On finite-memory determinacy of games on graphs

Patricia Bouyer
LSV, CNRS, Univ. Paris-Saclay, ENS Paris-Saclay
France

Based on joint work with Stéphane Le Roux, Youssouf Oualhadj,
Mickael Randour, Pierre Vandenhove
(Published at CONCUR'20)

The talk in one slide

Strategy synthesis for two-player games

- Find good and simple controllers for systems interacting with an antagonistic environment

The talk in one slide

Strategy synthesis for two-player games

- Find good and simple controllers for systems interacting with an antagonistic environment

« Good »?

- Performance w.r.t. objectives / payoffs / preference relations

The talk in one slide

Strategy synthesis for two-player games

- Find good and simple controllers for systems interacting with an antagonistic environment

« Good »?

- Performance w.r.t. objectives / payoffs / preference relations

« Simple »?

- Memoryless strategies
- Finite-memory strategies

The talk in one slide

Strategy synthesis for two-player games

- Find good and simple controllers for systems interacting with an antagonistic environment

« Good »?

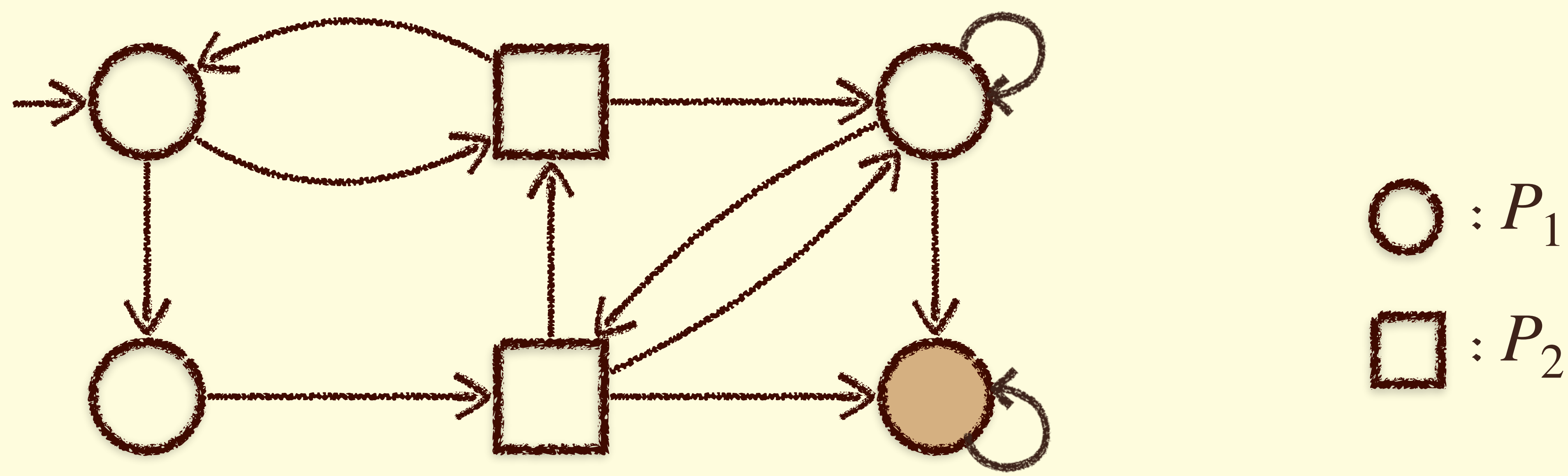
- Performance w.r.t. objectives / payoffs / preference relations

« Simple »?

- Memoryless strategies
- Finite-memory strategies

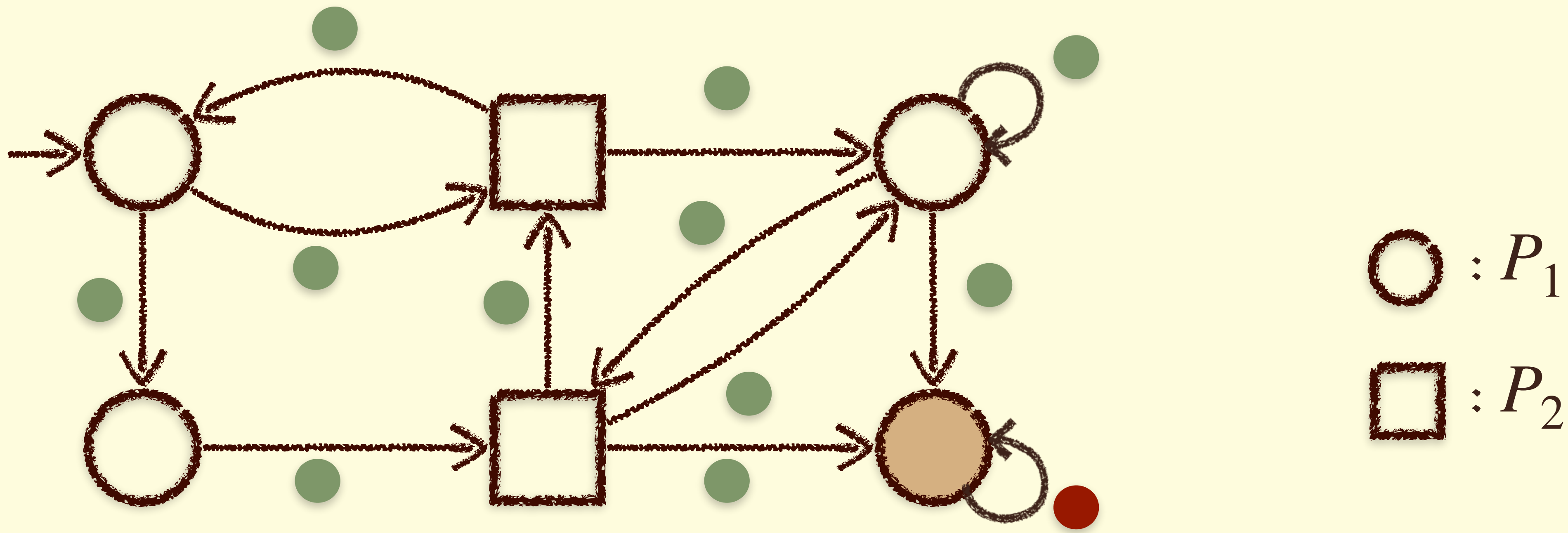
When are simple strategies sufficient to play optimally?

The setting - Example of a game



Reachability winning condition for P_1

The setting - Example of a game

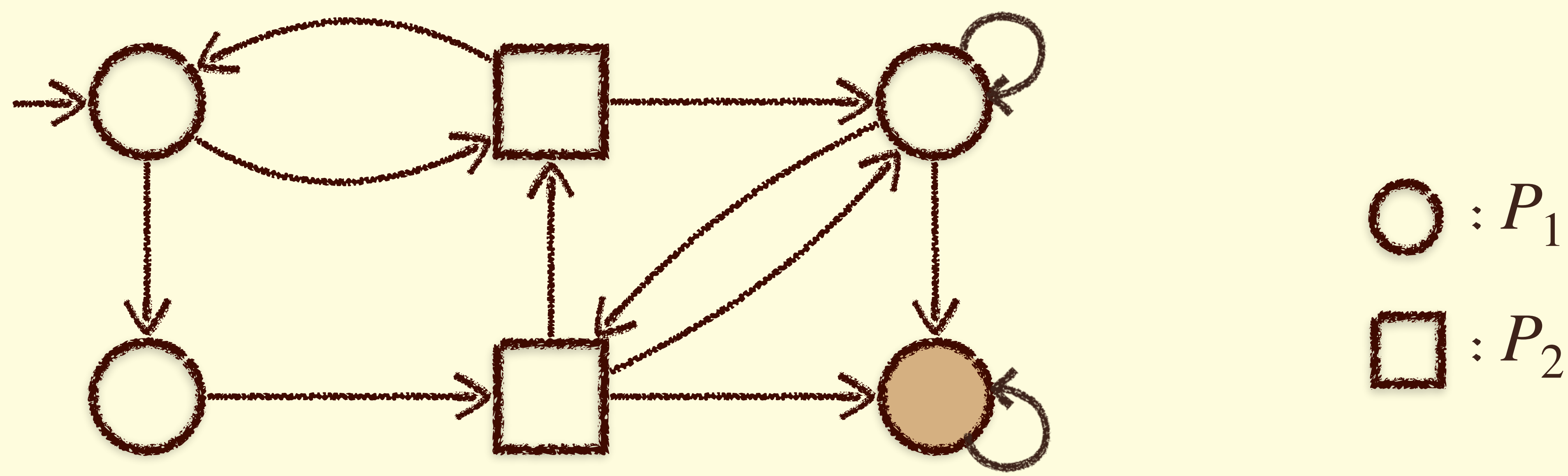


Reachability winning condition for P_1

Use of colors to define winning condition/preference relation

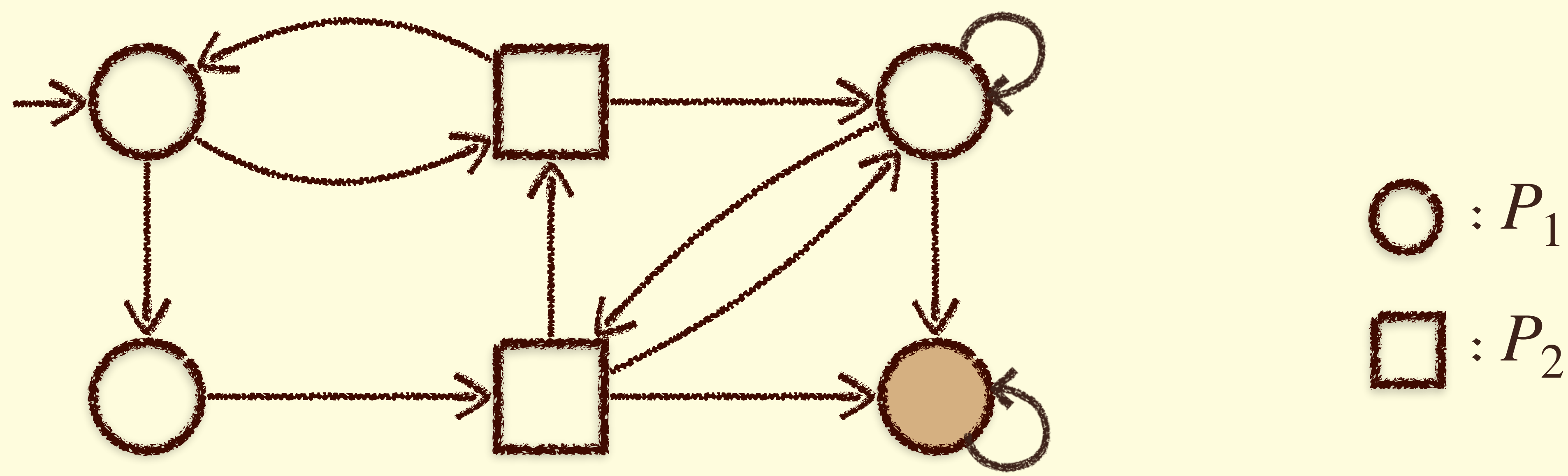
$$\bullet^{*} \; \bullet \, (\bullet + \bullet)^{\omega}$$

The setting - Example of a game



Reachability winning condition for P_1

The setting - Example of a game



Reachability winning condition for P_1

The game is played using strategies:

$$\sigma_i : S^*S_i \rightarrow E$$

Families of strategies

$$\sigma_i : S^* S_i \to E$$

Families of strategies

$$\sigma_i : S^*S_i \rightarrow E$$

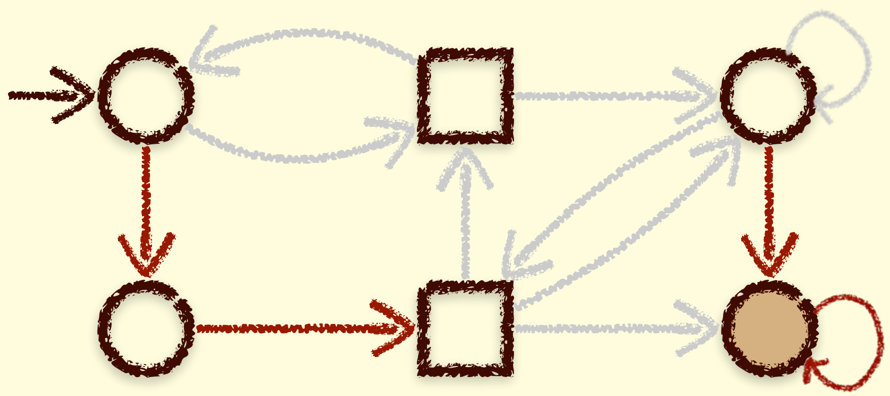
Subclasses of interest

Families of strategies

$$\sigma_i : S^*S_i \rightarrow E$$

Subclasses of interest

- Memoryless strategy: $\sigma_i : S_i \rightarrow E$



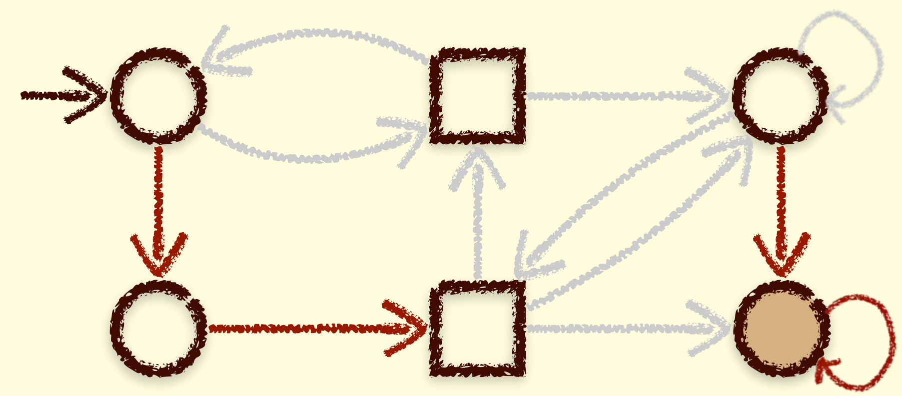
« Reach the target »

Families of strategies

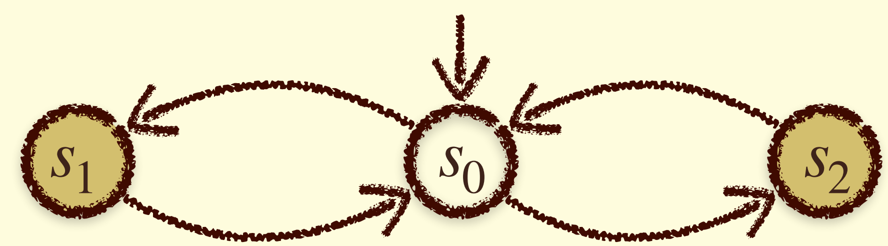
$$\sigma_i : S^*S_i \to E$$

Subclasses of interest

- Memoryless strategy: $\sigma_i : S_i \to E$
- Finite-memory strategy: σ_i defined by a finite-state Mealy machine



« Reach the target »



« Visit both s_1 and s_2 »

Every odd visit to s_0, go to s_1

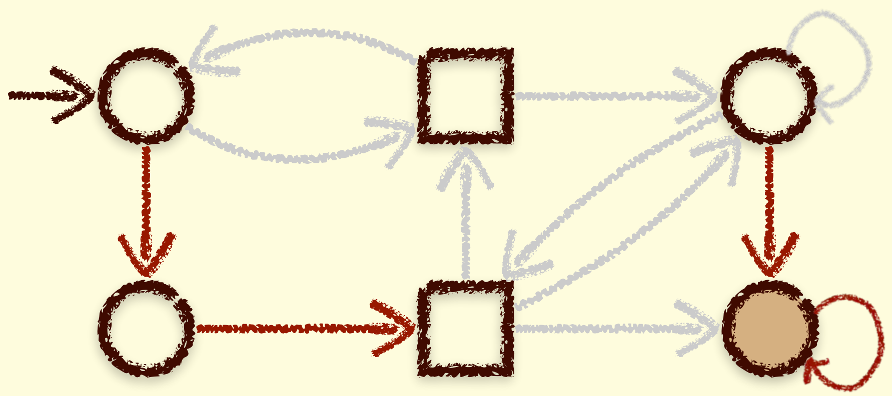
Every even visit to s_0, go to s_2

Families of strategies

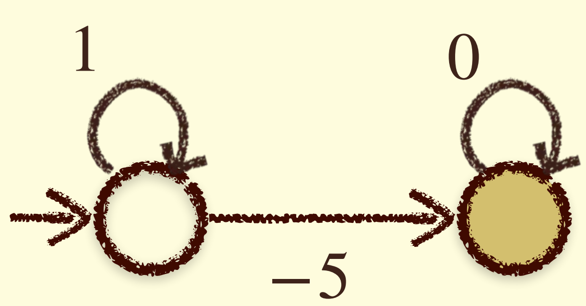
$$\sigma_i : S^* S_i \to E$$

Subclasses of interest

- Memoryless strategy: $\sigma_i : S_i \to E$
- Finite-memory strategy: σ_i defined by a finite-state Mealy machine

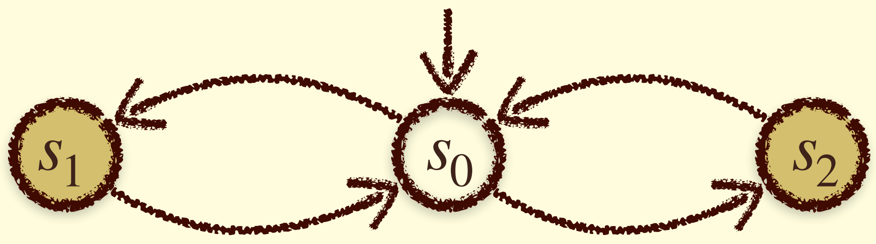


« Reach the target »



« Reach the target with energy 0 »

Loop 5 times in the initial state



« Visit both s_1 and s_2 »

Every odd visit to s_0, go to s_1

Every even visit to s_0, go to s_2

The setting - Preference relation

A preference relation $\sqsubseteq$ is a total preorder on C^ω.

$\pi \sqsubseteq \pi'$ and $\pi' \sqsubseteq \pi$ means that π and π' are equally appreciated

$\pi \sqsubseteq \pi'$ and $\pi' \not\sqsubseteq \pi$ means that π' is preferred over π

The setting - Preference relation

A preference relation $\sqsubseteq$ is a total preorder on C^ω.

$\pi \sqsubseteq \pi'$ and $\pi' \sqsubseteq \pi$ means that π and π' are equally appreciated

$\pi \sqsubseteq \pi'$ and $\pi' \not\sqsubseteq \pi$ means that π' is preferred over π

Examples

- $W \subseteq C^\omega$ winning condition:

  $\pi \sqsubseteq \pi'$ if either $\pi' \in W$ or $\pi \notin W$
- Quantitative real payoff f

  $\pi \sqsubseteq \pi'$ if $f(\pi) \leq f(\pi')$

  Ex: MP, AE, TP

The setting - Preference relation

A preference relation $\sqsubseteq$ is a total preorder on C^ω.

$\pi \sqsubseteq \pi'$ and $\pi' \sqsubseteq \pi$ means that π and π' are equally appreciated

$\pi \sqsubseteq \pi'$ and $\pi' \not\sqsubseteq \pi$ means that π' is preferred over π

Examples

- $W \subseteq C^\omega$ winning condition:

  $\pi \sqsubseteq \pi'$ if either $\pi' \in W$ or $\pi \notin W$
- Quantitative real payoff f

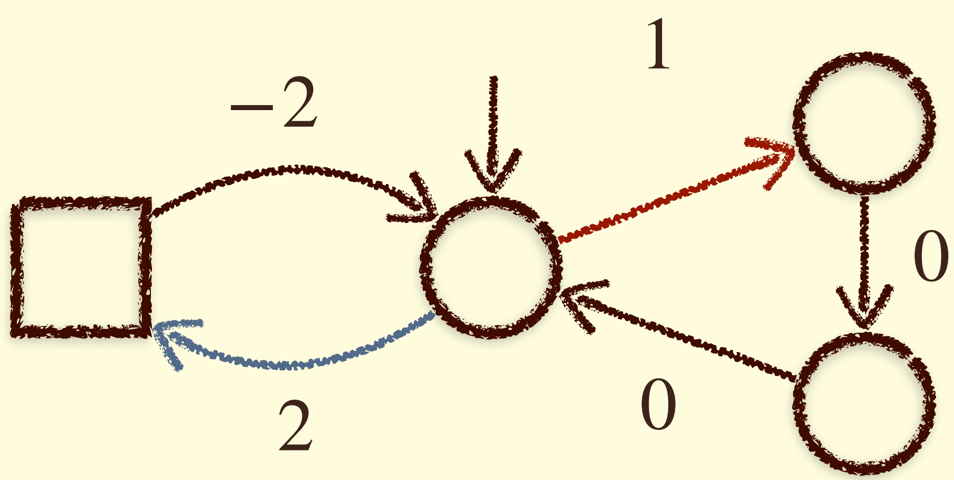
  $\pi \sqsubseteq \pi'$ if $f(\pi) \leq f(\pi')$

  Ex: MP, AE, TP

Zero-sum assumption:

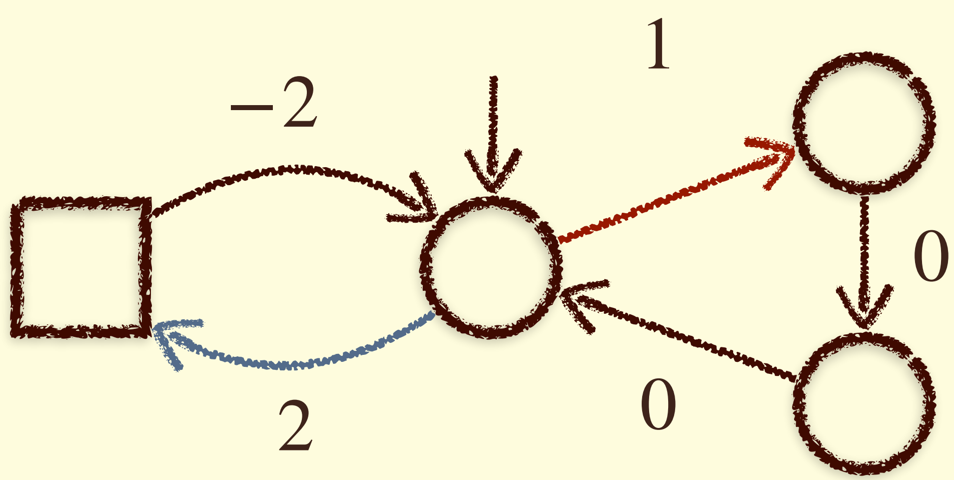
- Preference of P_1 is $\sqsubseteq$
- Preference of P_2 is $\sqsubseteq^{-1}$

