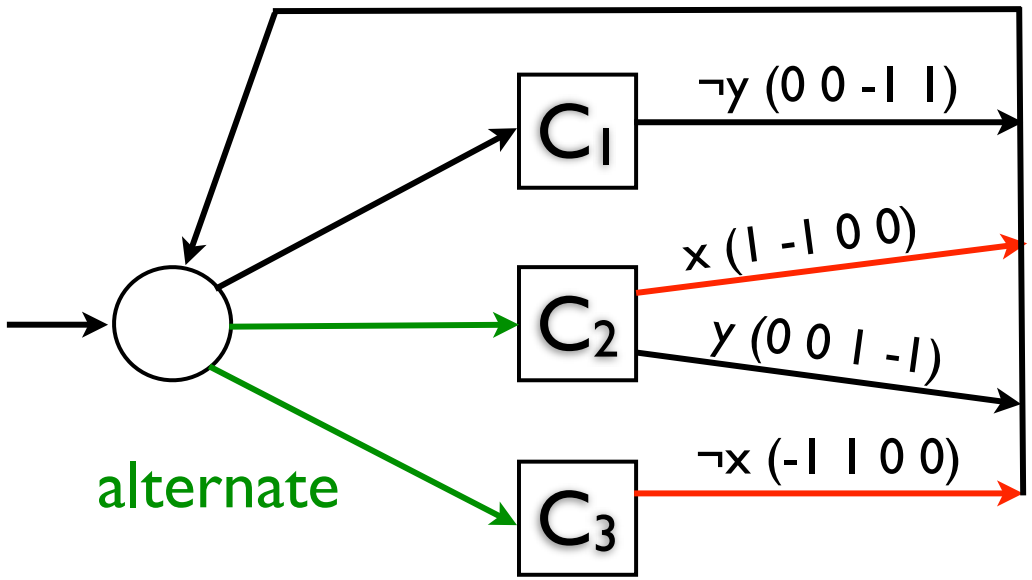


Complexity of gEGs

Φ is unsatisfiable implies Player 1 is winning
or equivalently **Φ is unsatisfiable implies Player 2 is not winning**

As Φ is unsatisfiable, when Player 2 chooses one literal per clause (we know that he can play optimally without memory), it has to choose two literals that are **complementary**. Let assume that the choice of Player 2 are complementary for clauses C_i and C_j. In that case, the winning strategy for Player 1 is to alternate between C_i and C_j. This strategy is winning for a initial credit of 1 in all dimension.

$$\Phi = (\neg y) \wedge (x \vee y) \wedge (\neg x)$$



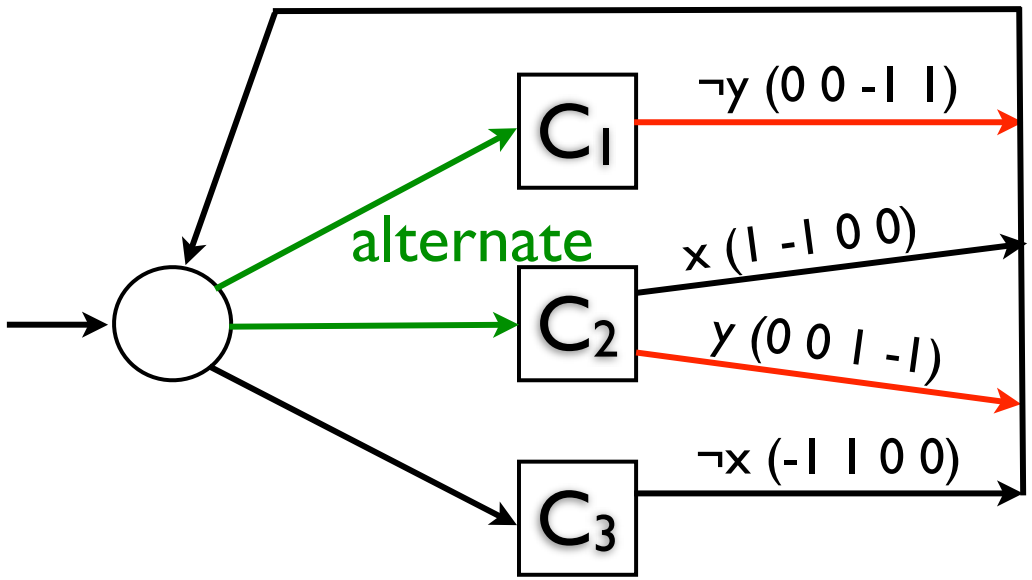
(1 -1 0 0)
+
(-1 1 0 0)
=
(0 0 0 0)