Payoffs based on energy

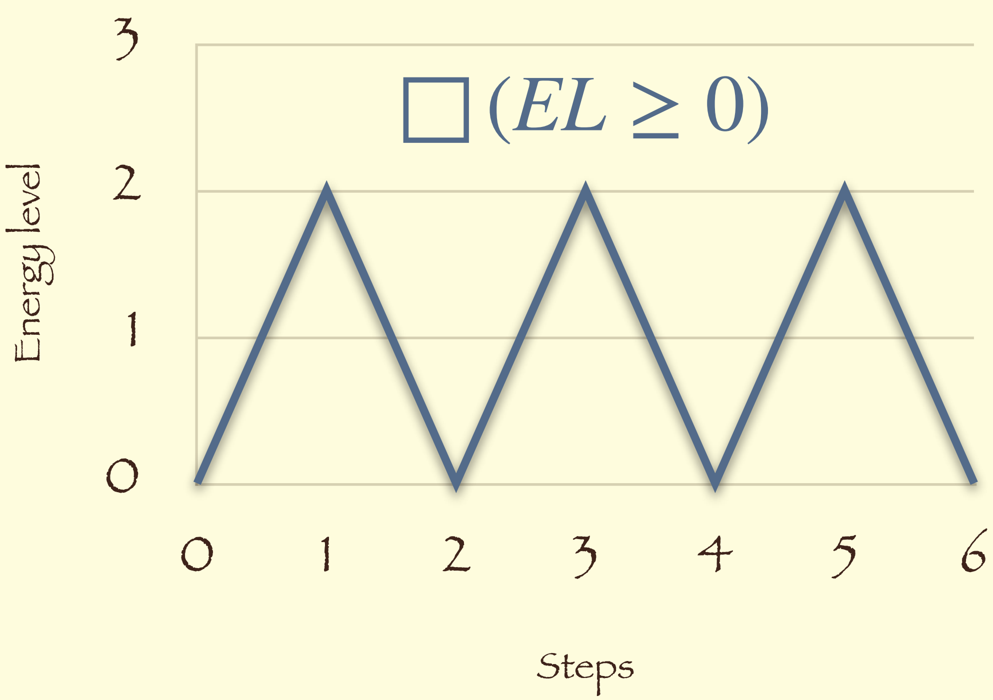


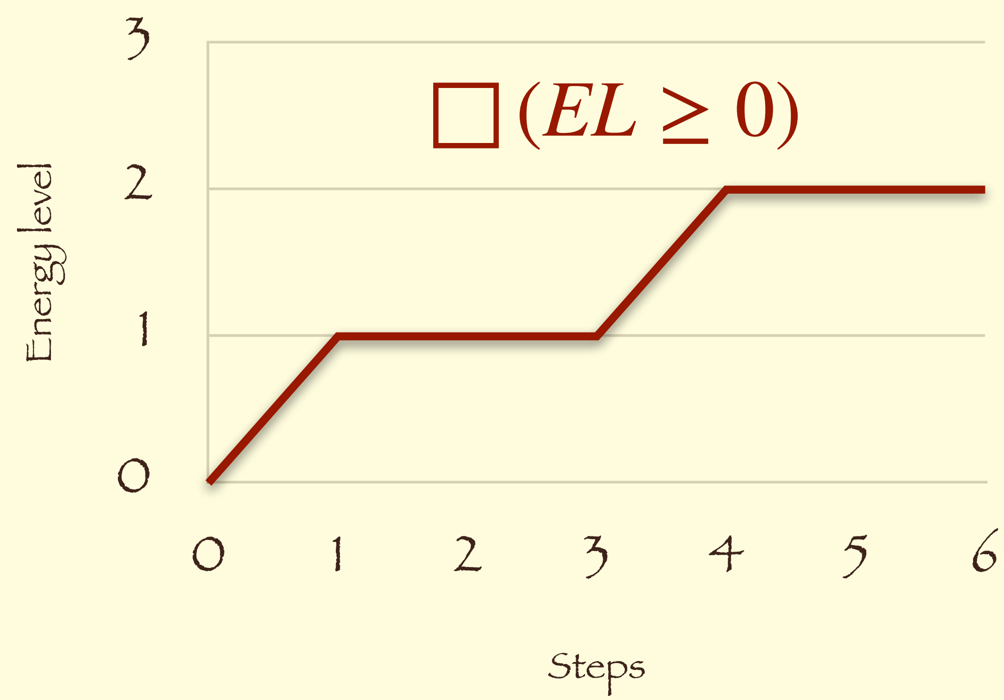
Focus on two memoryless strategies

Payoffs based on energy



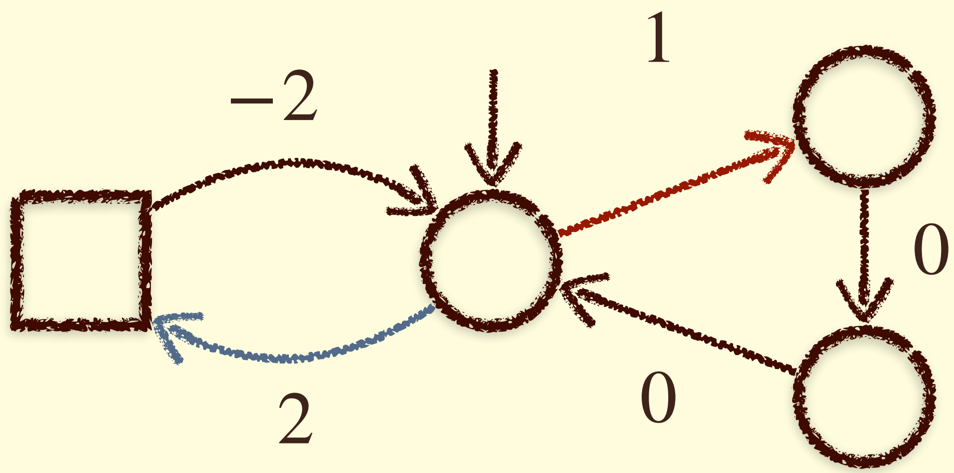
Focus on two memoryless strategies



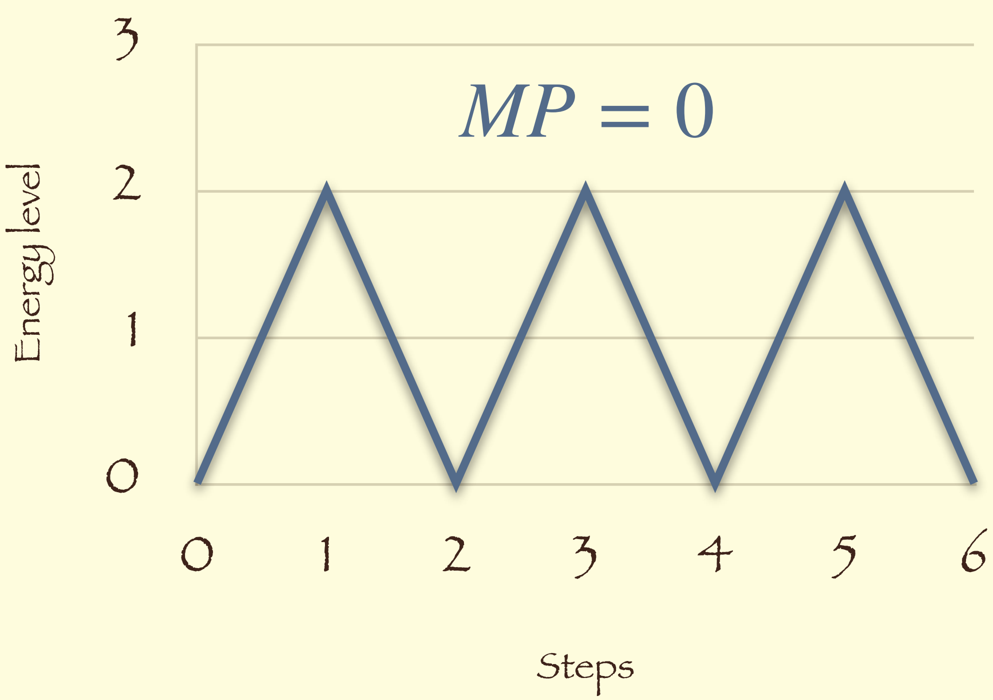


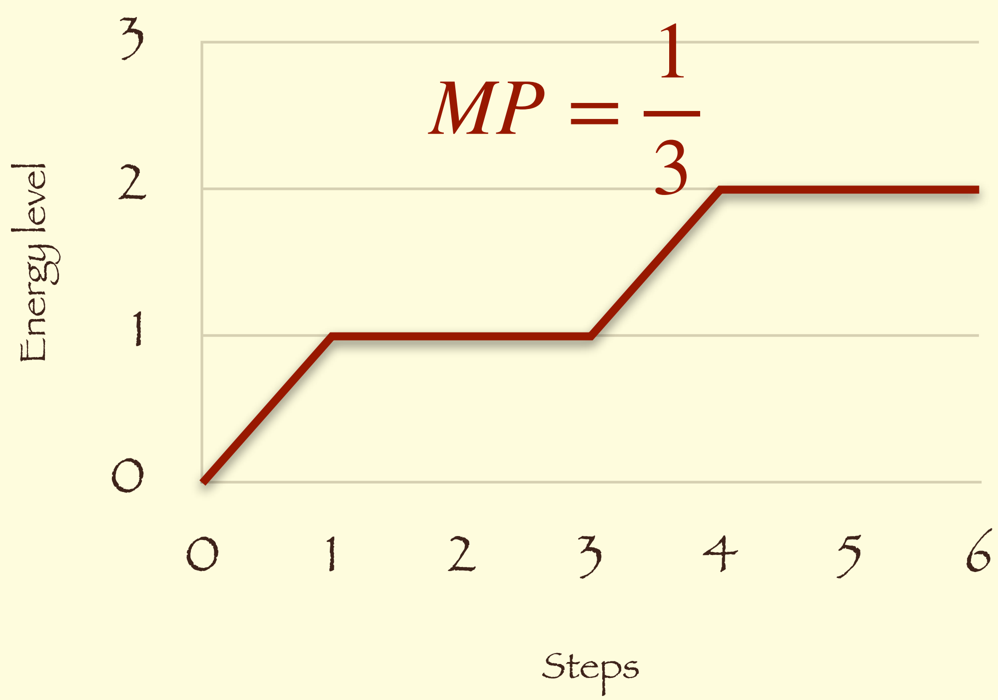
- Constraint on the energy level (EL)

Payoffs based on energy



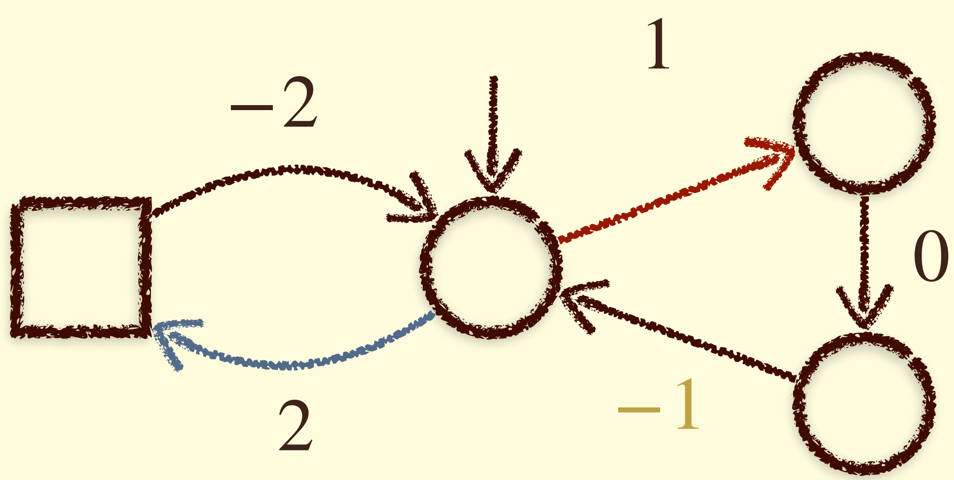
Focus on two memoryless strategies



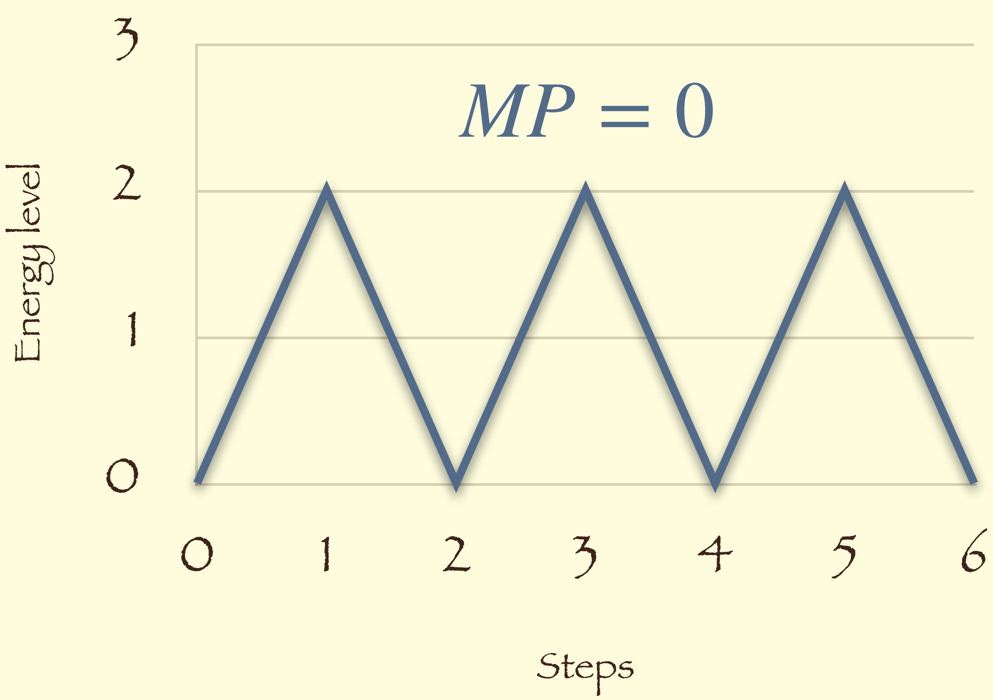


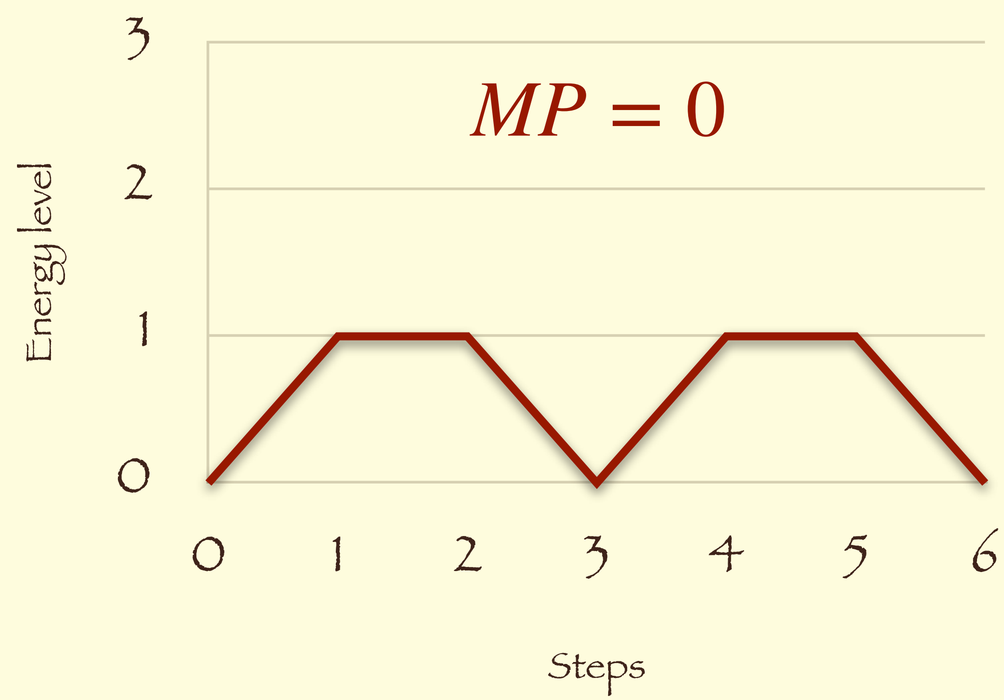
- Constraint on the energy level (EL)
- Mean-payoff (MP): long-run average payoff per transition

Payoffs based on energy



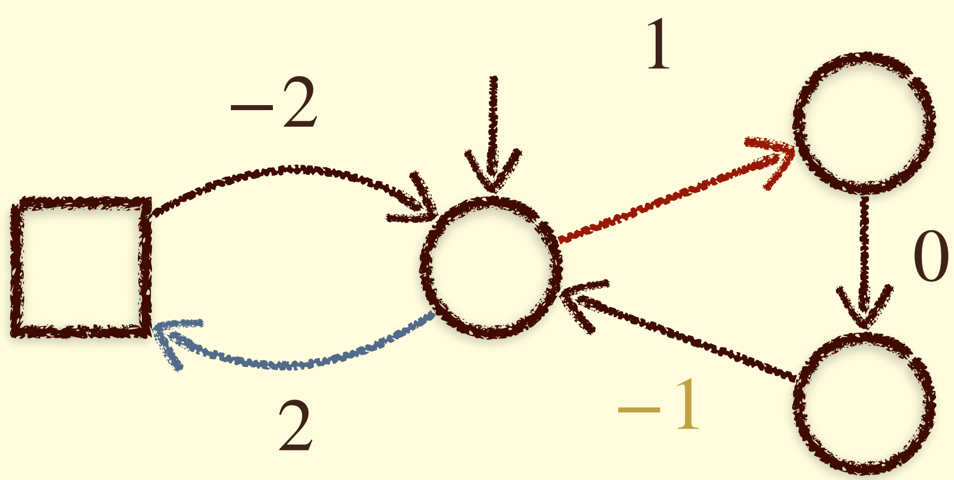
Focus on two memoryless strategies



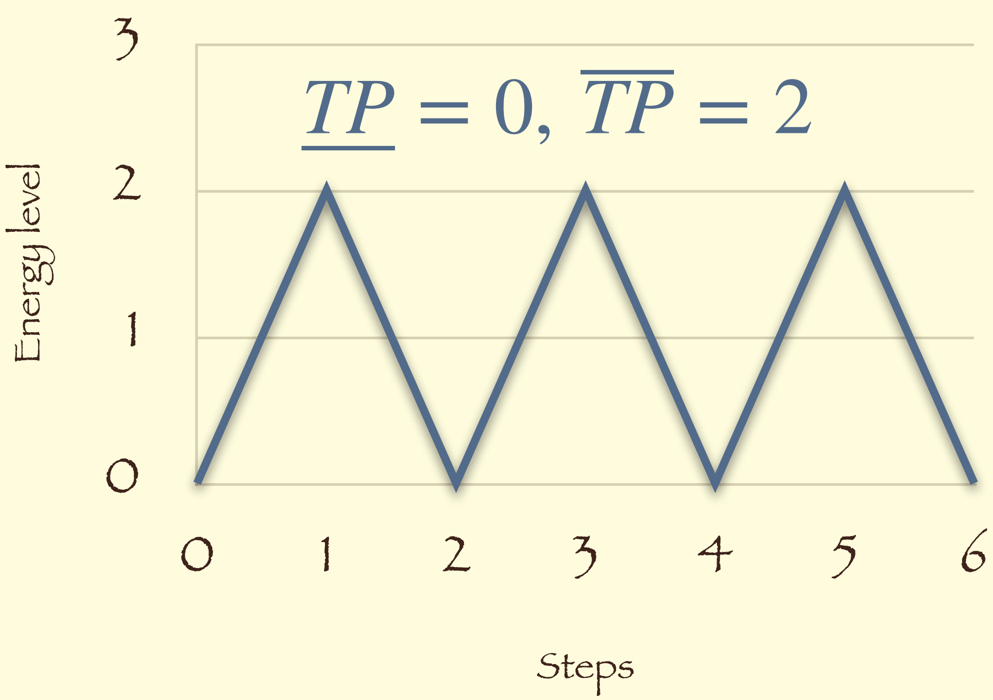


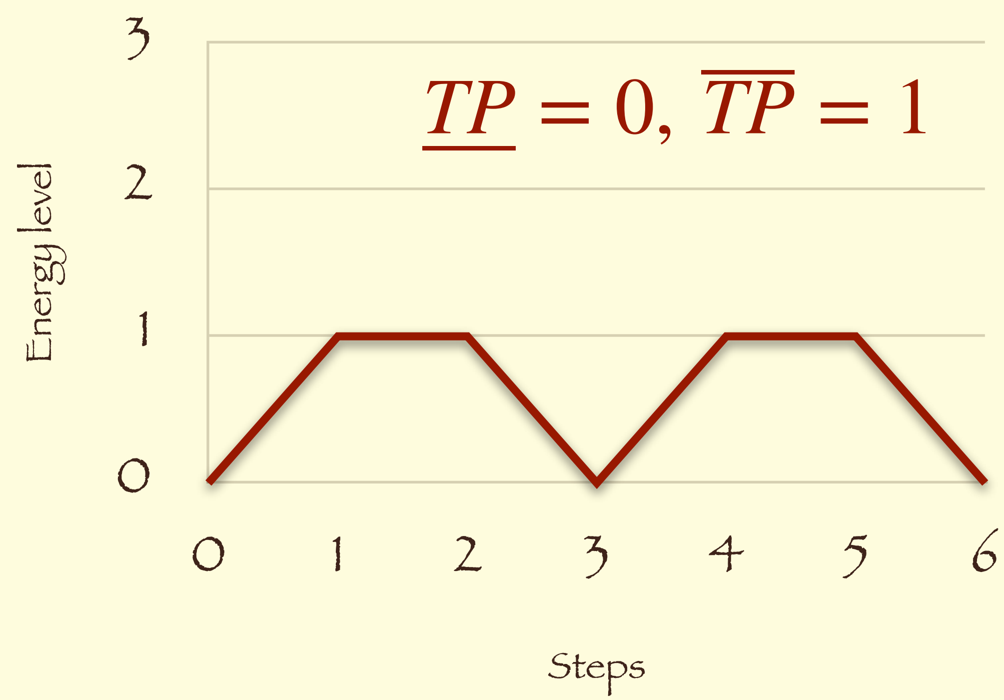
- Constraint on the energy level (EL)
- Mean-payoff (MP): long-run average payoff per transition

Payoffs based on energy



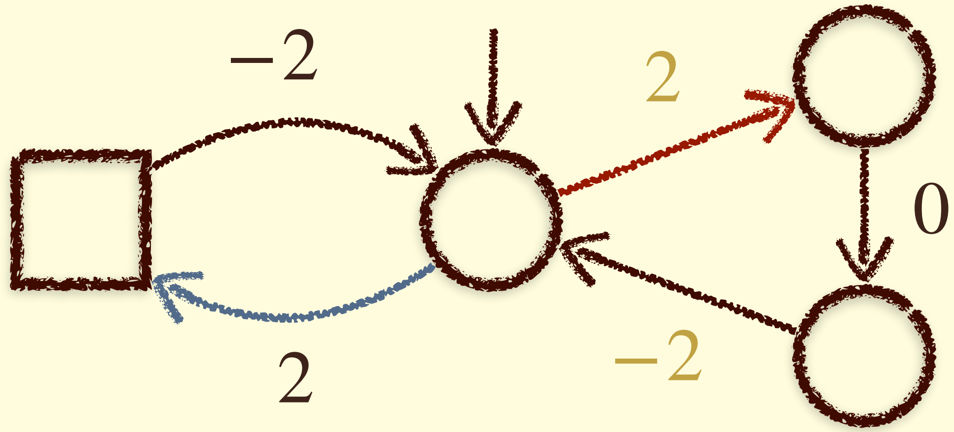
Focus on two memoryless strategies



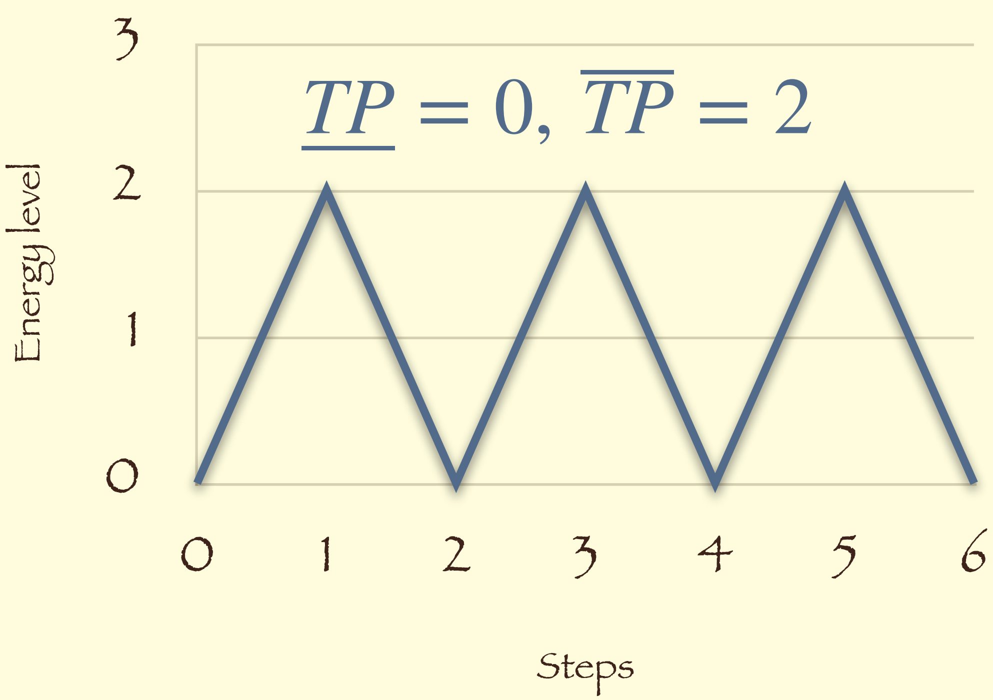


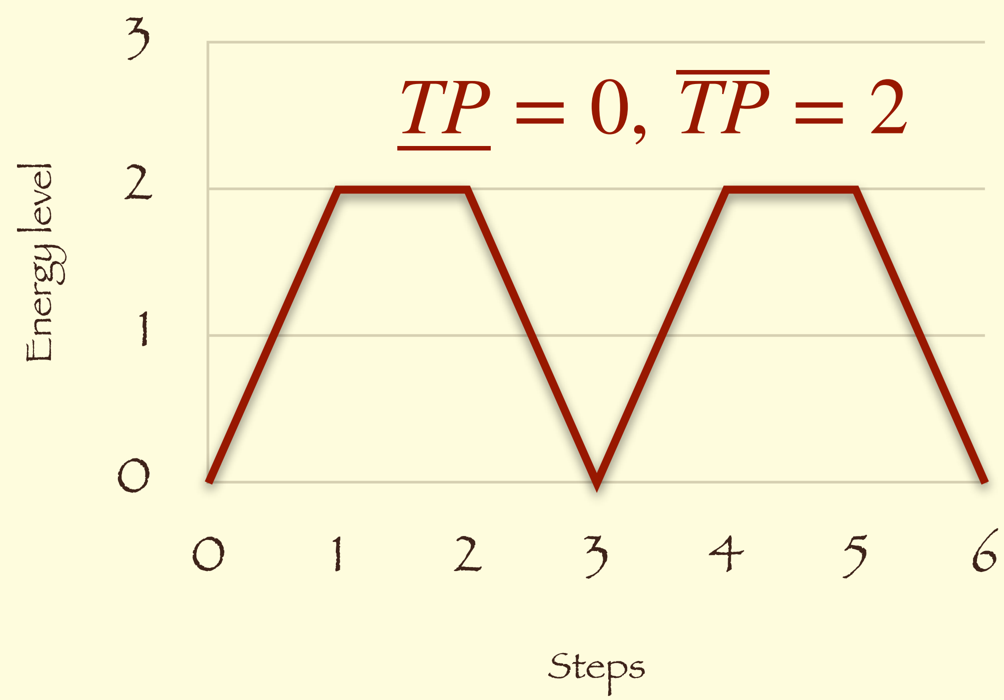
- Constraint on the energy level (EL)
- Mean-payoff (MP): long-run average payoff per transition
- Total-payoff (TP)

Payoffs based on energy



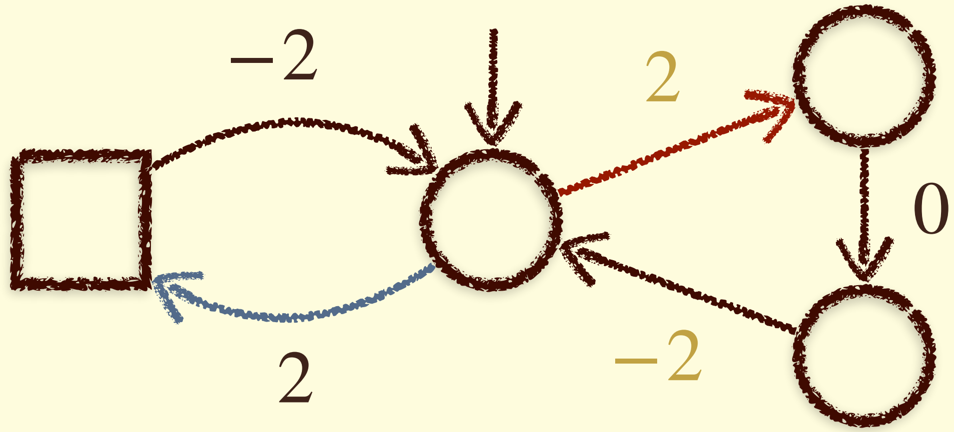
Focus on two memoryless strategies



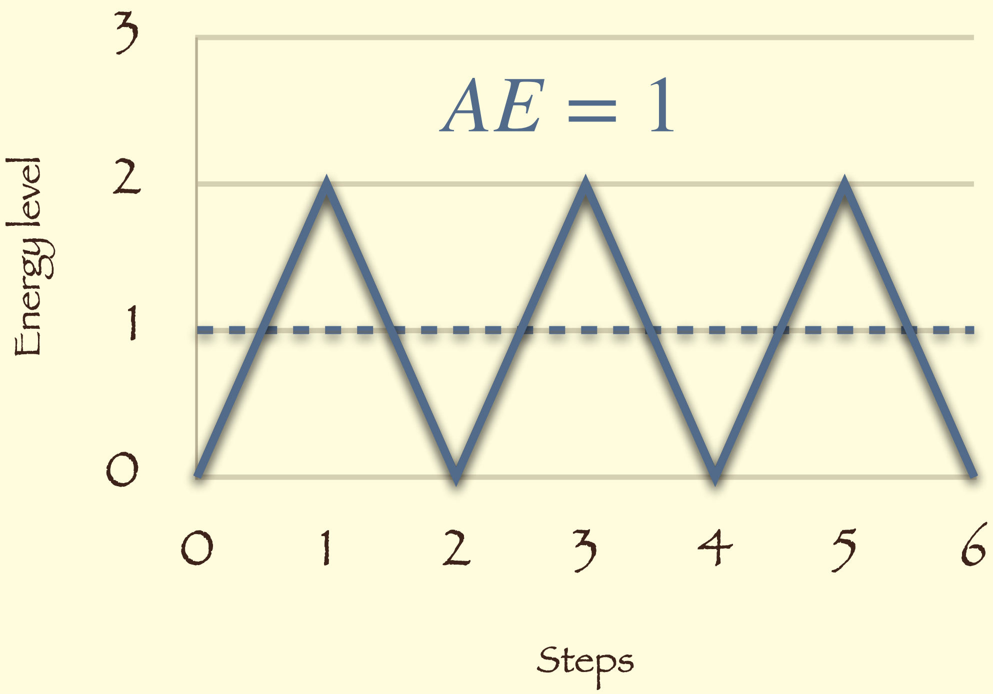


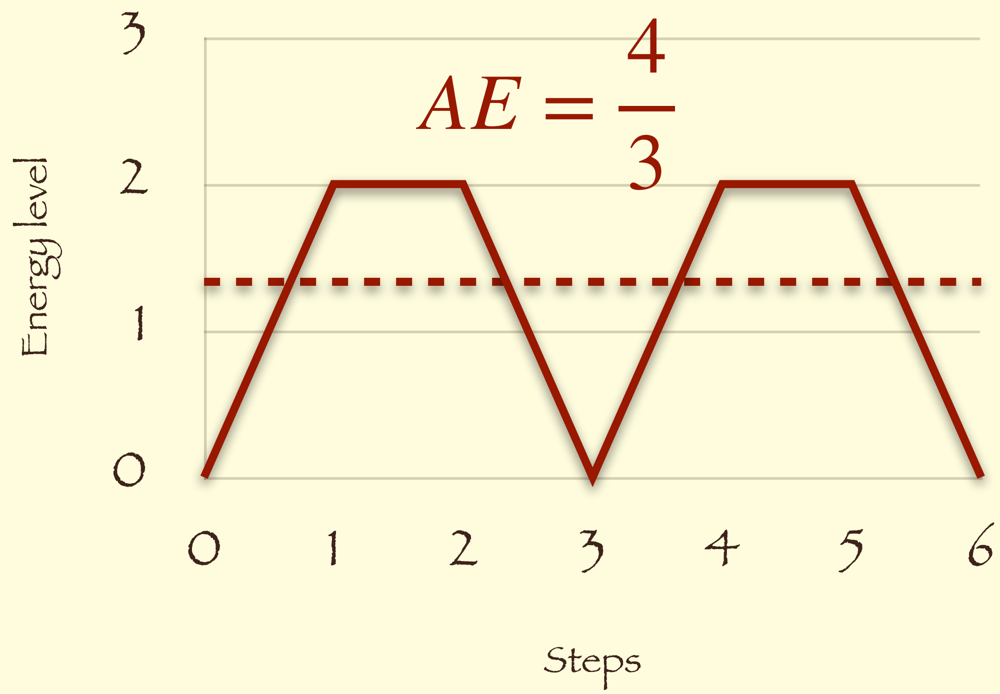
- Constraint on the energy level (EL)
- Mean-payoff (MP): long-run average payoff per transition
- Total-payoff (TP)

Payoffs based on energy



Focus on two memoryless strategies

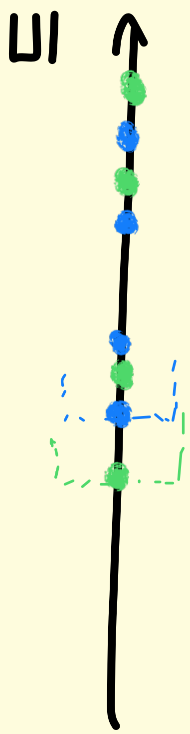




- Constraint on the energy level (EL)
- Mean-payoff (MP): long-run average payoff per transition
- Total-payoff (TP)
- Average-energy (AE)

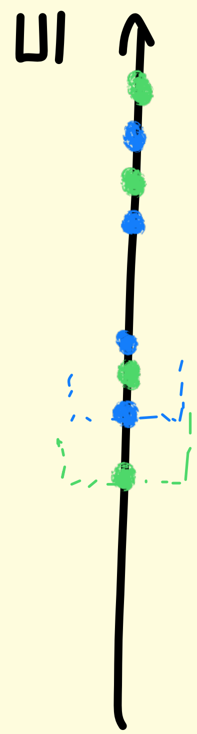
Optimality of strategies

Optimality of strategies



$\mathrm{Out}(\sigma_1)^{\uparrow} \subseteq \mathrm{Out}(\sigma'_1)^{\uparrow}$

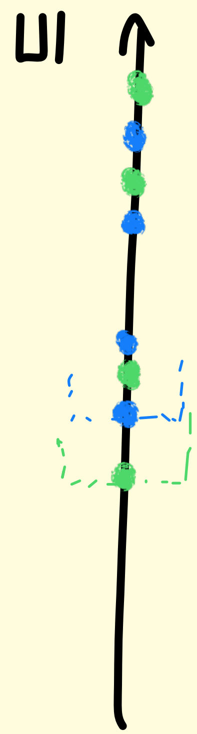
Optimality of strategies



$Out(\sigma_1)^{\uparrow} \subseteq Out(\sigma'_1)^{\uparrow}$

$\Rightarrow$ σ_1 is better than σ'_1

Optimality of strategies

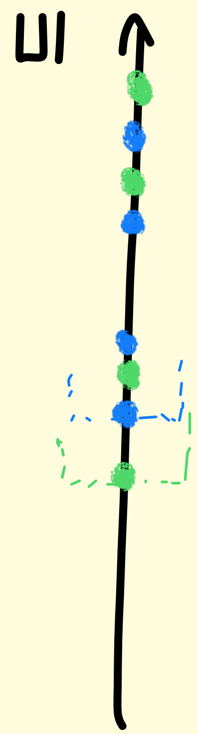


$$Out(\sigma_1)^{\uparrow} \subseteq Out(\sigma'_1)^{\uparrow}$$

$\Rightarrow$ σ_1 is better than σ'_1

σ_1 optimal whenever it is better than any other σ'_1

Optimality of strategies



$\mathrm{Out}(\sigma_1)^{\uparrow} \subseteq \mathrm{Out}(\sigma'_1)^{\uparrow}$

$\Rightarrow$ σ_1 is better than σ'_1

σ_1 optimal whenever it is better than any other σ'_1

Remark

- To be distinguished from:
  - ϵ-optimal
  - Subgame-perfect optimal (in our case: Nash equilibria)

A focus on memoryless strategies

When are memoryless strategies sufficient to play optimally?

Quite often!

When are memoryless strategies sufficient to play optimally?

Quite often!

Examples

- Reachability, safety, Büchi, parity, MP, EL ≥ 0, TP, AE, etc…

When are memoryless strategies sufficient to play optimally?

Quite often!

Examples

- Reachability, safety, Büchi, parity, MP, EL ≥ 0, TP, AE, etc...

Can we characterize when they are?

When are memoryless strategies sufficient to play optimally?

Quite often!

Examples

- Reachability, safety, Büchi, parity, MP, EL ≥ 0, TP, AE, etc...

Can we characterize when they are?

YES !

When are memoryless strategies sufficient to play optimally?

Quite often!

Examples

- Reachability, safety, Büchi, parity, MP, EL ≥ 0, TP, AE, etc...

Can we characterize when they are?

YES !

And this is a beautiful result by Gimbert and Zielonka, CONCUR'05

The memoryless story

Sufficient conditions

The memoryless story

Sufficient conditions

- Sufficient conditions to guarantee memoryless optimal strategies for both player [GZ04,AR17]

The memoryless story

Sufficient conditions

- Sufficient conditions to guarantee memoryless optimal strategies for both player [GZ04,AR17]
- Sufficient conditions to guarantee memoryless optimal strategies for one player (« half-positional ») [Kop06,Gim07,GK14]

The memoryless story

Sufficient conditions

- Sufficient conditions to guarantee memoryless optimal strategies for both player [GZ04,AR17]
- Sufficient conditions to guarantee memoryless optimal strategies for one player (« half-positional ») [Kop06,Gim07,GK14]

- Characterization of the preference relations admitting optimal memoryless strategies for both players in all finite games [GZ05]

The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever

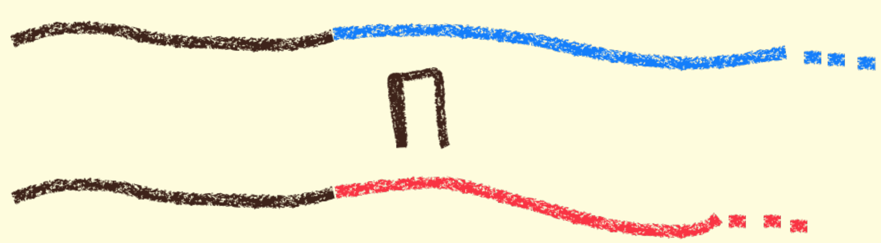
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



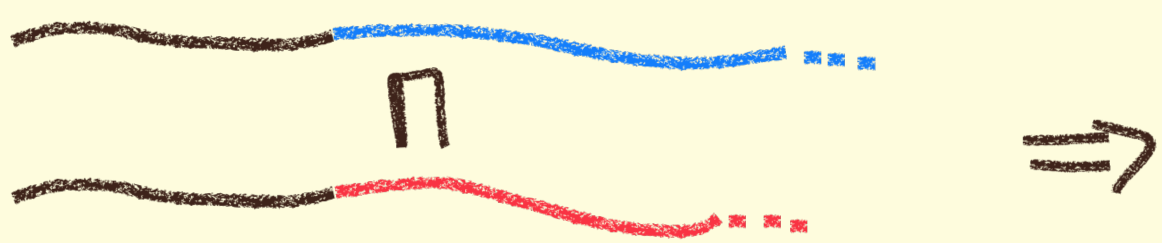
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



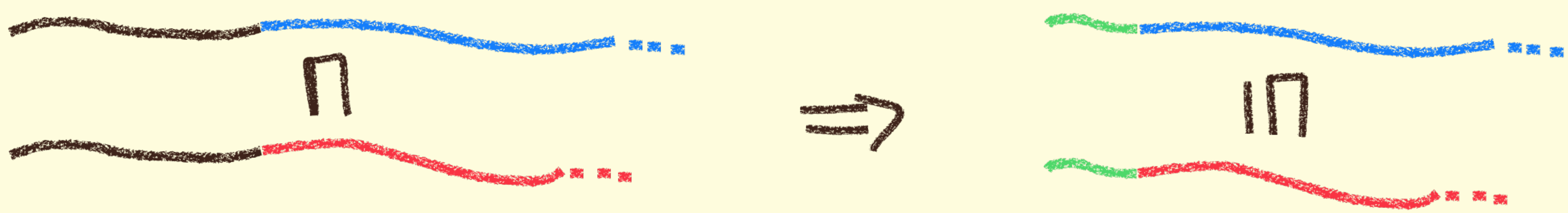
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



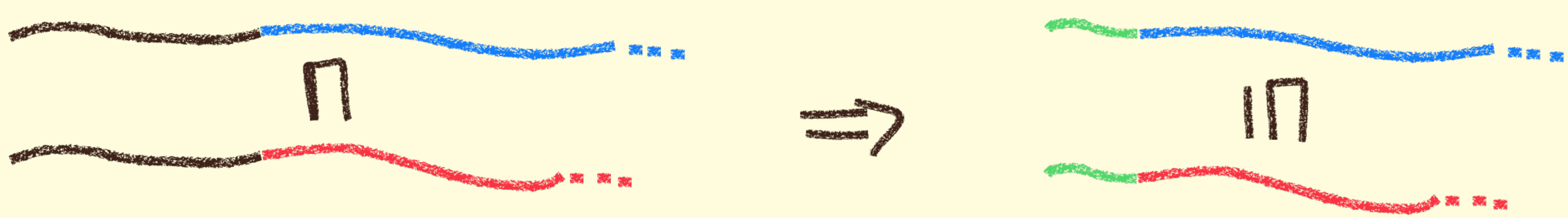
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



- selective whenever

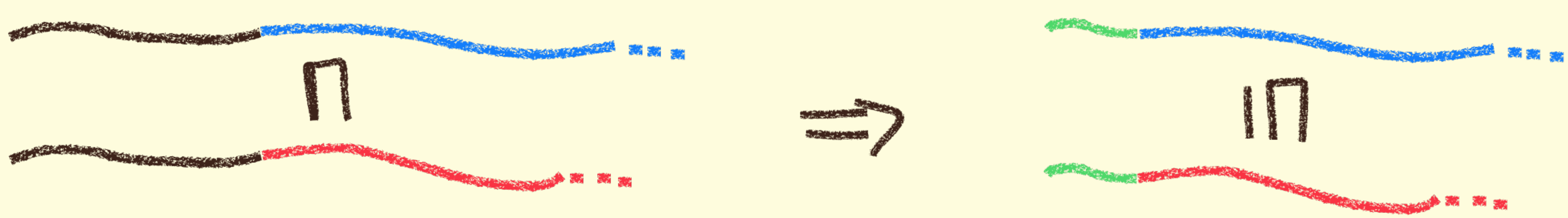
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

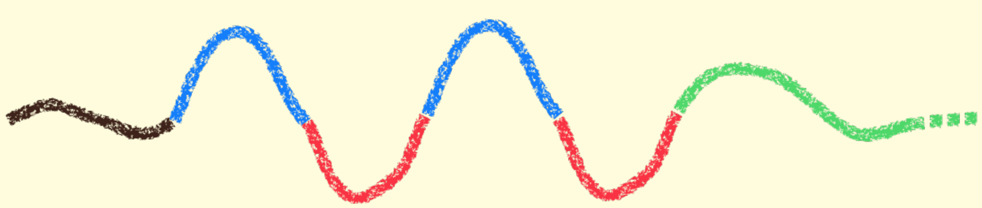
Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



- selective whenever



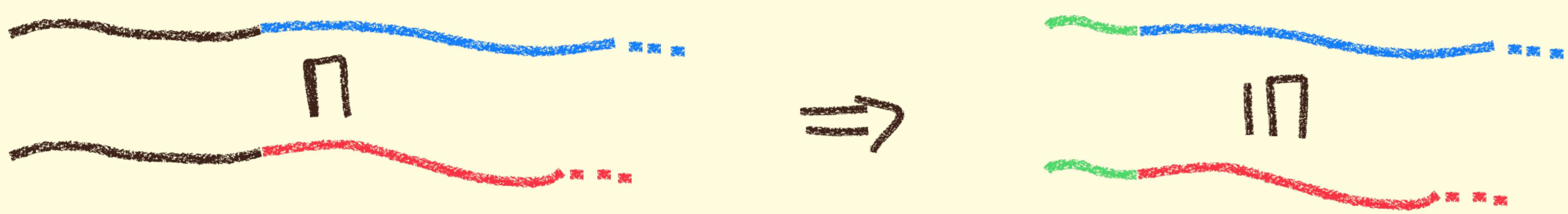
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

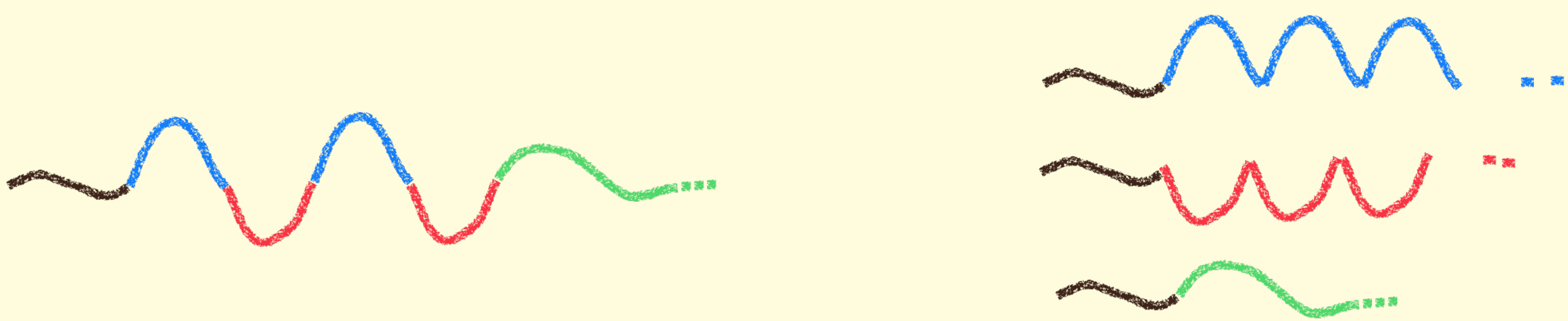
Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



- selective whenever

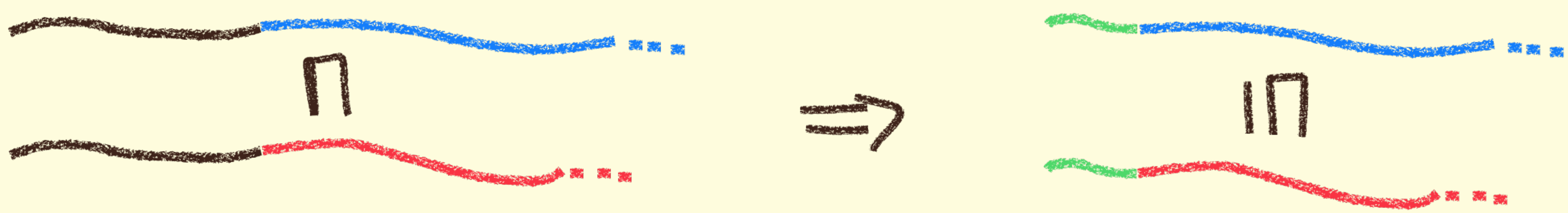


The Gimbert-Zielonka characterization for memory less determinacy (1) [GZ05]

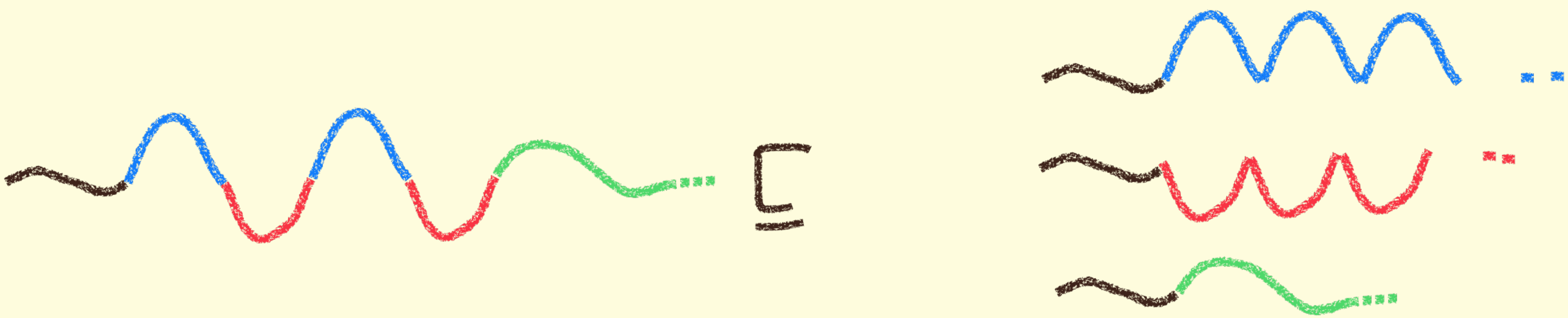
Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



- selective whenever



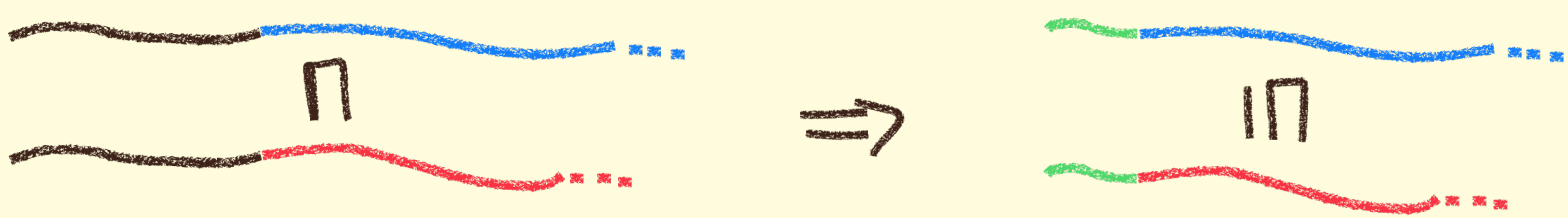
The Gimbert-Zielonka characterization for memory less determinacy (1)

[GZ05]

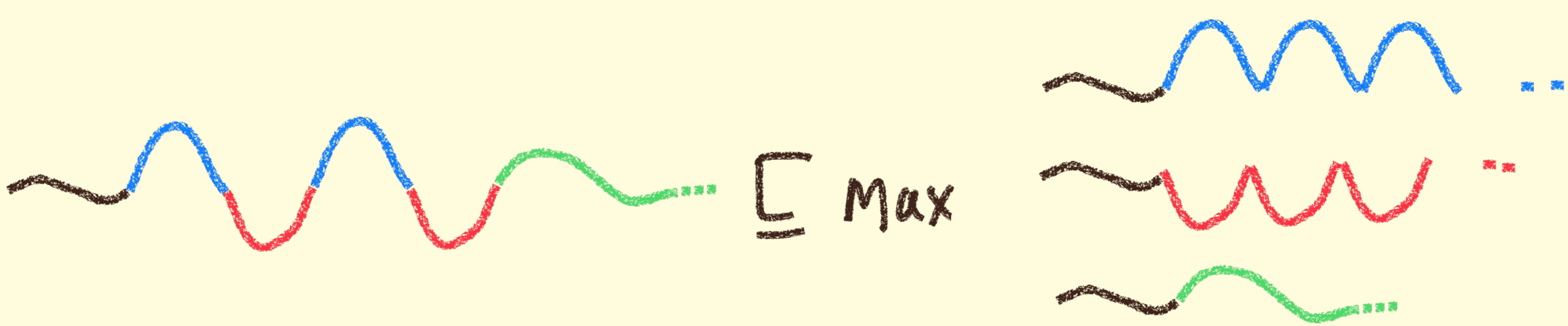
Let $\sqsubseteq$ be a preference relation.