Complexity of gEGs

Φ is unsatisfiable implies Player 1 is winning
or equivalently **Φ is unsatisfiable implies Player 2 is not winning**

As Φ is unsatisfiable, when Player 2 chooses one literal per clause (we know that he can play optimally without memory), it has to choose two literals that are **complementary**. Let assume that the choice of Player 2 are complementary for clauses C_i and C_j. In that case, the winning strategy for Player 1 is to alternate between C_i and C_j. This strategy is winning for a initial credit of 1 in all dimension.

$$\Phi = (\neg y) \wedge (x \vee y) \wedge (\neg x)$$



(0 0 -1 1)
+
(0 0 1 -1)
=
(0 0 0 0)

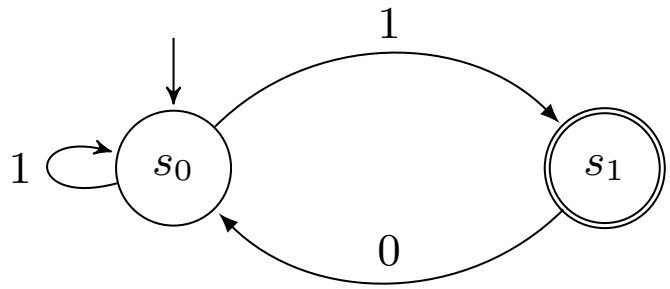
Complexity of gEGs

Theorem. The unknown initial credit problem in gEGs is **coNP-C**.

Trading memory for randomness

Trading memory for randomness

First, an example: Mean-payoff + Büchi game

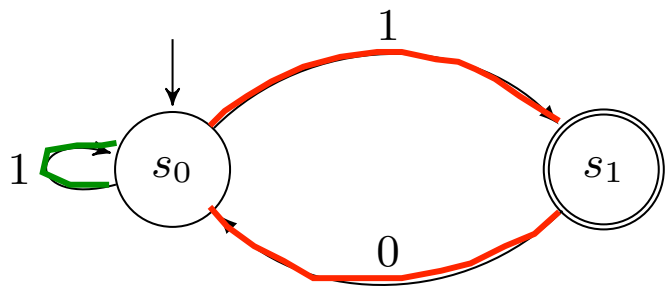


Value 1 requires **infinite memory**

Value 1-ε possible with finite (**exponential** in ε) memory.

Exponential memory strategy can be replaced by a memoryless **randomized** strategy: **1-ε** is reached with **probability 1**.

Trading memory for randomness



Combine two types of strategies:

(i) Good for payoff and (ii) Good for Büchi (F-attractor)

Both are **memoryless**

To play optimally, frequency of Good for payoff must strictly increase and tend to 1 but Good for Büchi played ∞-often.

To play ε-optimally, frequency of Good for payoff = **$1-(\varepsilon/2)$**

☞ either **count** or use **randomization**

Trading memory for randomness

Theorem. In (1-dim) **mean-payoff parity games**, ε-optimality can be achieved either using

- ➢ a finite (exponential) memory strategy, or
- ➢ a memoryless randomized strategy

Randomization does not help in energy games (safety).

Summary

	gEGs	gMPs	MP Parity
Memory for playing optimally	Pl. 1: Finite Pl. 2: Memoryless	Pl. 1: Infinite Pl. 2: Memoryless [Vel10]	Player 1: Infinite Exponential or Memoryless rand. for ε-opt
Complexity General case	coNP-C	NP∩coNP [Vel10]	NP∩coNP [CHJ05]
Complexity Special cases	NP-C ($\Sigma_{1,\mathbf{m}}$) P (2D,{-1,0,1}) [Cha10]	coNP-C ($\Sigma_{1,\mathbf{f}}$,$\Sigma_{2,\mathbf{f}}$)	