It is said :

- monotone whenever



- selective whenever



The Gimbert-Zielonka characterization for memory less determinacy (2)

[GZ05]

The Gimbert-Zielonka characterization for memory less determinacy (2)

[GZ05]

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have memoryless optimal strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are monotone and selective

The Gimbert-Zielonka characterization for memory less determinacy (2)

[GZ05]

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have memoryless optimal strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are monotone and selective

Characterization - One-player games

The two following assertions are equivalent :

1. All finite P_1-games have (uniform) memoryless optimal strategies
2. $\sqsubseteq$ is monotone and selective

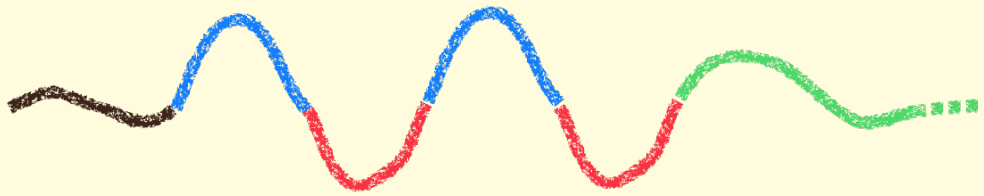
Why? Proof hint (1)

Why? Proof hint (1)

Assume all P_1-games have optimal memoryless strategies.

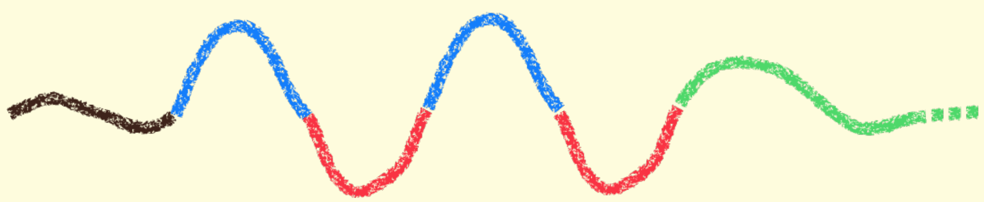
Why? Proof hint (1)

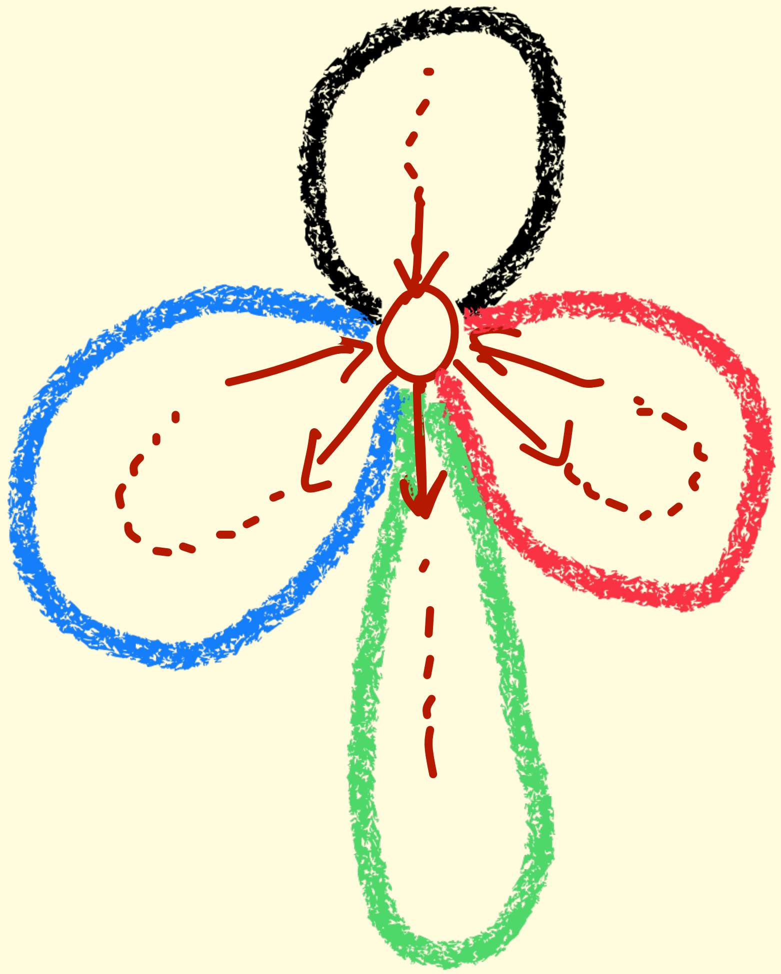
Assume all P_1-games have optimal memoryless strategies.



Why? Proof hint (1)

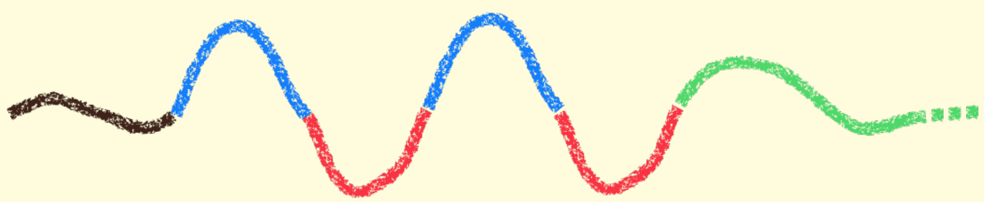
Assume all P_1-games have optimal memoryless strategies.



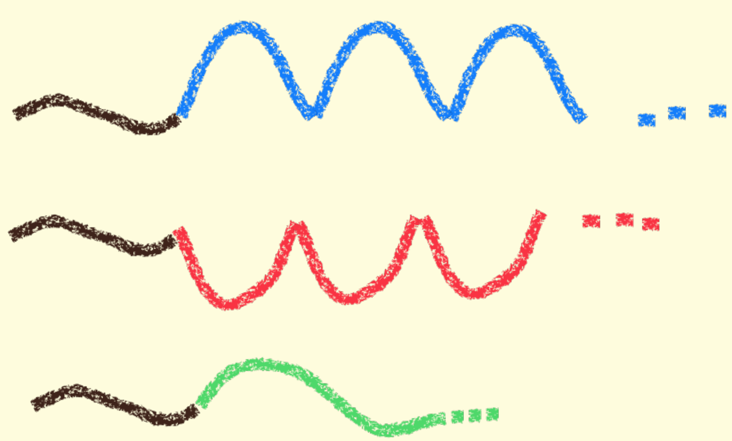


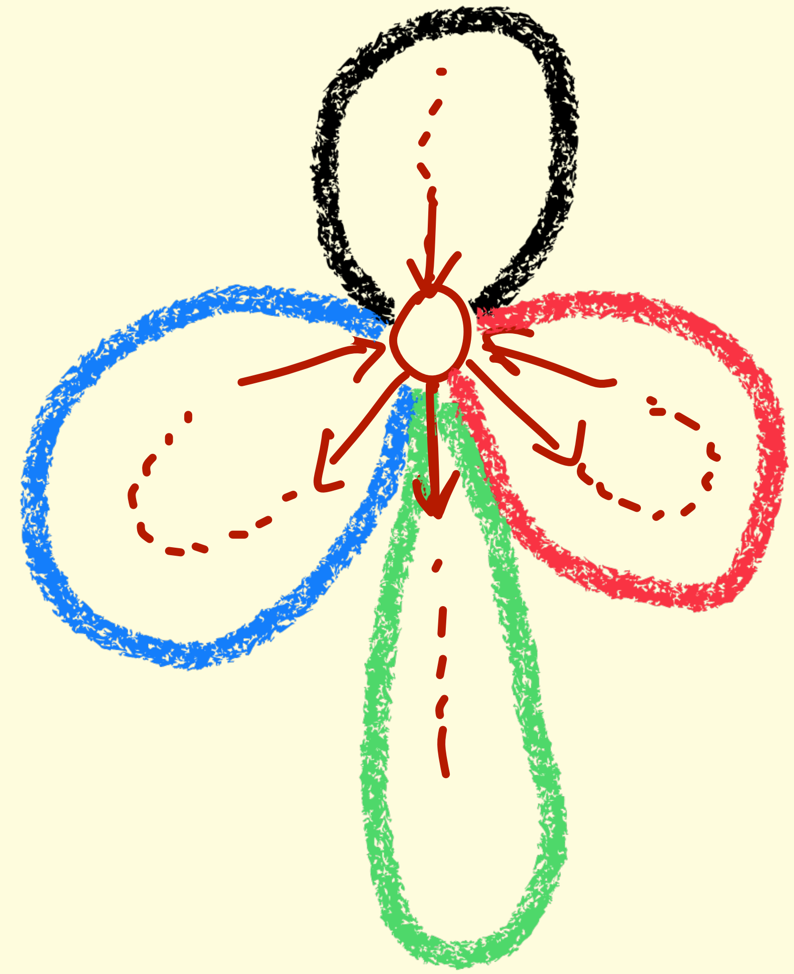
Why? Proof hint (1)

Assume all P_1-games have optimal memoryless strategies.



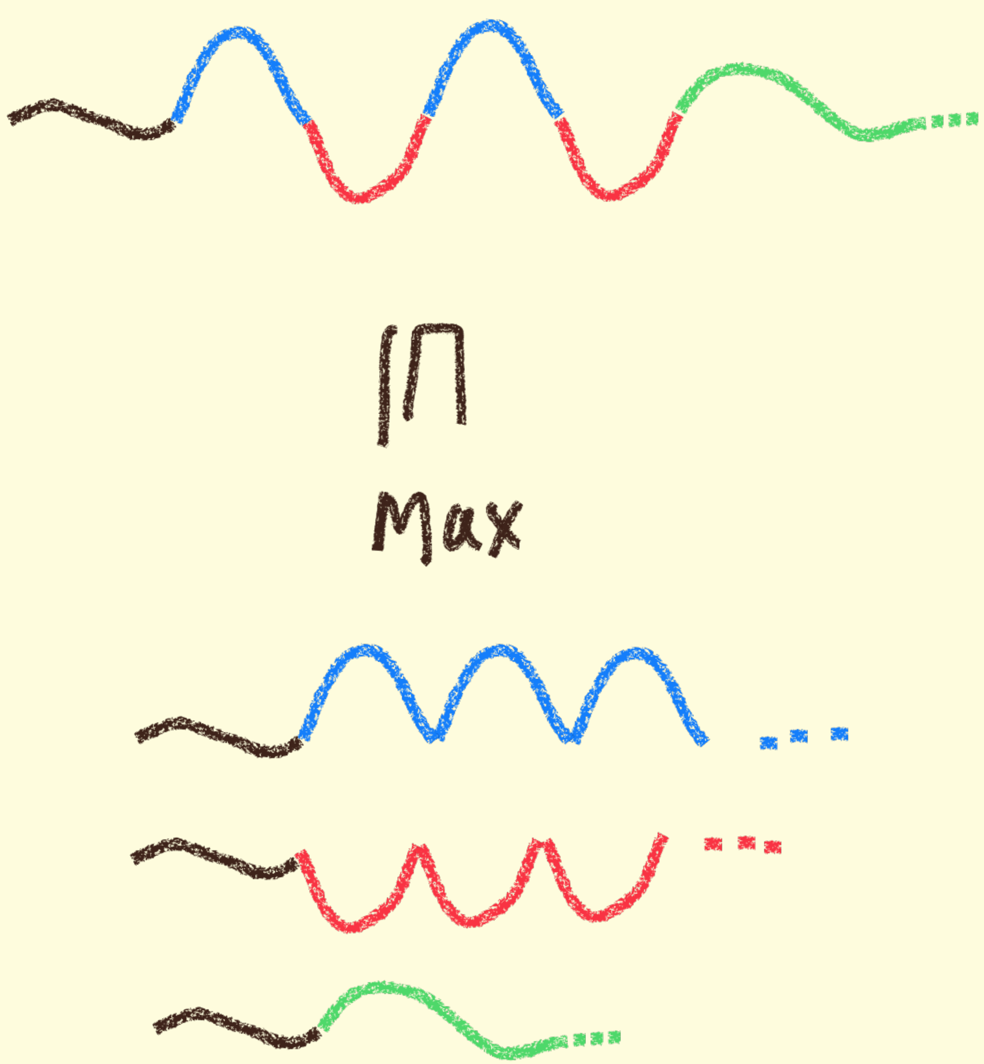
Max

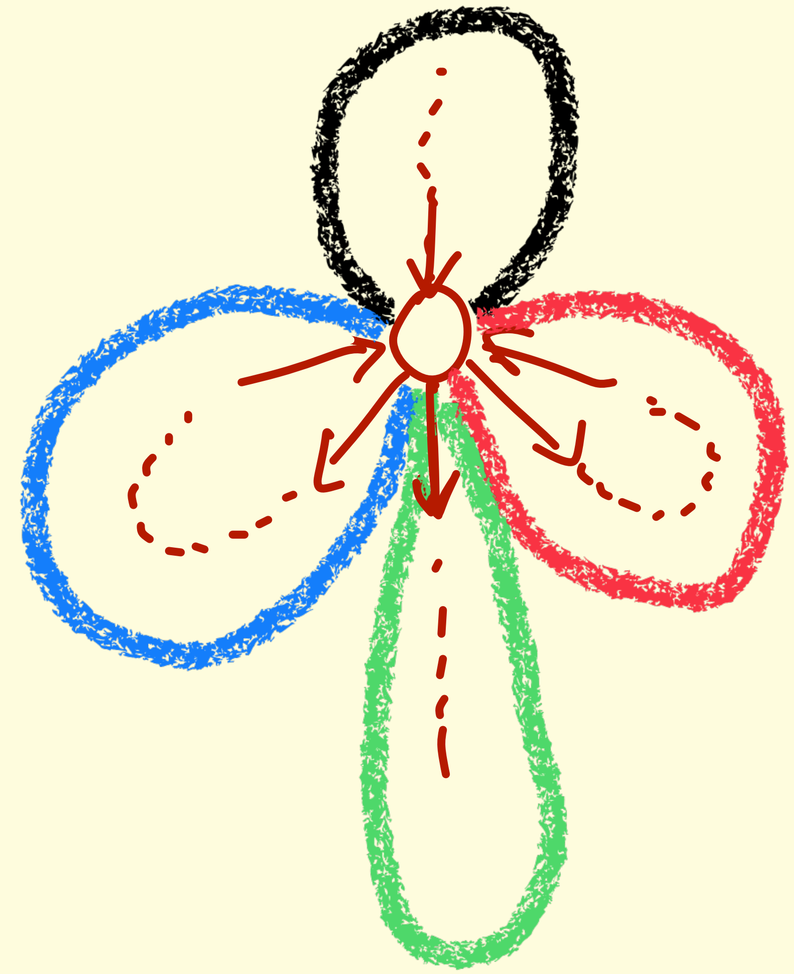




Why? Proof hint (1)

Assume all P_1-games have optimal memoryless strategies.



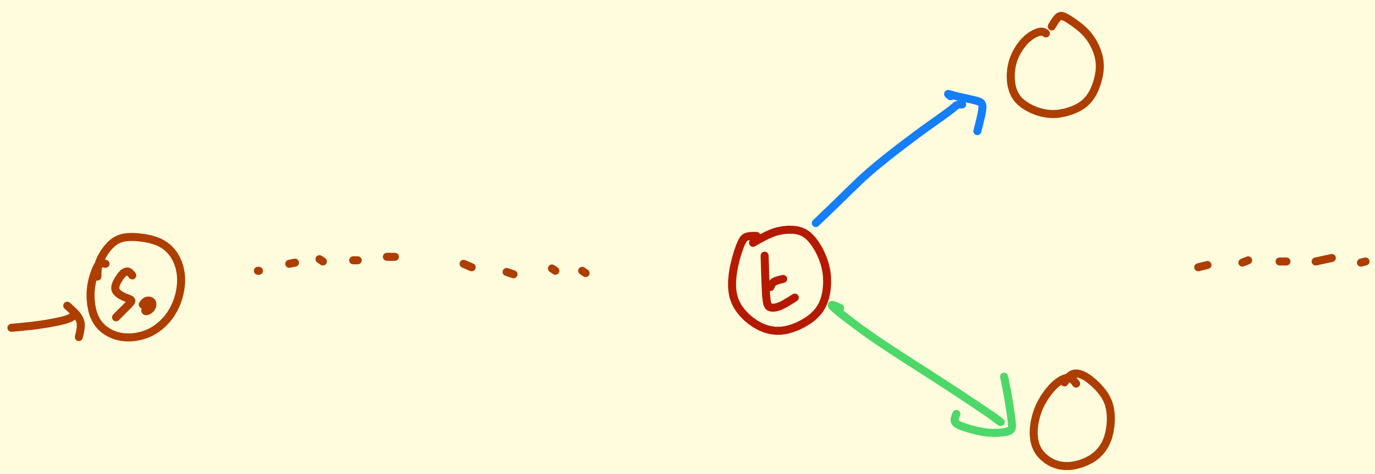


$\sqsubseteq$ is selective

Why? Proof hint (2)

Assume $\sqsubseteq$ is monotone and selective.

The case of one-player games



one best choice between and (monotony)

+ no reason to swap at t (selectivity)

No memory required at t!

Applications

Lifting theorem

- If in all finite one-player game for player P_i, P_i has uniform memoryless optimal strategies, then both players have memoryless optimal strategies in all finite two-player games.

Applications

Lifting theorem

- If in all finite one-player game for player P_i, P_i has uniform memoryless optimal strategies, then both players have memoryless optimal strategies in all finite two-player games.

Very powerful and extremely useful in practice!

Applications

Lifting theorem

- If in all finite one-player game for player P_i, P_i has uniform memoryless optimal strategies, then both players have memoryless optimal strategies in all finite two-player games.

Very powerful and extremely useful in practice!

Discussion

- Easy to analyse the one-player case (graph analysis)
  - Mean-payoff, average-energy [BMRLL15]
- Allows to deduce properties in the two-player case

Discussion of examples

Examples

Discussion of examples

Examples

- Reachability, safety:

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:
  - Prefix-independent hence monotone

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:
  - Prefix-independent hence monotone
  - Selective

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:
  - Prefix-independent hence monotone
  - Selective
- Priority mean payoff [GZ05]

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:
  - Prefix-independent hence monotone
  - Selective
- Priority mean payoff [GZ05]
- Average-energy games [BMRLL15]

Discussion of examples

Examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective
- Parity, mean-payoff:
  - Prefix-independent hence monotone
  - Selective
- Priority mean payoff [GZ05]
- Average-energy games [BMRLL15]
  - Lifting theorem!!

Discussion

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

$$\limsup_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} c_i \in \mathbb{Q}$$

$$\liminf_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} c_i \in \mathbb{Q}$$

Discussion

Winning condition for P_1:

$$((\mathsf{MP} \in \mathbb{Q}) \land \text{Büchi}(A)) \lor \text{coBüchi}(\boldsymbol{B})$$

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \mathrm{Büchi}(A)) \vee \mathrm{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \mathrm{Büchi}(A)) \vee \mathrm{coBüchi}(B)$$

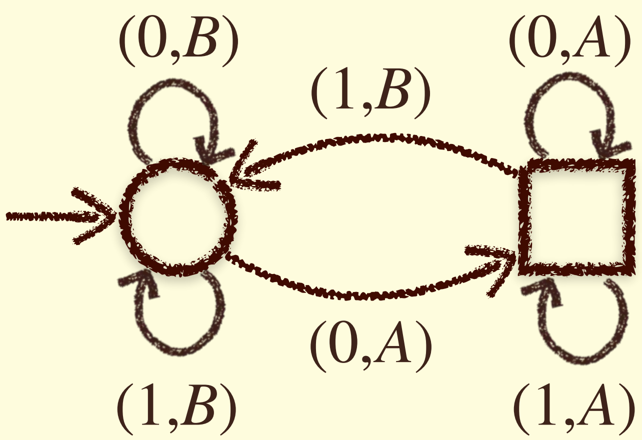
- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective

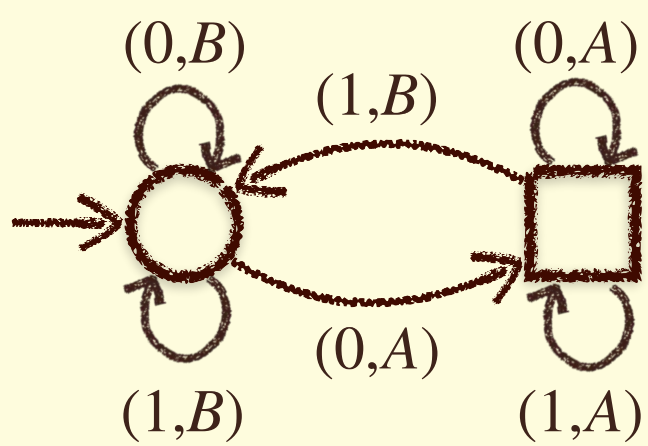


Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



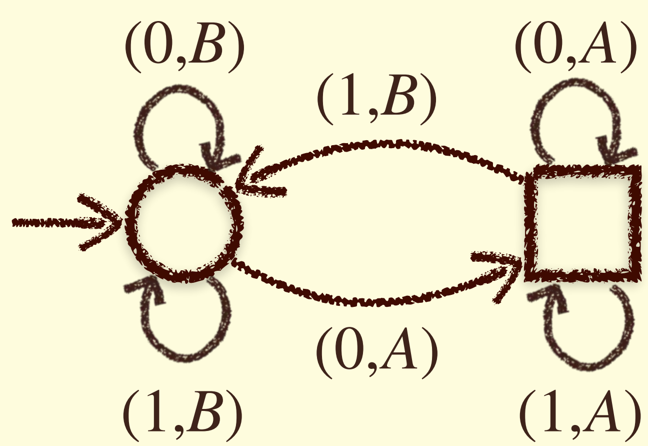
How should P_1 play this game?

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



How should P_1 play this game?

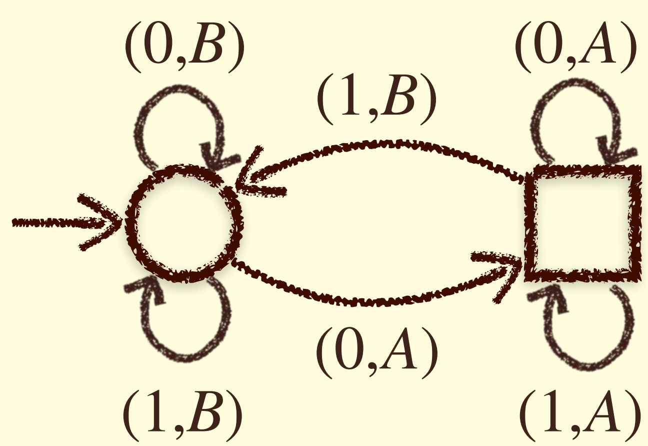
- P_1 wins this game:

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



How should P_1 play this game?

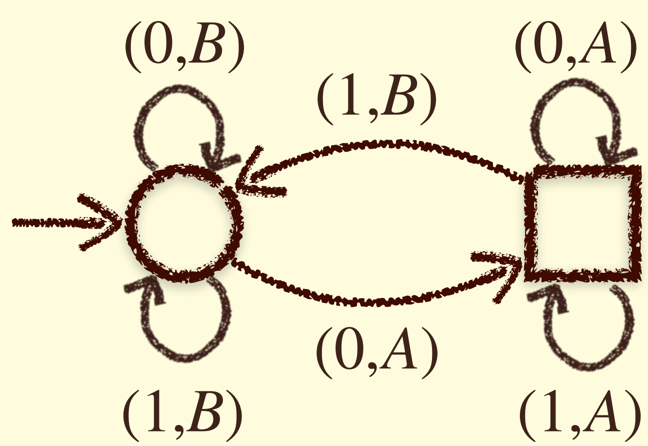
- P_1 wins this game:
  - Infinitely often, give the hand back to P_2

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



How should P_1 play this game?

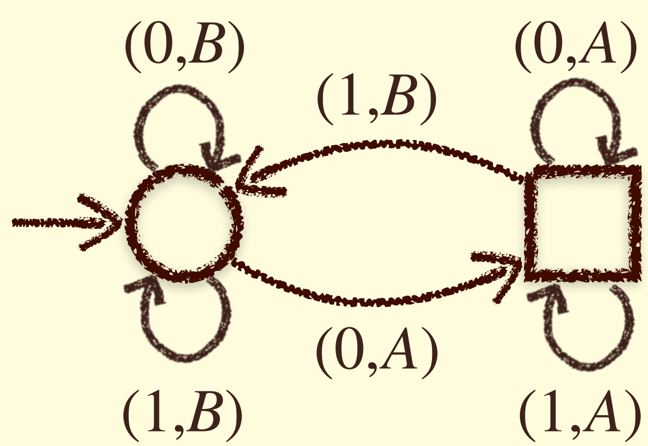
- P_1 wins this game:
  - Infinitely often, give the hand back to P_2
  - Play for a long time the edge labelled $(0,B)$ to approach 0

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



How should P_1 play this game?

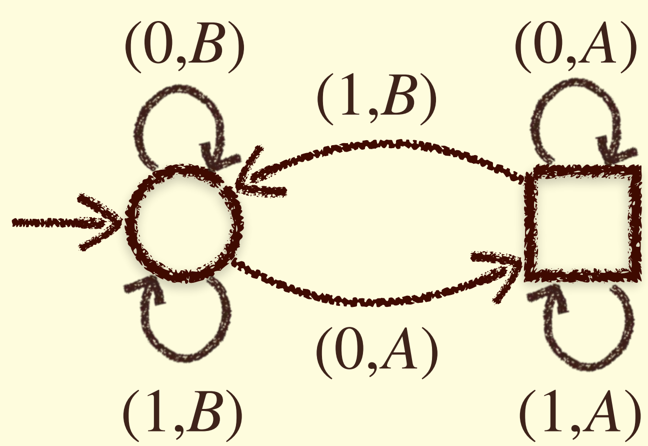
- P_1 wins this game:
  - Infinitely often, give the hand back to P_2
  - Play for a long time the edge labelled $(0,B)$ to approach 0
  - Play for a long time the edge labelled $(1,B)$ to approach 1

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

- In all one-player games, P_1 has a memoryless uniform optimal strategy
- Hence: the winning condition is monotone and selective



How should P_1 play this game?

- P_1 wins this game:
  - Infinitely often, give the hand back to P_2
  - Play for a long time the edge labelled $(0,B)$ to approach 0
  - Play for a long time the edge labelled $(1,B)$ to approach 1
- It requires infinite memory!

Discussion

Winning condition for P_1:

$$((\mathrm{MP} \in \mathbb{Q}) \wedge \text{Büchi}(A)) \vee \text{coBüchi}(B)$$

If only $\sqsubseteq$ is monotone and selective, P_1 might not have a memoryless optimal strategy

Finite-memory strategies

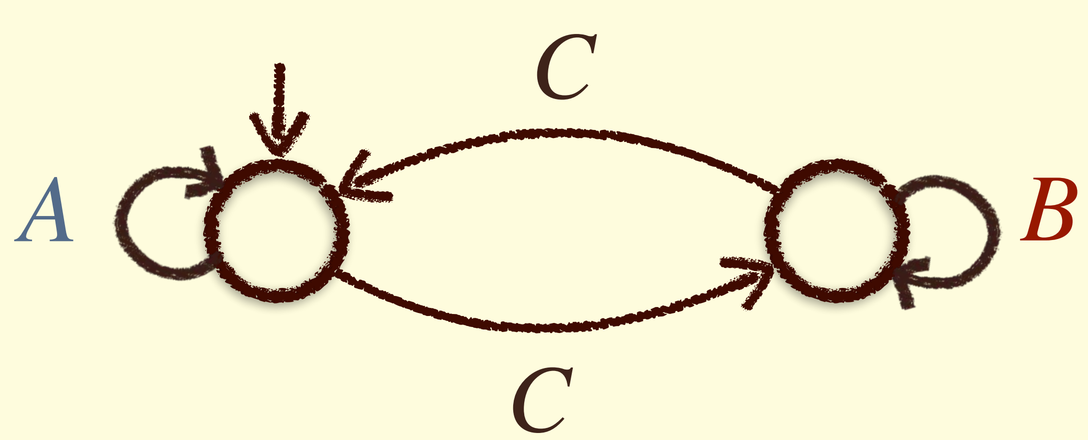
We need memory!

Objectives/preference relations become more and more complex

We need memory!

Objectives/preference relations become more and more complex

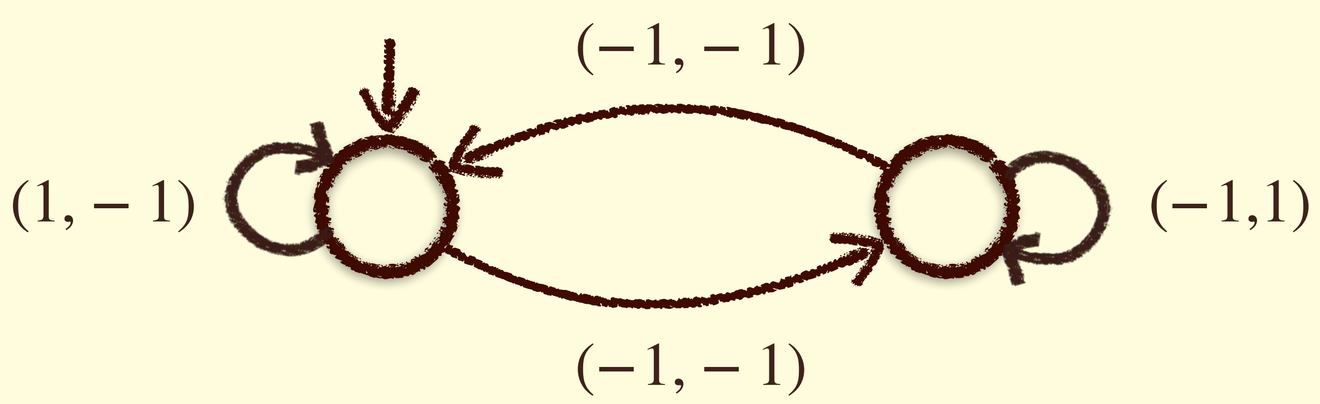
- Büchi(A)$\wedge$Büchi(B) requires finite memory



We need memory!

Objectives/preference relations become more and more complex

- Büchi(A)$\wedge$Büchi(B) requires finite memory
- $MP_1 \geq 0 \wedge MP_2 \geq 0$ requires infinite memory



Can we lift [GZ05] to finite memory?

Can we lift [GZ05] to finite memory?

A priori no...

Can we lift [GZ05] to finite memory?

A priori no...

Consider the following winning condition for P_1:

$$\liminf_n \sum_{i=1}^{n} c_i = +\infty \quad \text{or} \quad \exists^\infty n \text{ s.t. } \sum_{i=1}^{n} c_i = 0$$

Can we lift [GZ05] to finite memory?

A priori no…

Consider the following winning condition for P_1:

$$\liminf_{n} \sum_{i=1}^{n} c_i = +\infty \quad \text{or} \quad \exists^{\infty} n \text{ s.t. } \sum_{i=1}^{n} c_i = 0$$

- Optimal finite-memory strategies in one-player games

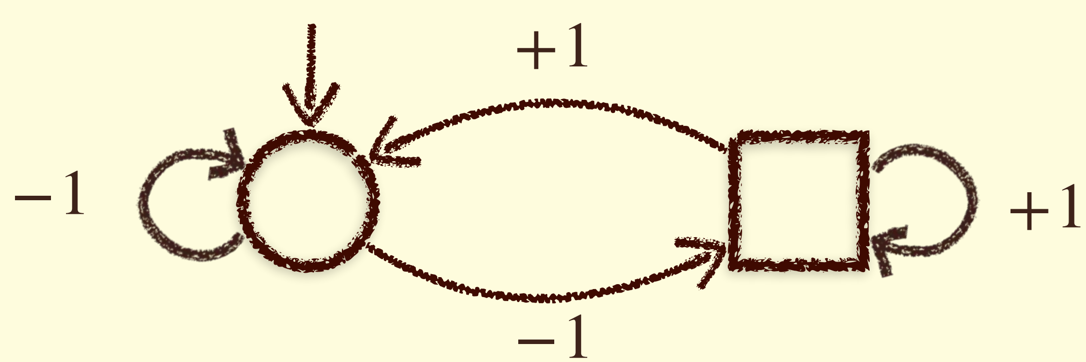
Can we lift [GZ05] to finite memory?

A priori no...

Consider the following winning condition for P_1:

$$\liminf_{n} \sum_{i=1}^{n} c_i = +\infty \quad \text{or} \quad \exists^{\infty} n \text{ s.t. } \sum_{i=1}^{n} c_i = 0$$

- Optimal finite-memory strategies in one-player games
- But not in two-player games!!



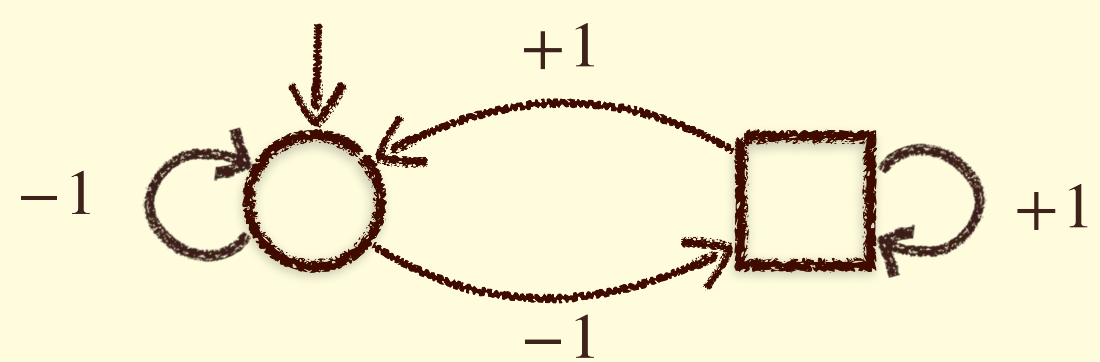
Can we lift [GZ05] to finite memory?

A priori no...

Consider the following winning condition for P_1:

$$\liminf_{n} \sum_{i=1}^{n} c_i = +\infty \quad \text{or} \quad \exists^{\infty} n \text{ s.t. } \sum_{i=1}^{n} c_i = 0$$

- Optimal finite-memory strategies in one-player games
- But not in two-player games!!



P_1 wins but uses infinite memory!

How do we formalize finite memory?

How do we formalize finite memory?

Standardly

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}}, \alpha_{\mathsf{next}})$ where M is finite, $m_{\mathsf{init}} \in M$, $\alpha_{\mathsf{upd}} : M \times S \to M$ and $\alpha_{\mathsf{next}} : M \times S_i \to E$
  - $(M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ is a memory mechanism

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$
  - $(M, m_{\text{init}}, \alpha_{\text{upd}})$ is a memory mechanism
  - α_{next} gives the next move

How do we formalize finite memory?

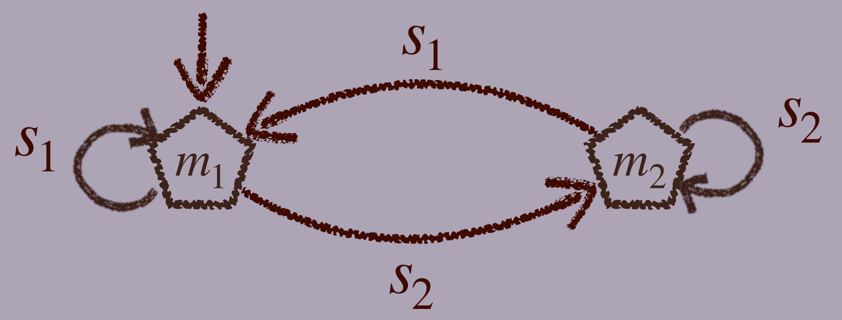
Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$
  - $(M, m_{\text{init}}, \alpha_{\text{upd}})$ is a memory mechanism
  - α_{next} gives the next move

How do we formalize finite memory?

Standardly

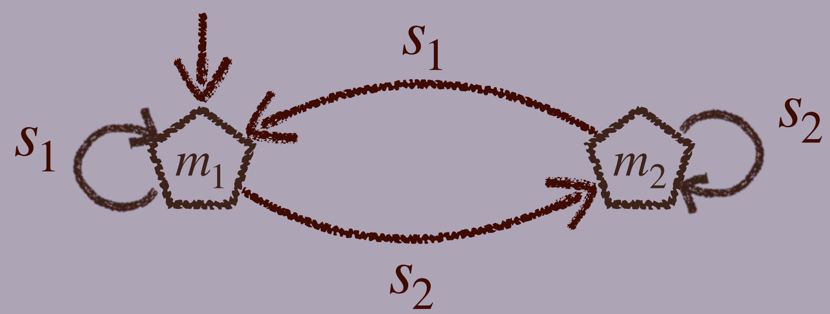
- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}}, \alpha_{\mathsf{next}})$ where M is finite, $m_{\mathsf{init}} \in M$, $\alpha_{\mathsf{upd}} : M \times S \to M$ and $\alpha_{\mathsf{next}} : M \times S_i \to E$
  - $(M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ is a memory mechanism
  - α_{next} gives the next move



How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$
  - $(M, m_{\text{init}}, \alpha_{\text{upd}})$ is a memory mechanism
  - α_{next} gives the next move

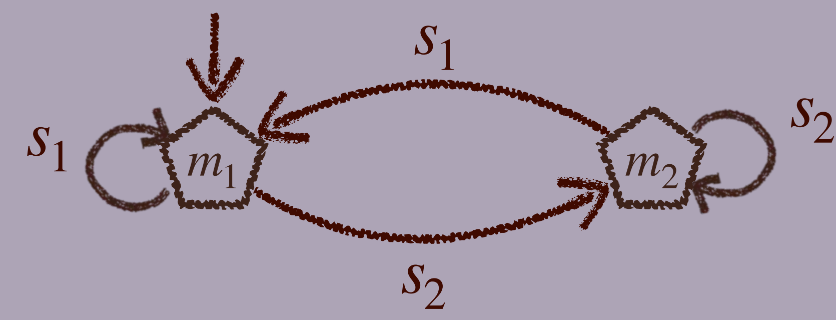


To have an abstract theorem…

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$
  - $(M, m_{\text{init}}, \alpha_{\text{upd}})$ is a memory mechanism
  - α_{next} gives the next move



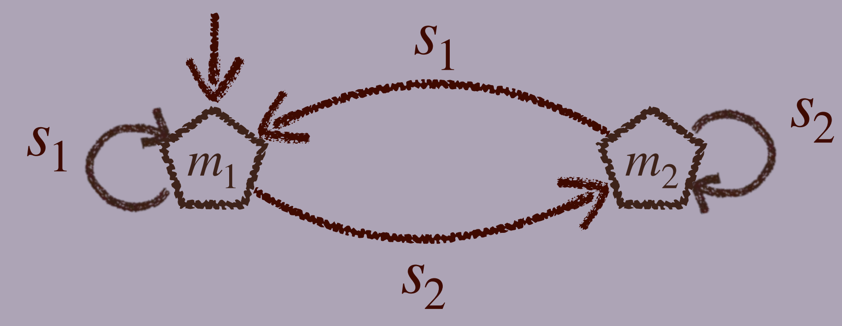
To have an abstract theorem…

- The memory mechanism should not speak about information specific to particular games, hence:

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\mathrm{init}}, \alpha_{\mathrm{upd}}, \alpha_{\mathrm{next}})$ where M is finite, $m_{\mathrm{init}} \in M$, $\alpha_{\mathrm{upd}} : M \times S \to M$ and $\alpha_{\mathrm{next}} : M \times S_i \to E$
  - $(M, m_{\mathrm{init}}, \alpha_{\mathrm{upd}})$ is a memory mechanism
  - α_{next} gives the next move



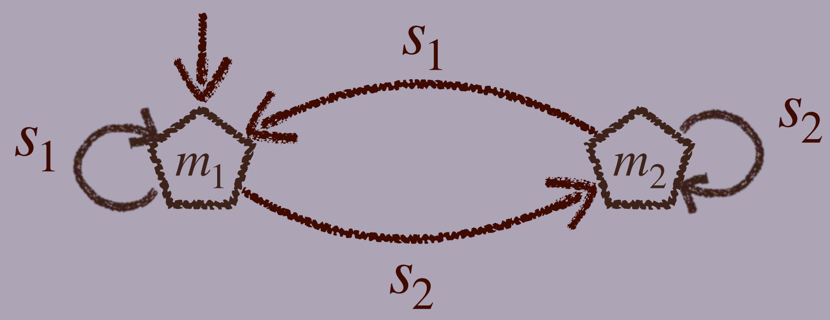
To have an abstract theorem…

- The memory mechanism should not speak about information specific to particular games, hence:
  - α_{upd} should not speak of states

How do we formalize finite memory?

Standardly

- A strategy σ_i of player P_i has finite memory if it can be encoded as a Mealy machine $(M, m_{\text{init}}, \alpha_{\text{upd}}, \alpha_{\text{next}})$ where M is finite, $m_{\text{init}} \in M$, $\alpha_{\text{upd}} : M \times S \to M$ and $\alpha_{\text{next}} : M \times S_i \to E$
  - $(M, m_{\text{init}}, \alpha_{\text{upd}})$ is a memory mechanism
  - α_{next} gives the next move



To have an abstract theorem…

- The memory mechanism should not speak about information specific to particular games, hence:
  - α_{upd} should not speak of states
  - α_{upd} can speak of colors (notion of « chromatic strategy » by Kopczynski)

Arena-independent memory management

Arena-independent memory management

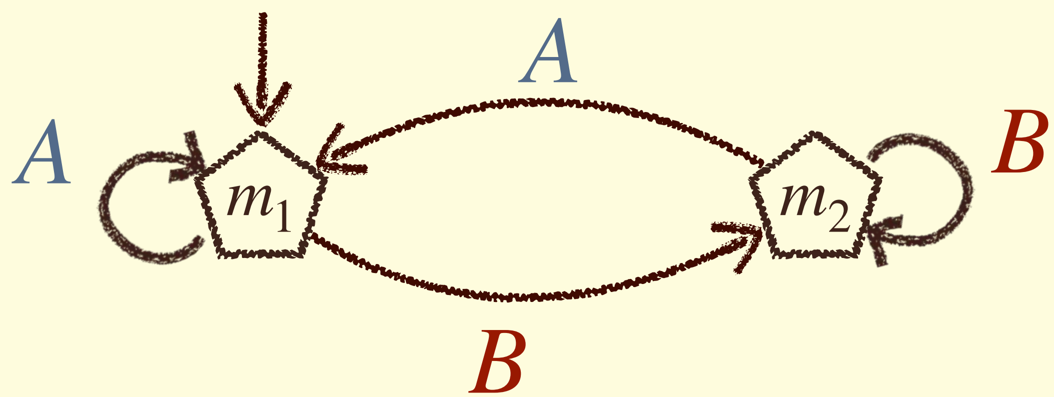
Memory skeleton

- $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ with $m_{\mathsf{init}} \in M$ and $\alpha_{\mathsf{upd}} : M \times C \to M$

Arena-independent memory management

Memory skeleton

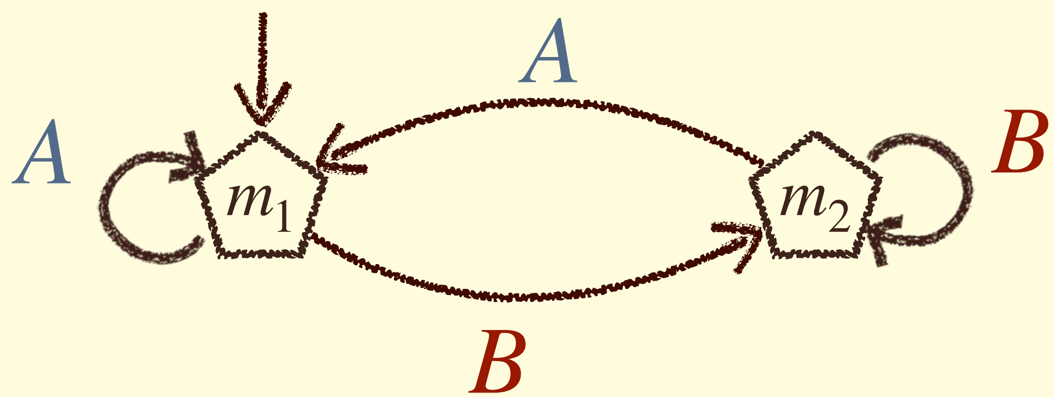
- $\mathscr{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ with $m_{\text{init}} \in M$ and $\alpha_{\text{upd}} : M \times C \to M$



Arena-independent memory management

Memory skeleton

- $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ with $m_{\mathsf{init}} \in M$ and $\alpha_{\mathsf{upd}} : M \times C \to M$

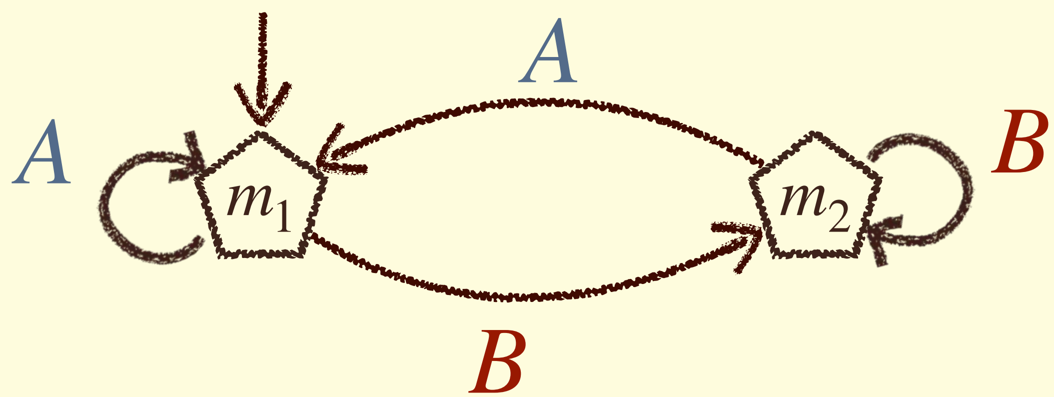


Not yet a strategy!

Arena-independent memory management

Memory skeleton

- $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ with $m_{\mathsf{init}} \in M$ and $\alpha_{\mathsf{upd}} : M \times C \to M$



Not yet a strategy!

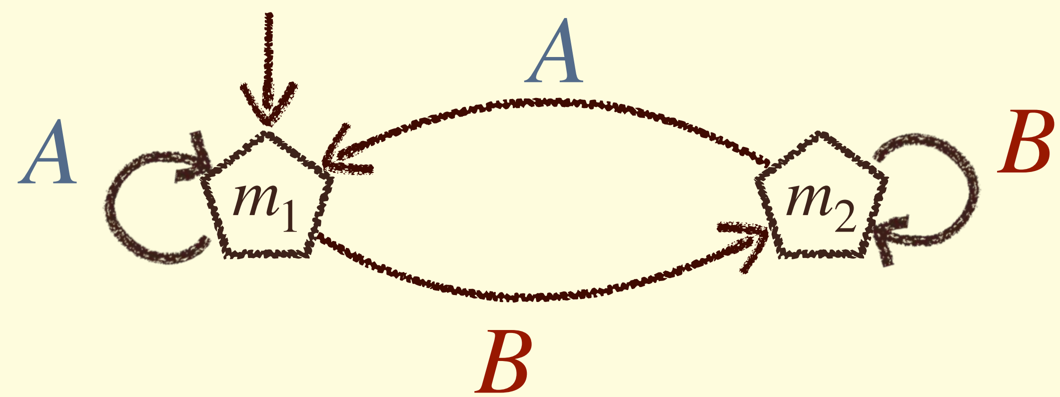
Strategy with memory $\mathcal{M}$

- Additional next-move function: $\alpha_{\mathsf{next}} : M \times S_i \to E$

Arena-independent memory management

Memory skeleton

- $\mathcal{M} = (M, m_{\mathrm{init}}, \alpha_{\mathrm{upd}})$ with $m_{\mathrm{init}} \in M$ and $\alpha_{\mathrm{upd}} : M \times C \to M$



Not yet a strategy!

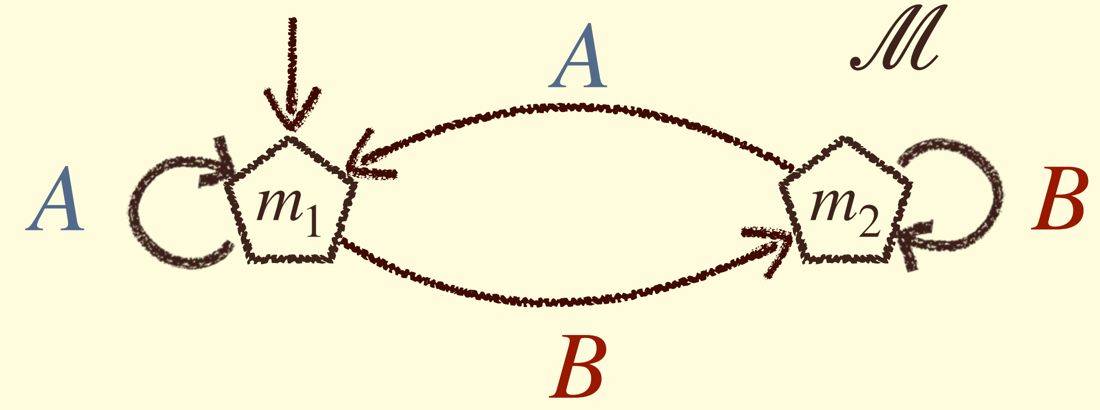
Strategy with memory $\mathcal{M}$

- Additional next-move function: $\alpha_{\mathrm{next}} : M \times S_i \to E$

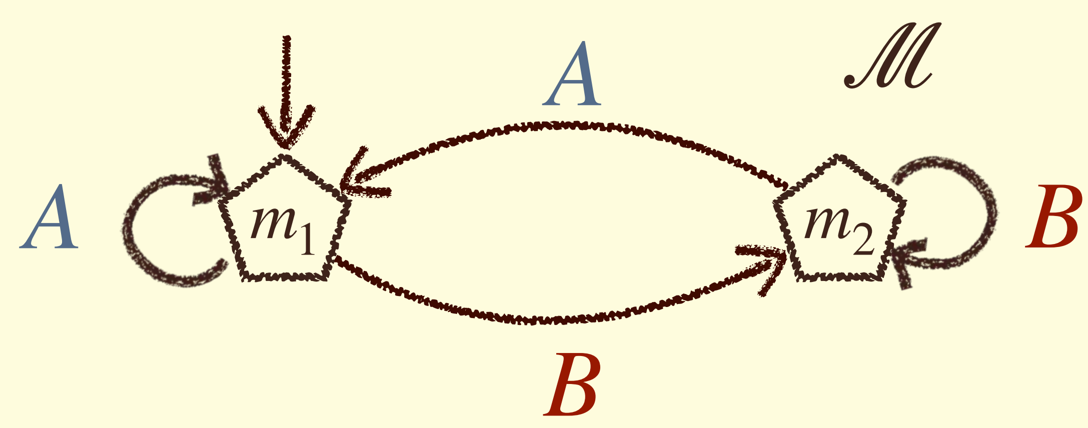
The above skeleton is sufficient for the winning condition

Büchi$(A) \wedge$Büchi(B)

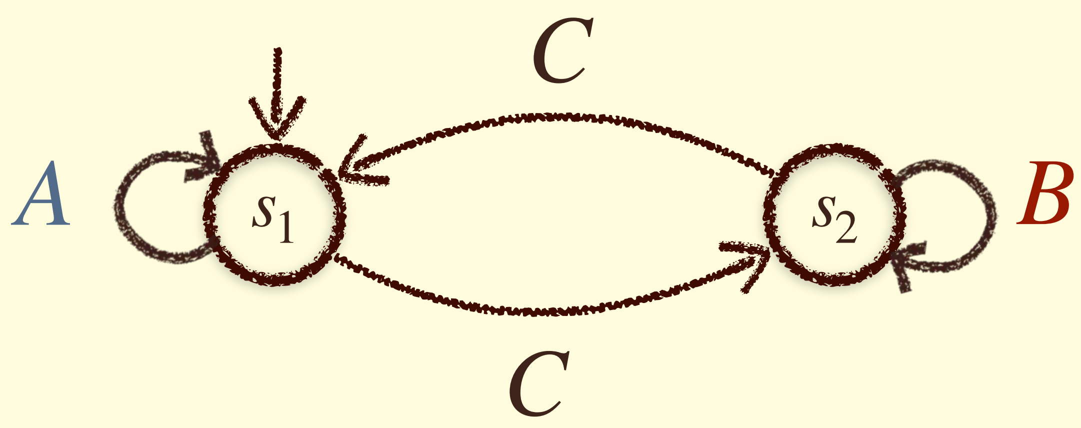
Example



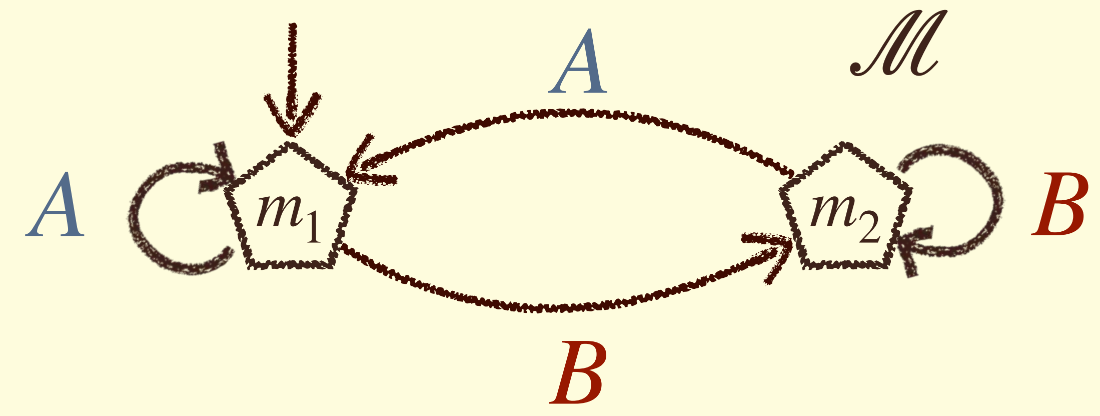
Example



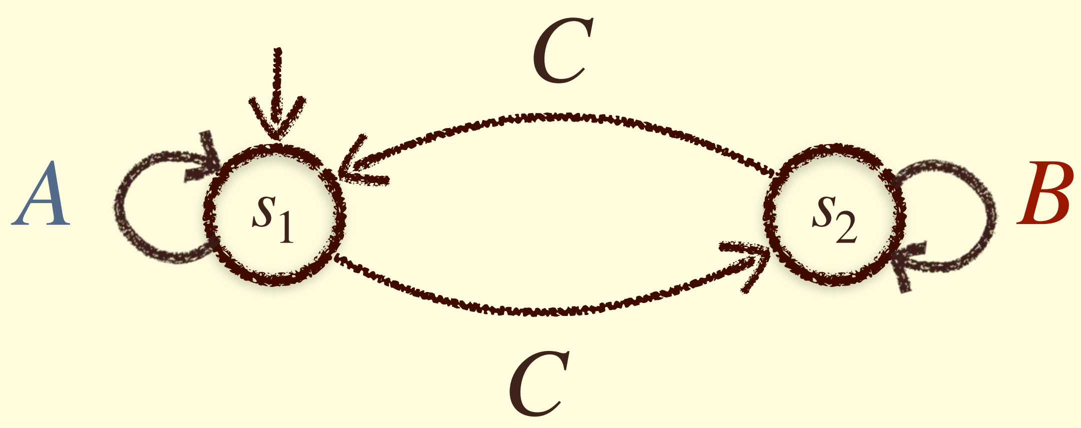
Game arena $\mathscr{A}$:



Example

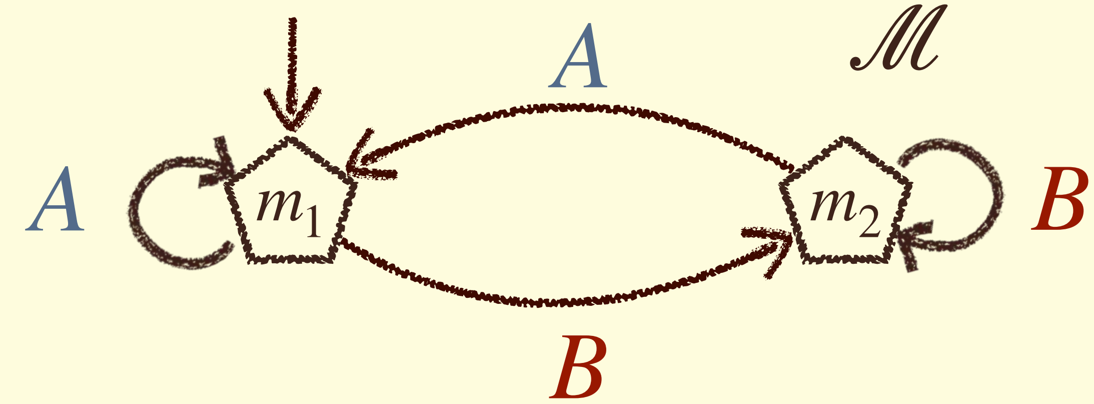


Game arena $\mathscr{A}$:

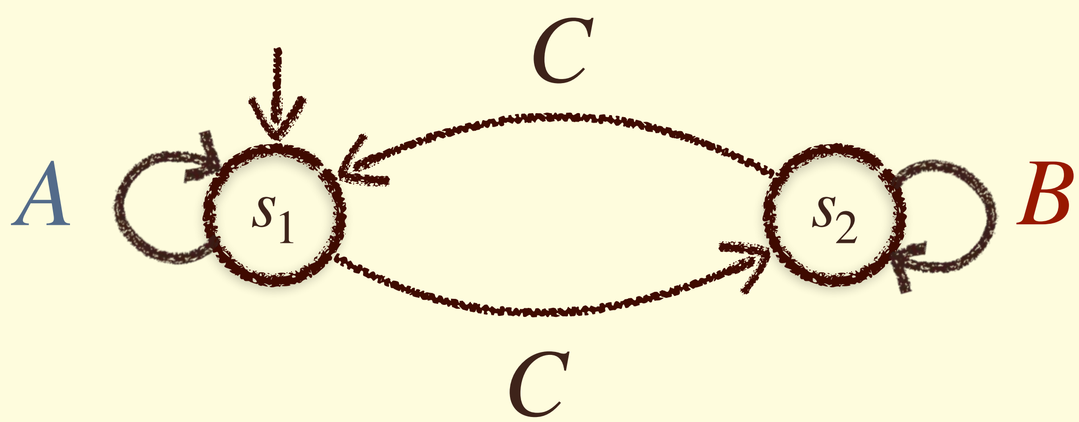


$$(s_1, m_1) \mapsto (s_1, s_2)$$
$$(s_1, m_2) \mapsto (s_1, s_1)$$
$$(s_2, m_1) \mapsto (s_2, s_2)$$
$$(s_2, m_2) \mapsto (s_2, s_1)$$

Example

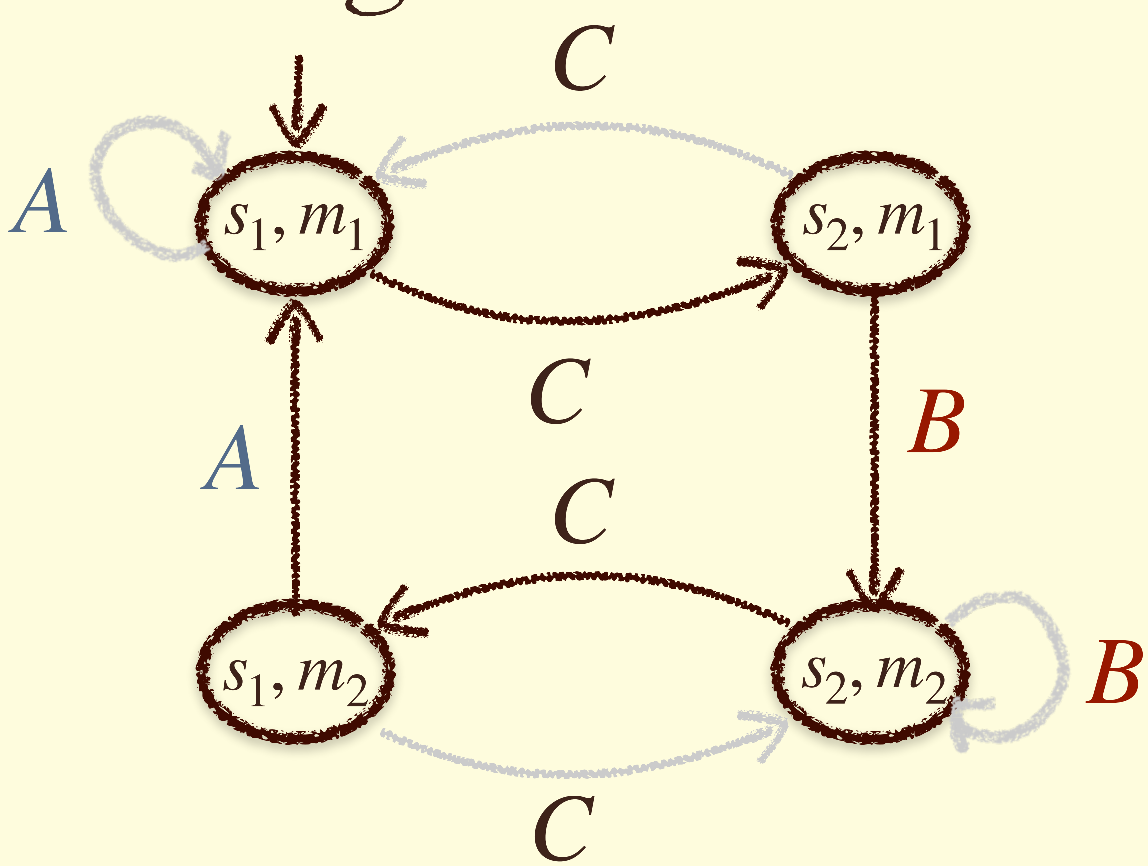


Game arena $\mathscr{A}$:

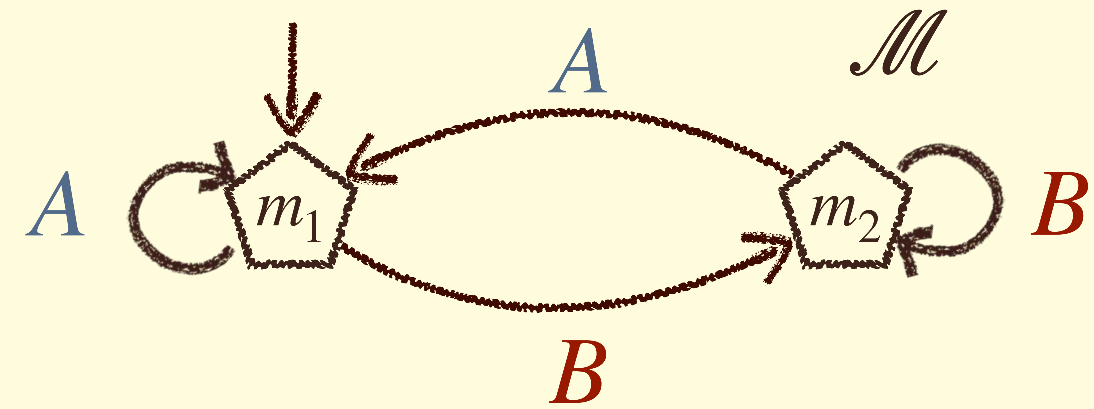


$$
\begin{aligned}
(s_1, m_1) &\mapsto (s_1, s_2)\\
(s_1, m_2) &\mapsto (s_1, s_1)\\
(s_2, m_1) &\mapsto (s_2, s_2)\\
(s_2, m_2) &\mapsto (s_2, s_1)
\end{aligned}
$$

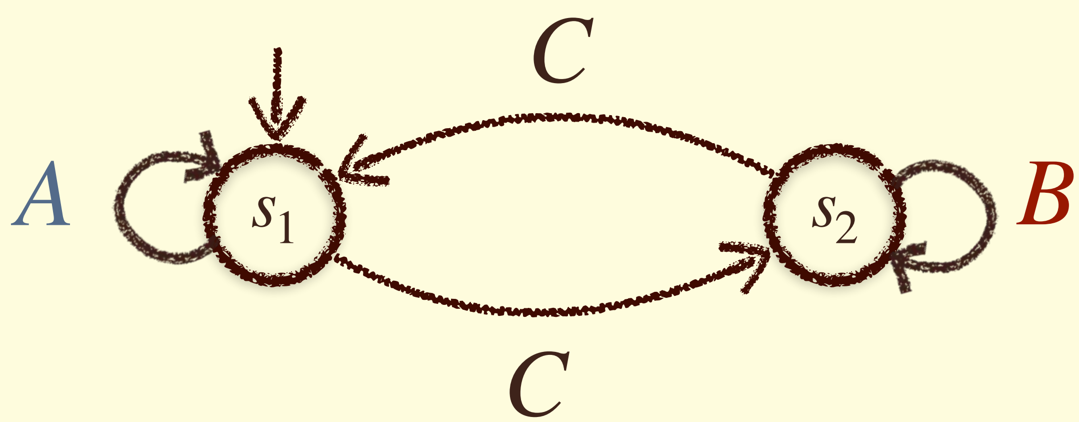
Product game $\mathscr{A} \ltimes \mathscr{M}$:



Example

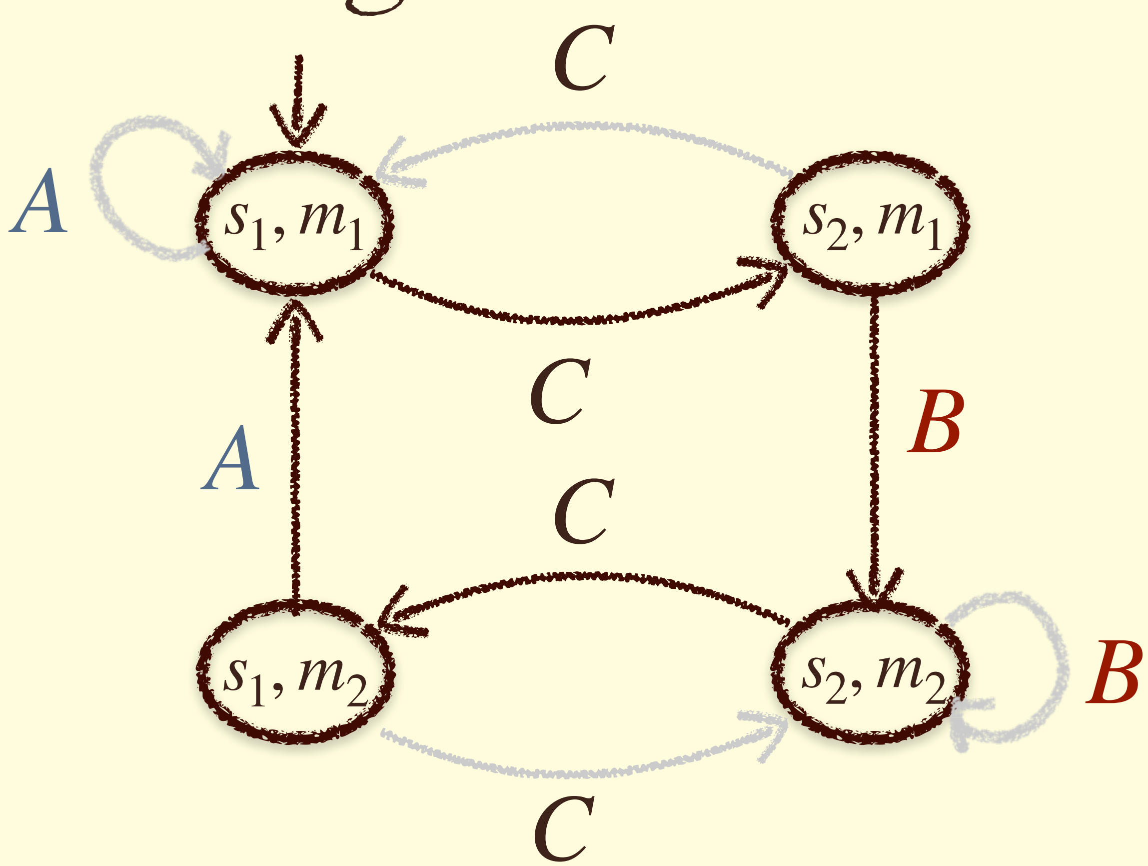


Game arena $\mathscr{A}$:



$$(s_1, m_1) \mapsto (s_1, s_2)$$
$$(s_1, m_2) \mapsto (s_1, s_1)$$
$$(s_2, m_1) \mapsto (s_2, s_2)$$
$$(s_2, m_2) \mapsto (s_2, s_1)$$

Product game $\mathscr{A} \ltimes \mathscr{M}$:



- One can however not apply the [GZ05] result to product games!

Memory-dependent monotony and selectivity

Memory-dependent monotony and selectivity

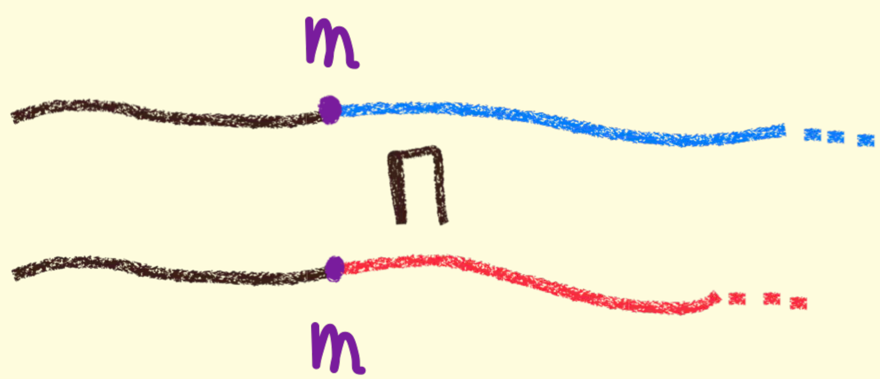
Let $\sqsubseteq$ be a preference relation and $\mathcal{M}$ a memory skeleton.

Memory-dependent monotony and selectivity

Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever

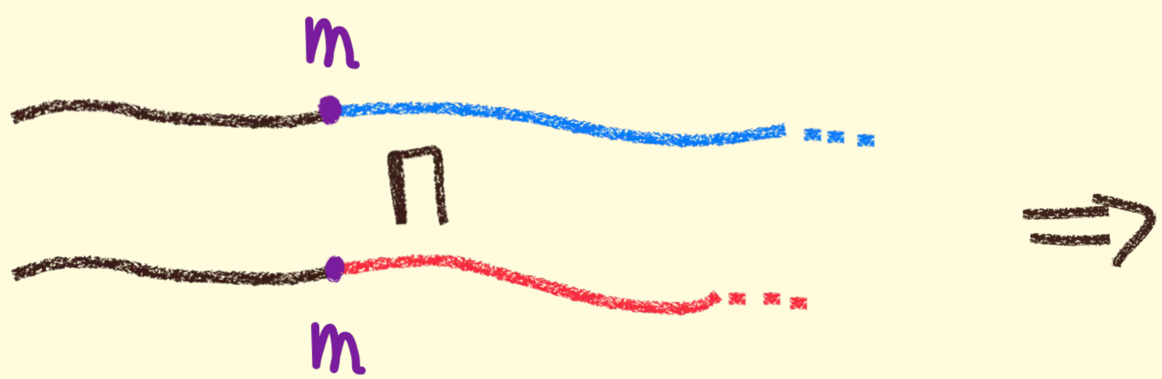


Memory-dependent monotony and selectivity

Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever

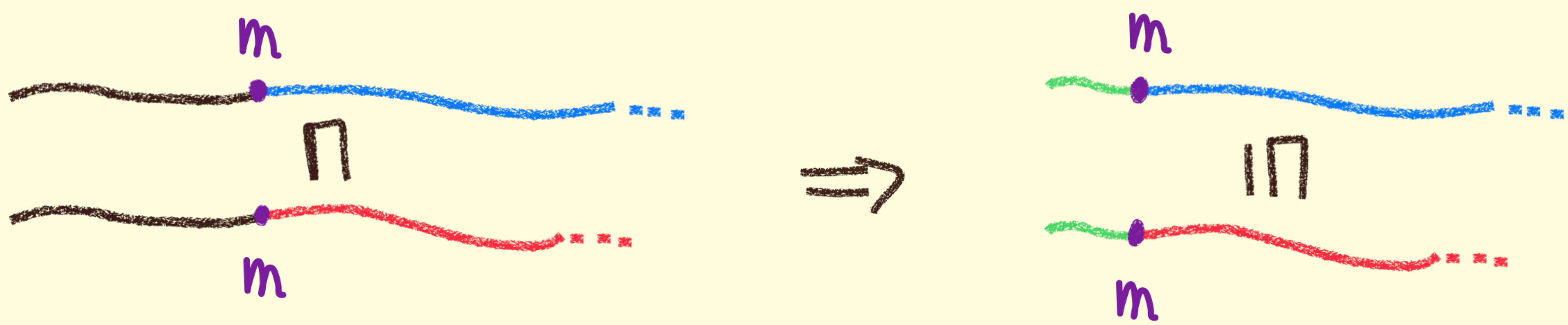


Memory-dependent monotony and selectivity

Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever

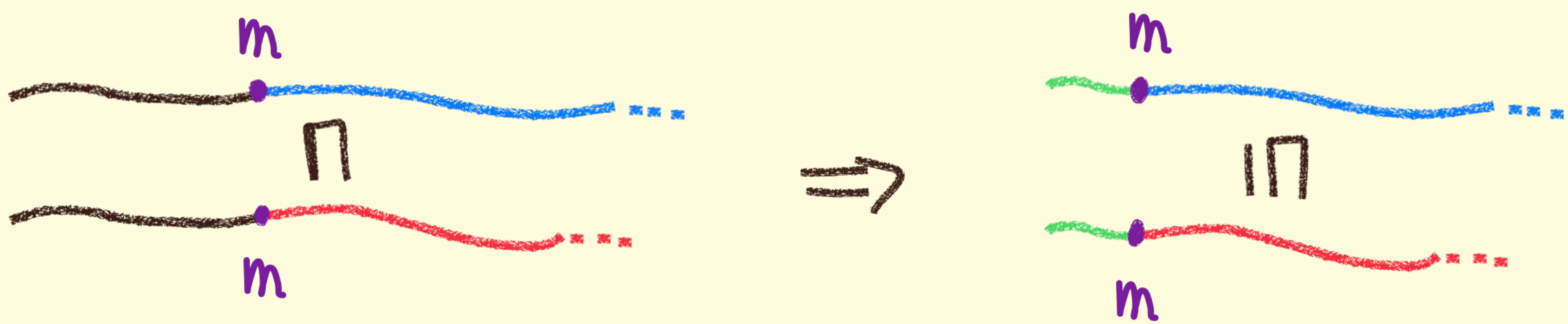


Memory-dependent monotony and selectivity

Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever



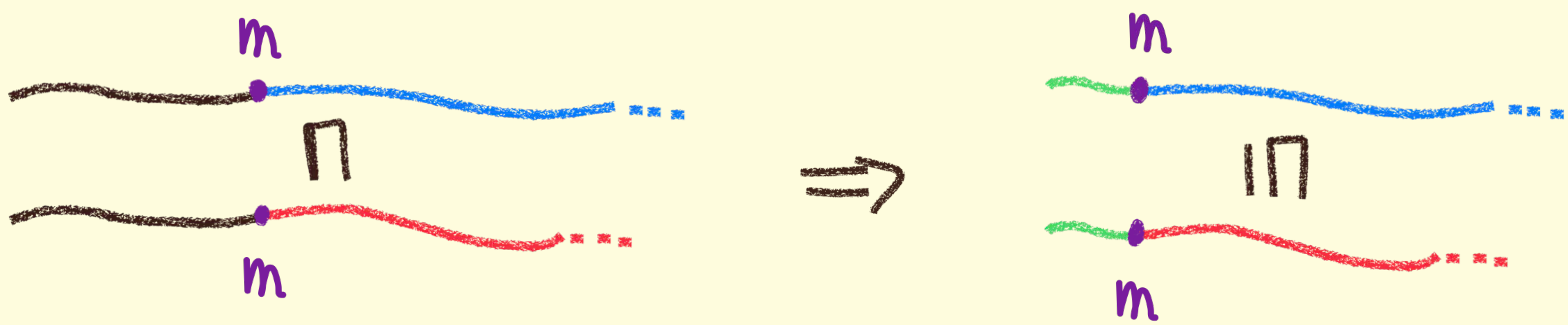
- $\mathscr{M}$-selective whenever

Memory-dependent monotony and selectivity

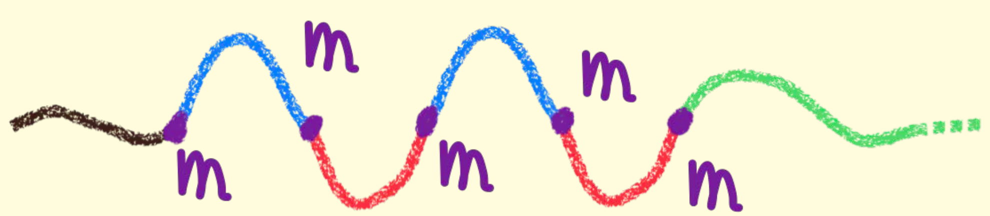
Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever



- $\mathscr{M}$-selective whenever

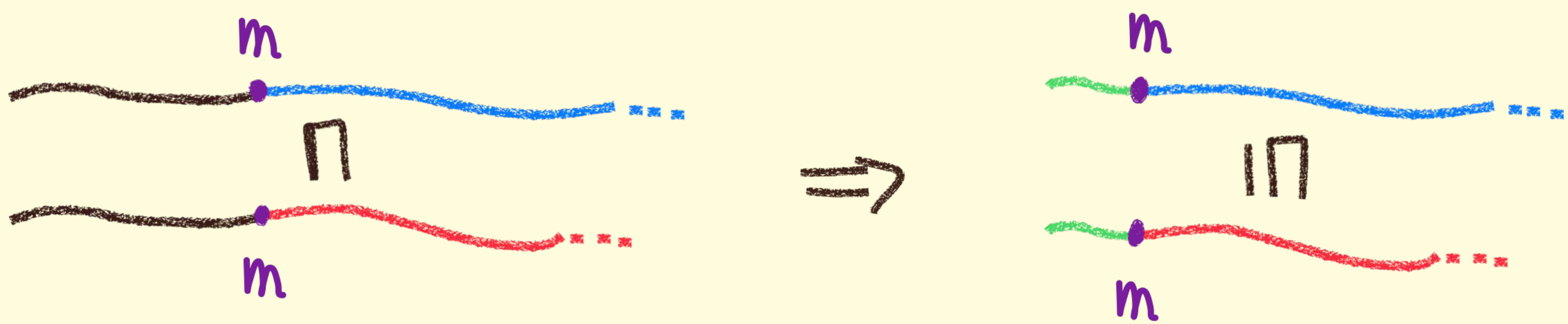


Memory-dependent monotony and selectivity

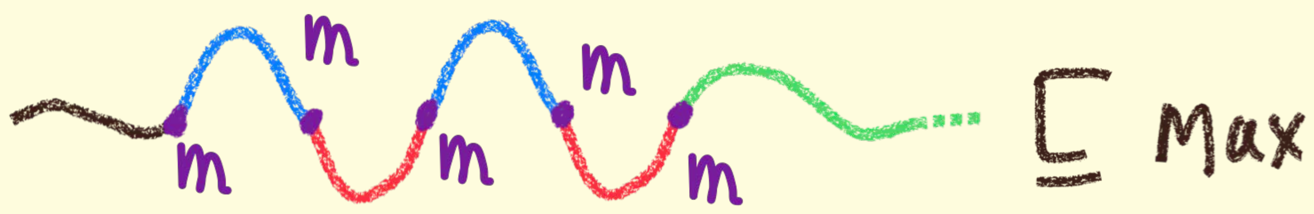
Let $\sqsubseteq$ be a preference relation and $\mathcal{M}$ a memory skeleton.

It is said :

- $\mathcal{M}$-monotone whenever



- $\mathcal{M}$-selective whenever

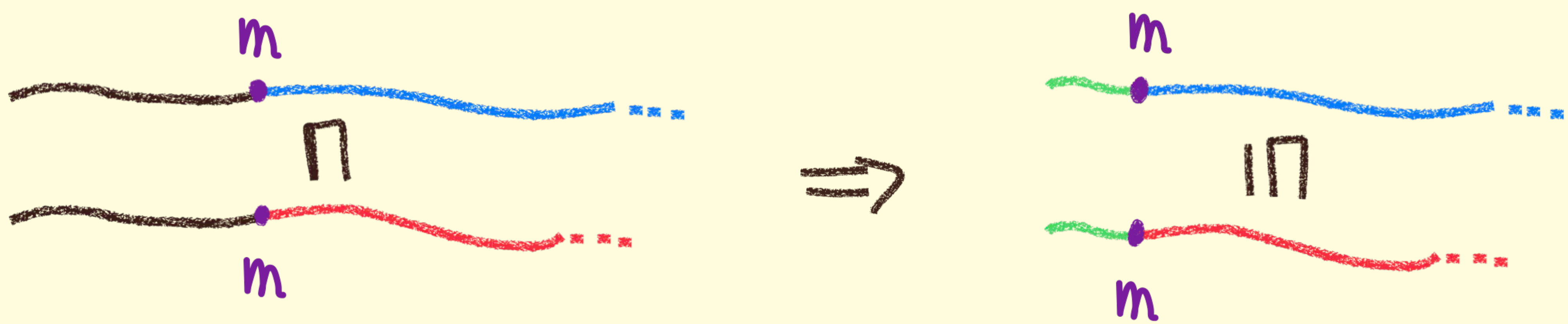


Memory-dependent monotony and selectivity

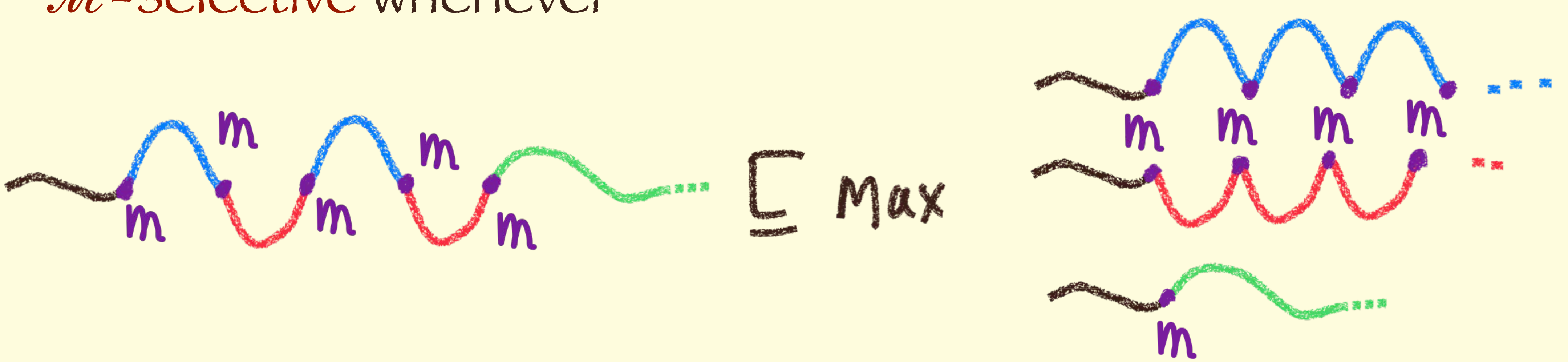
Let $\sqsubseteq$ be a preference relation and $\mathcal{M}$ a memory skeleton.

It is said :

- $\mathcal{M}$-monotone whenever



- $\mathcal{M}$-selective whenever

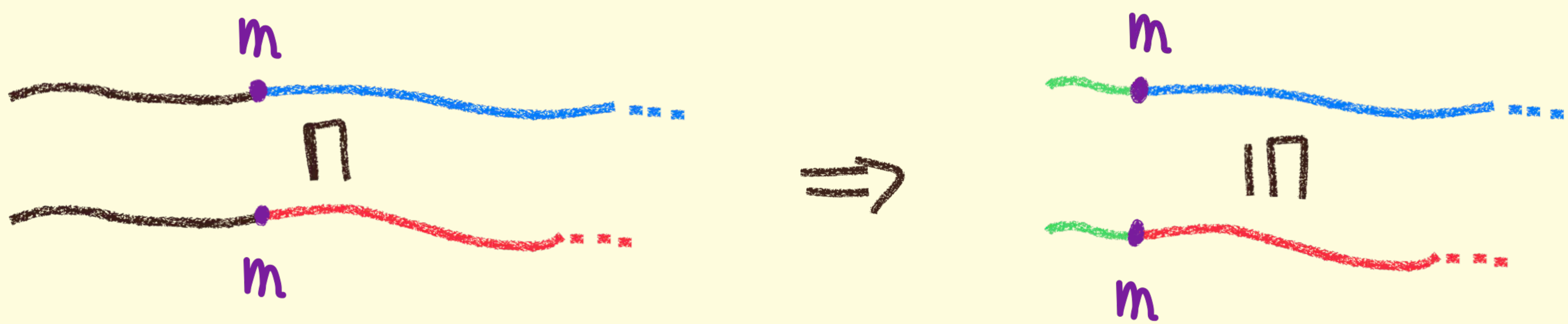


Memory-dependent monotony and selectivity

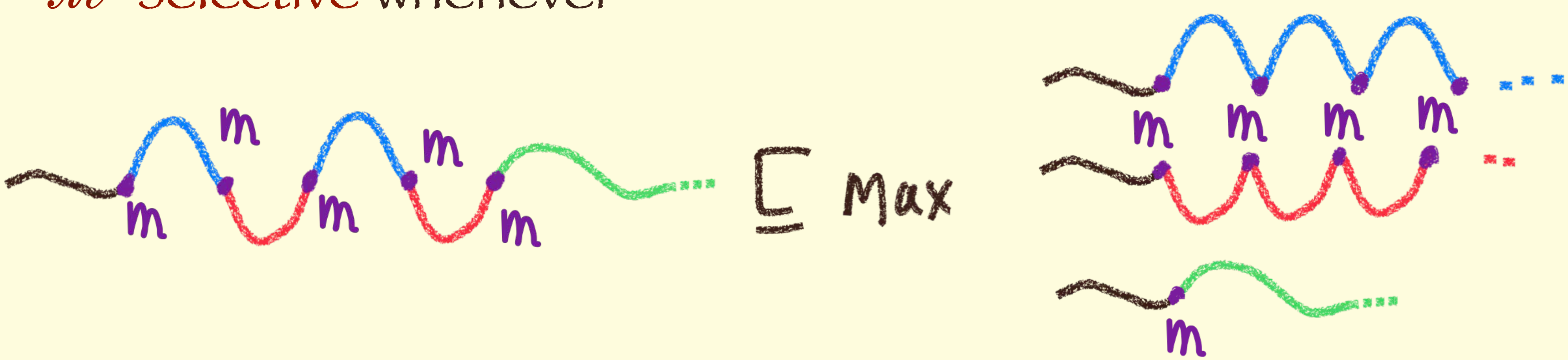
Let $\sqsubseteq$ be a preference relation and $\mathscr{M}$ a memory skeleton.

It is said :

- $\mathscr{M}$-monotone whenever



- $\mathscr{M}$-selective whenever



We look at how $\mathscr{M}$ classifies prefixes and cycles

Formal definitions of $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity

Definition ($\mathcal{M}$-monotony)

Let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory skeleton. A preference relation $\sqsubseteq$ is $\mathcal{M}$-*monotone* if for all $m \in M$, for all $K_1, K_2 \in \mathcal{R}(C)$,

$$\exists\, w \in L_{m_{\mathsf{init}},m},\ [wK_1] \sqsubset [wK_2] \implies \forall\, w' \in L_{m_{\mathsf{init}},m},\ [w'K_1] \sqsubseteq [w'K_2].$$

Definition ($\mathcal{M}$-selectivity)

Let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory skeleton. A preference relation $\sqsubseteq$ is $\mathcal{M}$-*selective* if for all $w \in C^*$, $m = \widehat{\alpha_{\mathsf{upd}}}(m_{\mathsf{init}}, w)$, for all $K_1, K_2 \in \mathcal{R}(C)$ such that $K_1, K_2 \subseteq L_{m,m}$, for all $K_3 \in \mathcal{R}(C)$,

$$[w(K_1 \cup K_2)^* K_3] \sqsubseteq [wK_1^*] \cup [wK_2^*] \cup [wK_3].$$

Our characterization for $\mathscr{M}$-determinacy

Our characterization for $\mathscr{M}$-determinacy

Characterization - Two-player games

Our characterization for $\mathcal{M}$-determinacy

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have optimal $\mathcal{M}$-strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective

Our characterization for $\mathcal{M}$-determinacy

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have optimal $\mathcal{M}$-strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective

Characterization - One-player games

Our characterization for $\mathcal{M}$-determinacy

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have optimal $\mathcal{M}$-strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective

Characterization - One-player games

The two following assertions are equivalent :

1. All finite P_1-games have (uniform) optimal $\mathcal{M}$-strategies
2. $\sqsubseteq$ is $\mathcal{M}$-monotone and $\mathcal{M}$-selective

Our characterization for $\mathcal{M}$-determinacy

Characterization - Two-player games

The two following assertions are equivalent :

1. All finite games have optimal $\mathcal{M}$-strategies for both players
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective

Characterization - One-player games

The two following assertions are equivalent :

1. All finite P_1-games have (uniform) optimal $\mathcal{M}$-strategies
2. $\sqsubseteq$ is $\mathcal{M}$-monotone and $\mathcal{M}$-selective

➡ We recover [GZ05] with $\mathcal{M} = \mathcal{M}_{\text{triv}}$

Applications

Applications

Transfer/Lifting theorem

- If in all finite one-player game for player P_i, P_i has optimal $\mathscr{M}_i$-strategies, then both players have optimal $\mathscr{M}_1 \times \mathscr{M}_2$-strategies in all finite two-player games.

Applications

Transfer/Lifting theorem

- If in all finite one-player game for player P_i, P_i has optimal $\mathcal{M}_i$-strategies, then both players have optimal $\mathcal{M}_1 \times \mathcal{M}_2$-strategies in all finite two-player games.

Very powerful and extremely useful in practice!

Applications

Transfer/Lifting theorem

- If in all finite one-player game for player P_i, P_i has optimal $\mathscr{M}_i$-strategies, then both players have optimal $\mathscr{M}_1 \times \mathscr{M}_2$-strategies in all finite two-player games.

Very powerful and extremely useful in practice!

Subclasses of games

- If both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathscr{M}$-monotone and $\mathscr{M}$-selective, then both players have optimal memoryless strategies in all $\mathscr{M}$-covered games.

Memory-covered arenas

Memory-covered arenas

If the game has enough information from $\mathcal{M}$, then memoryless strategies will be sufficient

Memory-covered arenas

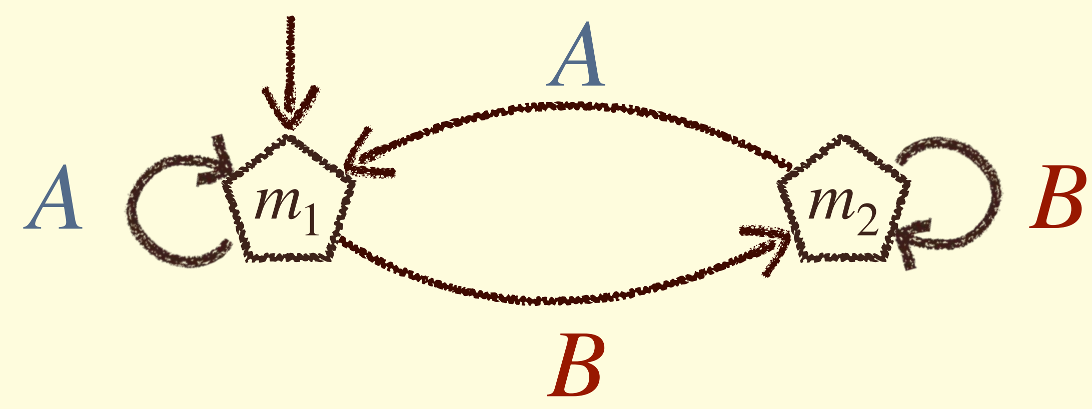
If the game has enough information from $\mathcal{M}$,
then memoryless strategies will be sufficient

Covered arenas ≈ same properties as product arenas

Memory-covered arenas

If the game has enough information from $\mathcal{M}$, then memoryless strategies will be sufficient

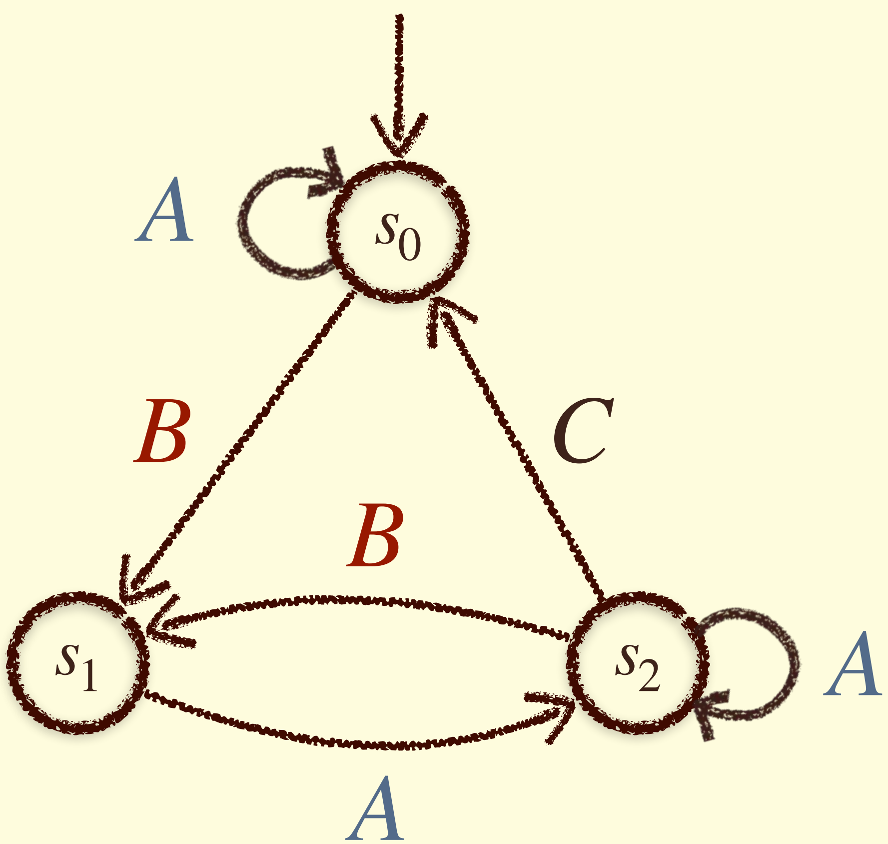
Covered arenas ≈ same properties as product arenas

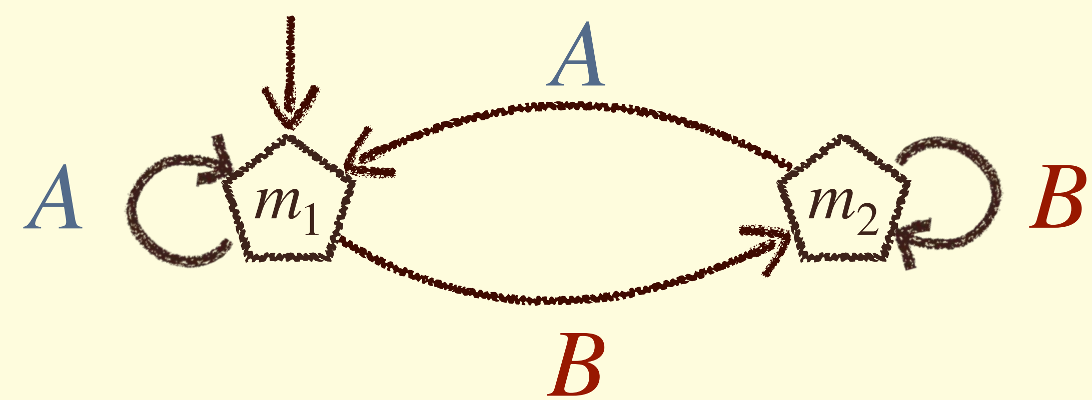


Memory-covered arenas

If the game has enough information from $\mathscr{M}$, then memoryless strategies will be sufficient

Covered arenas ≈ same properties as product arenas



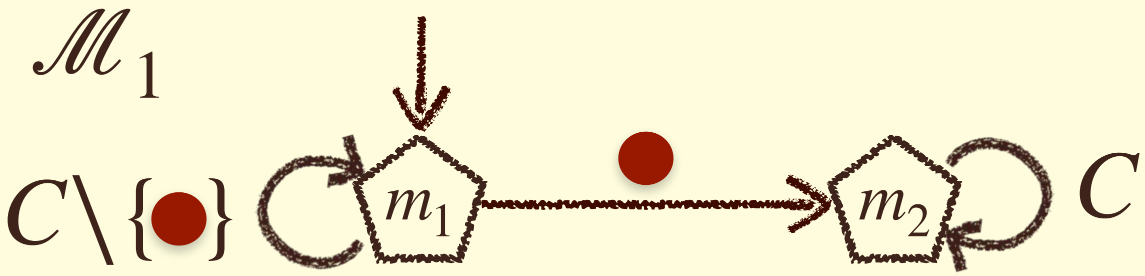


Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach(●) $\wedge$ Reach(●)

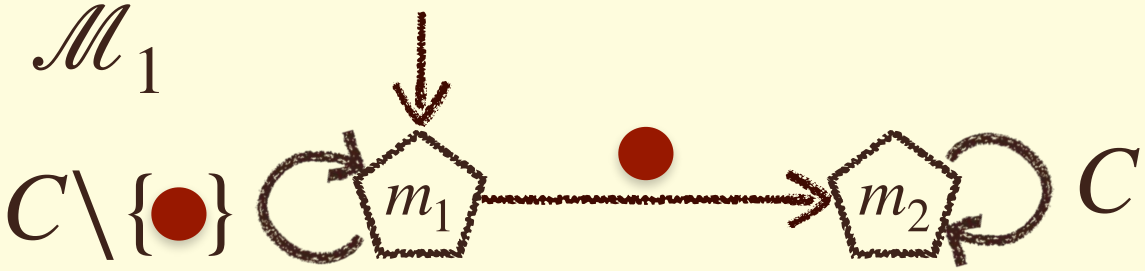
Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach(●) $\wedge$ Reach(●)



Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach(●) $\wedge$ Reach(●)

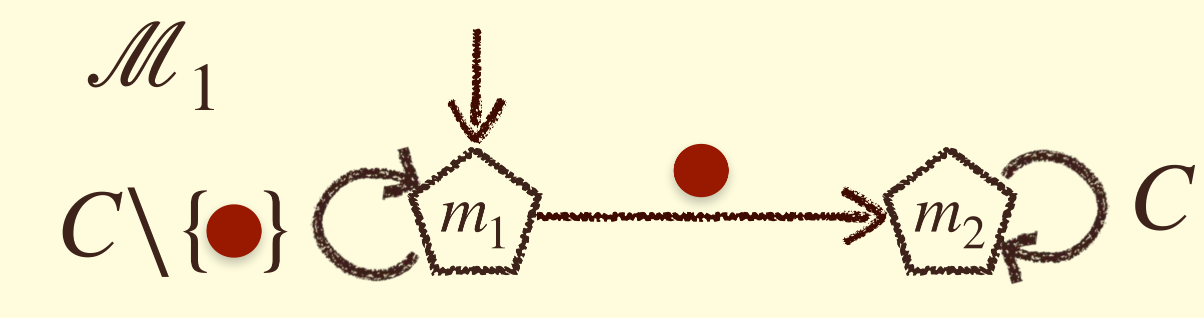


$\sqsubseteq$ is $\mathscr{M}_1$-monotone,

but not $\mathscr{M}_1$-selective

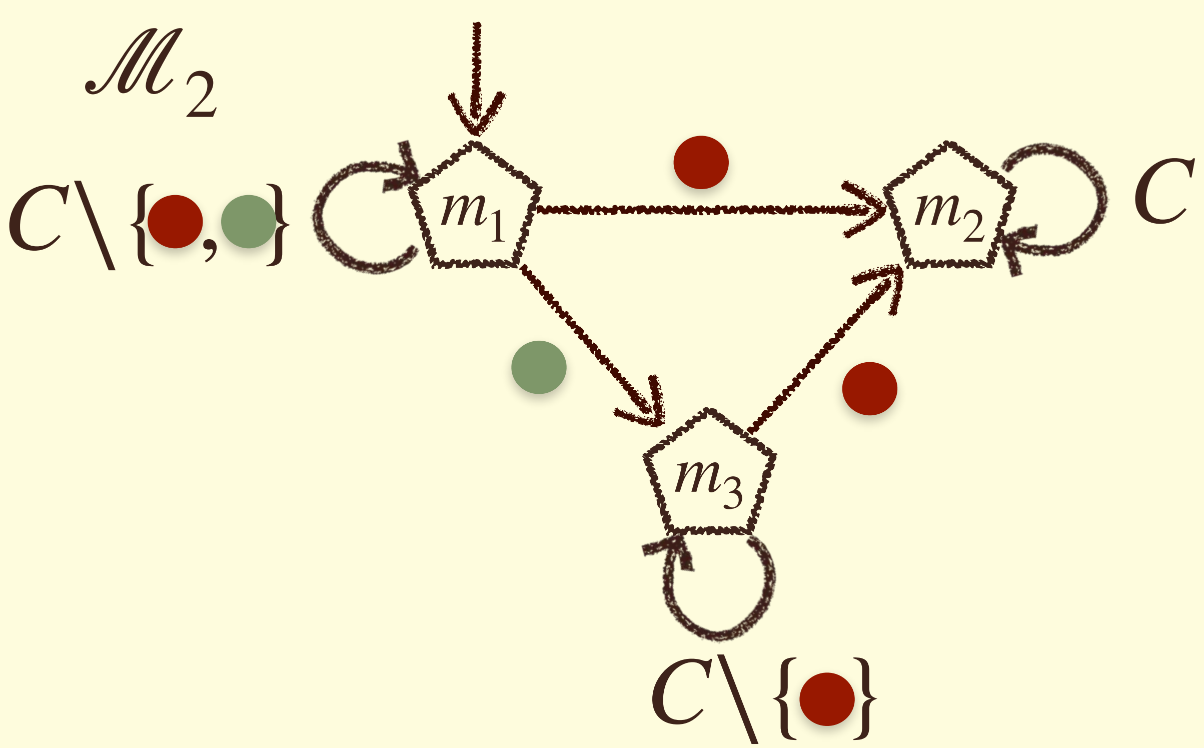
Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach(●) $\wedge$ Reach(●)



$\sqsubseteq$ is $\mathscr{M}_1$-monotone,

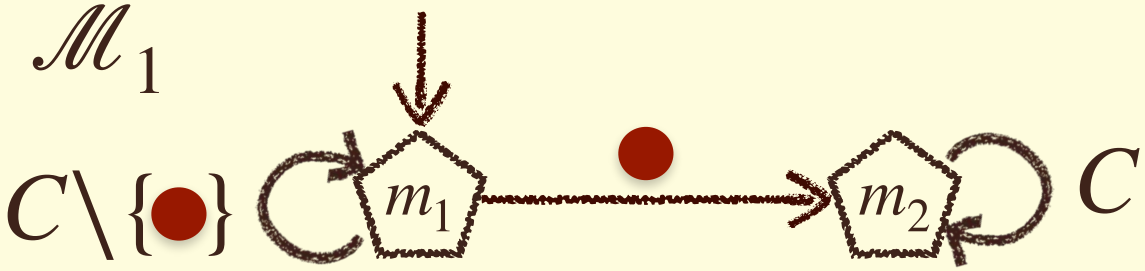
but not $\mathscr{M}_1$-selective



Example of application

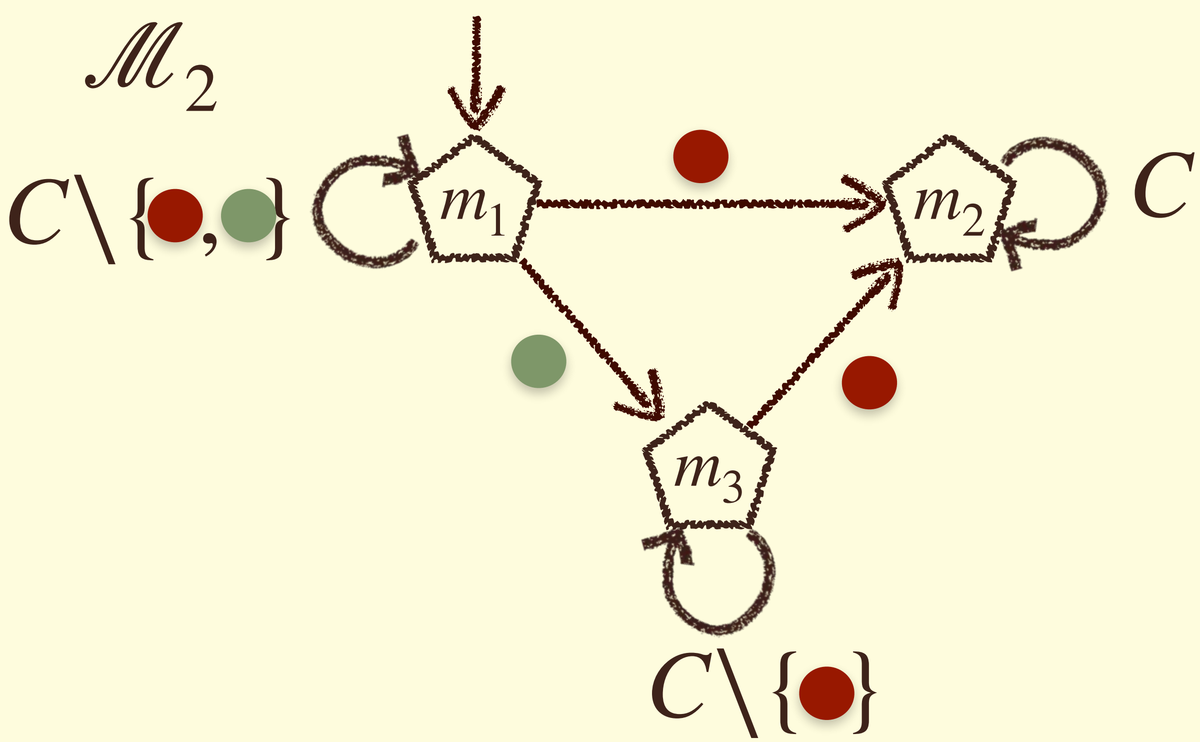
$\sqsubseteq$ defined by a conjunction of reachability Reach(●) $\wedge$ Reach(●)

$\mathscr{M}_1$



$\sqsubseteq$ is $\mathscr{M}_1$-monotone,
but not $\mathscr{M}_1$-selective

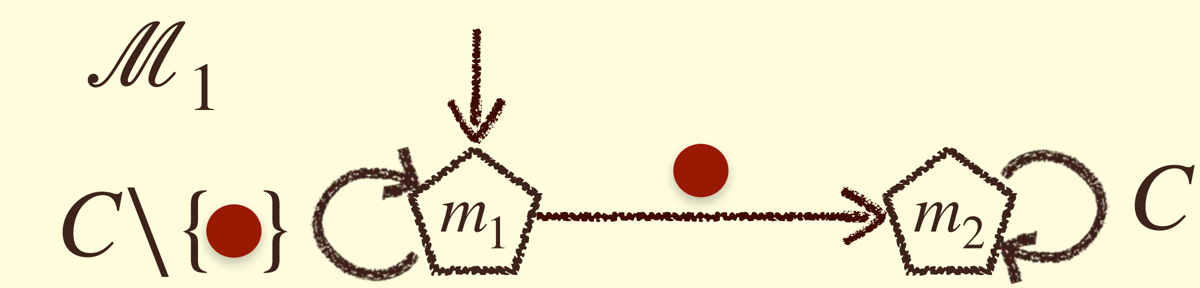
$\mathscr{M}_2$



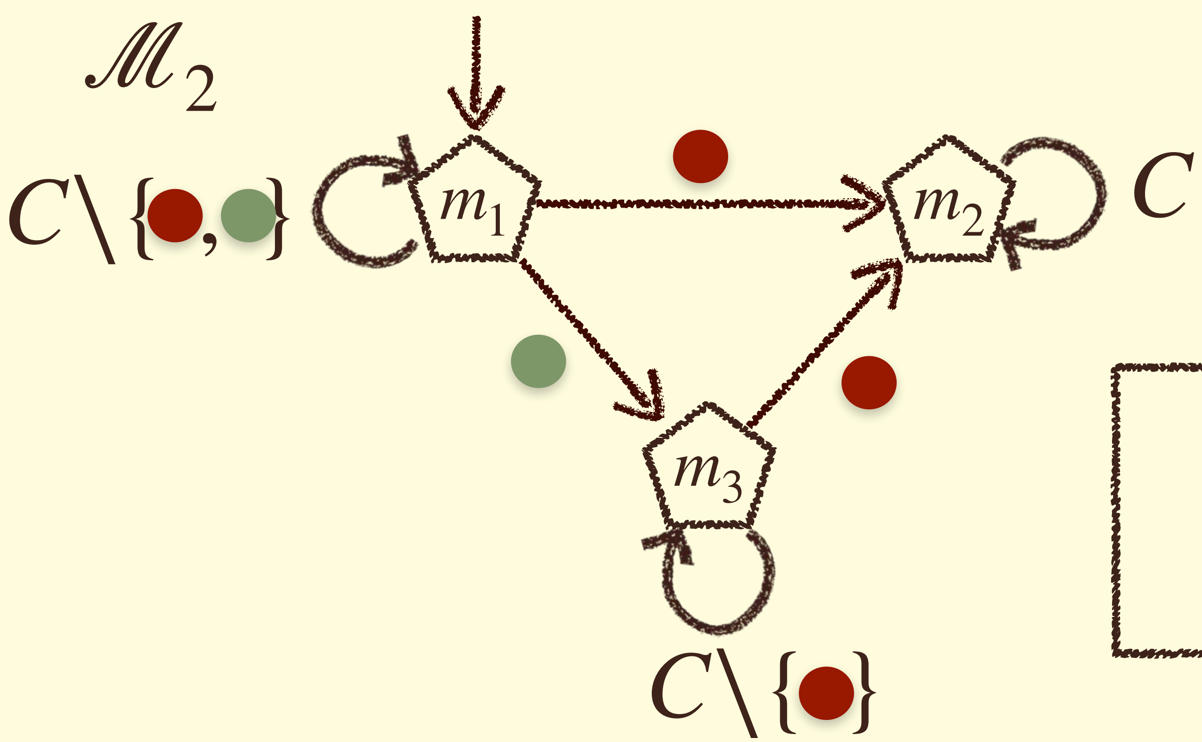
$\sqsubseteq$ is $\mathscr{M}_2$-selective

Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach($\bullet$) $\wedge$ Reach($\bullet$)



$\sqsubseteq$ is $\mathcal{M}_1$-monotone,
but not $\mathcal{M}_1$-selective

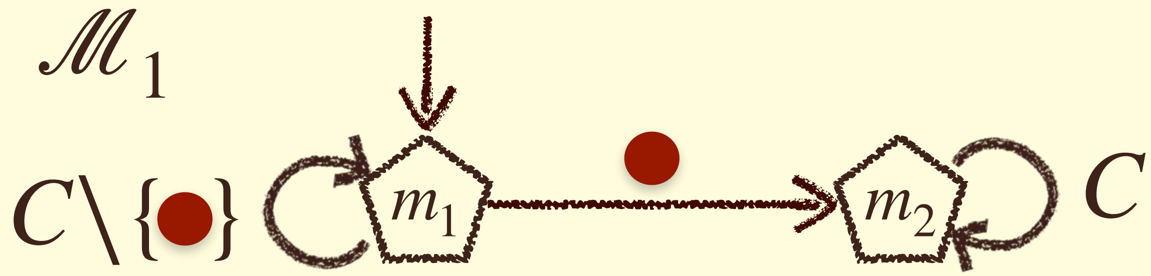


$\sqsubseteq$ is $\mathcal{M}_2$-selective

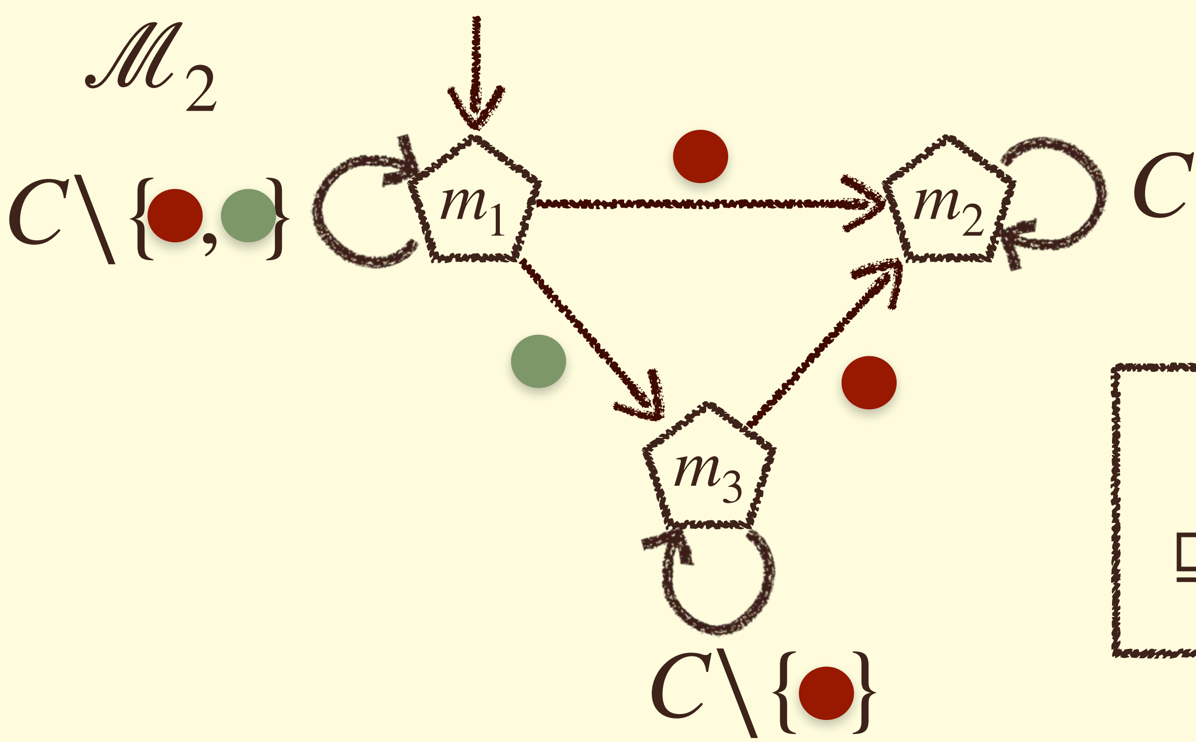
$\boxed{\sqsubseteq \text{ is } \mathcal{M}_1\text{-monotone and } \mathcal{M}_2\text{-selective}}$

Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach($\bullet$) $\wedge$ Reach($\bullet$)



$\sqsubseteq$ is $\mathcal{M}_1$-monotone,
but not $\mathcal{M}_1$-selective

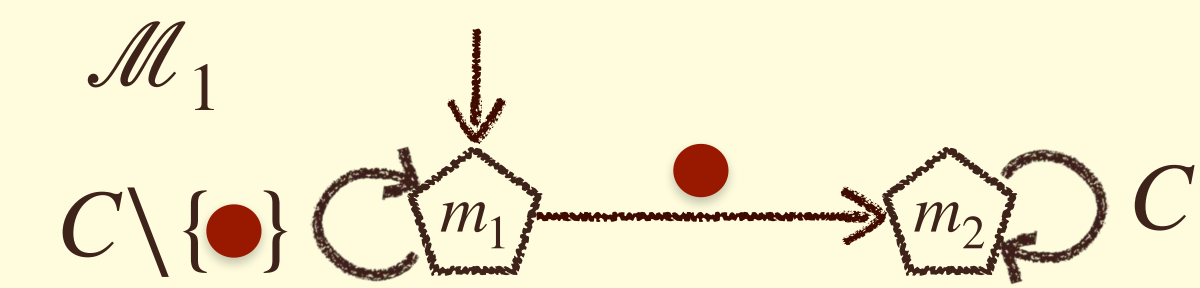


$\sqsubseteq$ is $\mathcal{M}_2$-selective

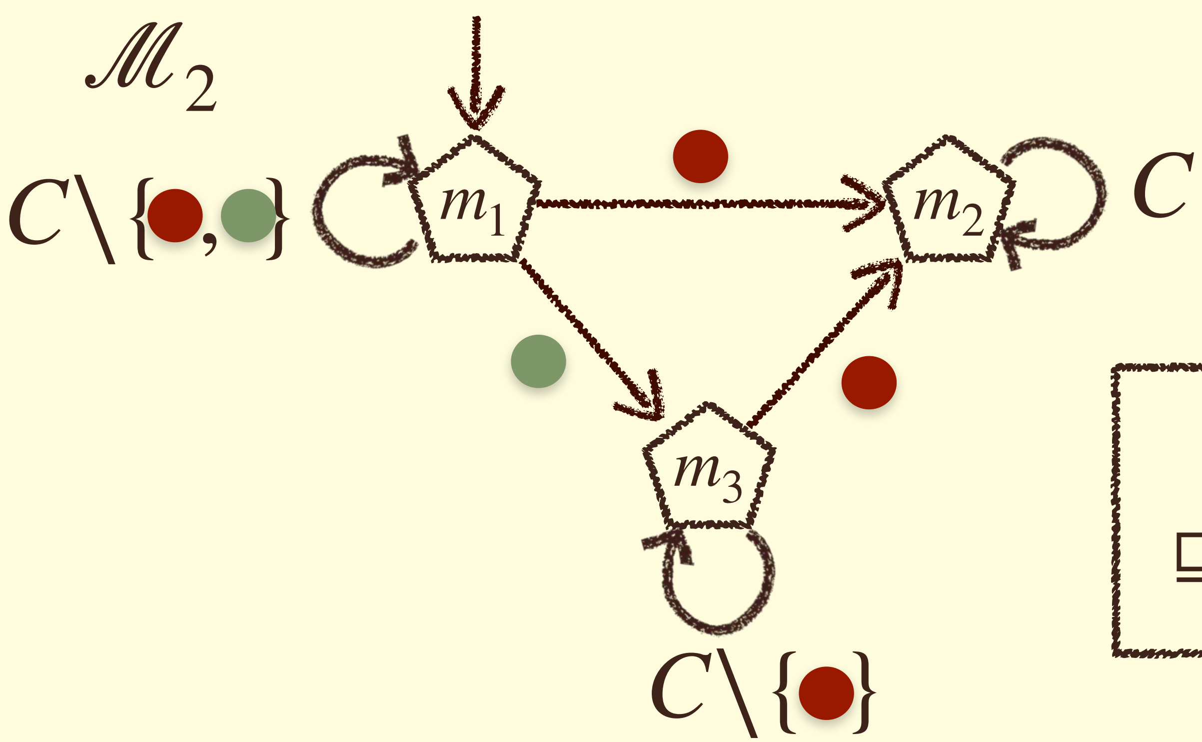
$\sqsubseteq$ is $\mathcal{M}_1$-monotone and $\mathcal{M}_2$-selective
$\sqsubseteq^{-1}$ is $\mathcal{M}_1$-monotone and $\mathcal{M}_{\text{triv}}$-selective

Example of application

$\sqsubseteq$ defined by a conjunction of reachability Reach(●) ∧ Reach(●)



$\sqsubseteq$ is $\mathcal{M}_1$-monotone,
but not $\mathcal{M}_1$-selective



$\sqsubseteq$ is $\mathcal{M}_2$-selective

$\sqsubseteq$ is $\mathcal{M}_1$-monotone and $\mathcal{M}_2$-selective
$\sqsubseteq^{-1}$ is $\mathcal{M}_1$-monotone and $\mathcal{M}_{\text{triv}}$-selective

➡ Memory $\mathcal{M}_2$ is sufficient for both players!!

Conclusion

A generalization of [GZ05]

- To arena-independent finite memory
- Applies to generalized reachability or parity, lower- and upper-bounded (multi-dimension) energy games

Conclusion

A generalization of [GZ05]

- To arena-independent finite memory
- Applies to generalized reachability or parity, lower- and upper-bounded (multi-dimension) energy games

Limitations

- Does only capture arena-independent finite memory
- Hard to generalize (remember counter-example)
- Does not apply to multi-dim. MP, MP+parity, energy+MP (infinite memory)

Conclusion

Conclusion

Other approaches

- Sufficient conditions giving half-memory management results
- Compositionality w.r.t. objectives [LPR18]

Conclusion

Other approaches

- Sufficient conditions giving half-memory management results
- Compositionality w.r.t. objectives [LPR18]

Further work

- Understand the arena-dependent framework
- Infinite arenas
- Probabilistic setting
- Other concepts (Nash equilibria)