Average-Price-per-Reward Games on Hybrid Automata with Strong Resets

Marcin Jurdziński, Ranko Lazić and **Michał Rutkowski**

Department of Computer Science
University of Warwick

GASICS Meeting, Brussels, Belgium, (5 – 6 March 2009)

INTRODUCTION

Hybrid systems:

- discrete and continuous behavior
- digital control of an analogue plant
- modeled by **hybrid automata**

Average Price-per-Reward

- Embedded Systems
- Reactive Behaviour

Games:

- controller synthesis

Hybrid systems:

- discrete and continuous behavior
- digital control of an analogue plant
- modeled by **hybrid automata**

Average Price-per-Reward

- Embedded Systems
- Reactive Behaviour

Games:

- controller synthesis

Hybrid systems:

- discrete and continuous behavior
- digital control of an analogue plant
- modeled by **hybrid automata**

Average Price-per-Reward

- Embedded Systems
- Reactive Behaviour

Games:

- controller synthesis

FURTHER INTRODUCTION

CONTEXT

- Hybrid automata with strong resets [Lygeros et. al '00, Brihaye et. al '04]
- Games on hybrid automata with strong resets [Bouyer et. al '06, Bouyer et. al '08]
- Average Price-per-Reward [Bouyer et. al '04, Jurdzinski and Trivedi '08]

GOAL

We are interested in computing game values and optimal strategies for average price-per-reward games.

SOLUTION

- Optimality equations [Jurdzinski and Trivedi '07, Bouyer et. al '08]
- Strategy improvement [Jurdzinski and Trivedi '07]
- Equivalence [Bouyer et. al '08]
- Graphs with edge games [NEW]

Further introduction

Context

- Hybrid automata with strong resets [Lygeros et. al '00, Brihaye et. al '04]
- Games on hybrid automata with strong resets [Bouyer et. al '06, Bouyer et. al '08]
- Average Price-per-Reward [Bouyer et. al '04, Jurdzinski and Trivedi '08]

Goal

We are interested in computing game values and optimal strategies for average price-per-reward games.

Solution

- Optimality equations [Jurdzinski and Trivedi '07, Bouyer et. al '08]
- Strategy improvement [Jurdzinski and Trivedi '07]
- Equivalence [Bouyer et. al '08]
- Graphs with edge games [NEW]

Further introduction

Context

- Hybrid automata with strong resets [Lygeros et. al '00, Brihaye et. al '04]
- Games on hybrid automata with strong resets [Bouyer et. al '06, Bouyer et. al '08]
- Average Price-per-Reward [Bouyer et. al '04, Jurdzinski and Trivedi '08]

Goal

We are interested in computing game values and optimal strategies for average price-per-reward games.

Solution

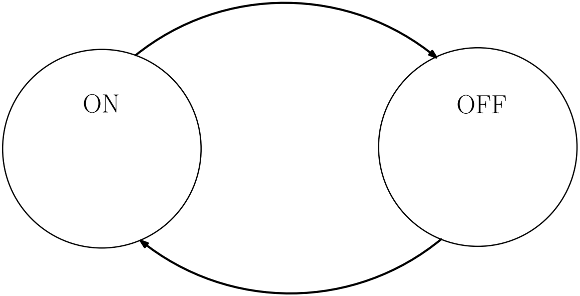
- Optimality equations [Jurdzinski and Trivedi '07, Bouyer et. al '08]
- Strategy improvement [Jurdzinski and Trivedi '07]
- Equivalence [Bouyer et. al '08]
- Graphs with edge games [NEW]

1. Hybrid Automata

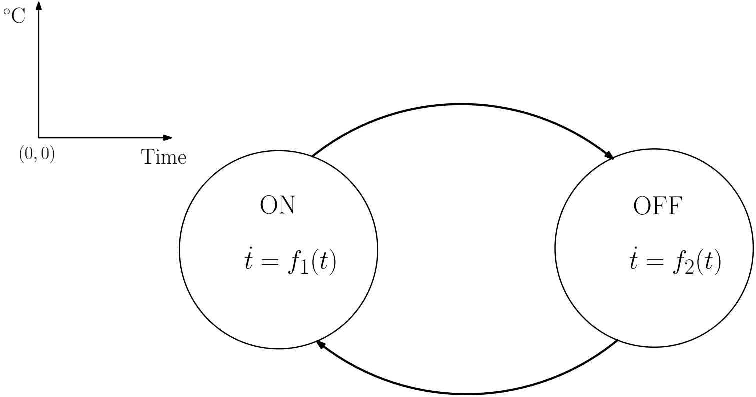
2. Average Price-per-Reward Games

3. Solving Hybrid Games

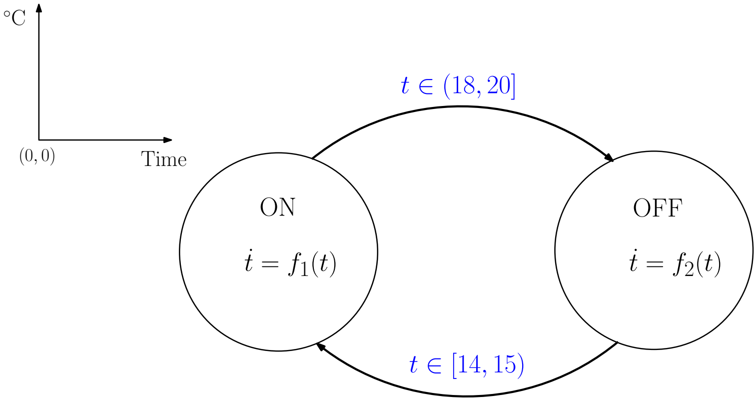
HYBRID AUTOMATON: THERMOSTAT



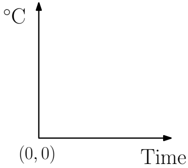
Hybrid Automaton: Thermostat

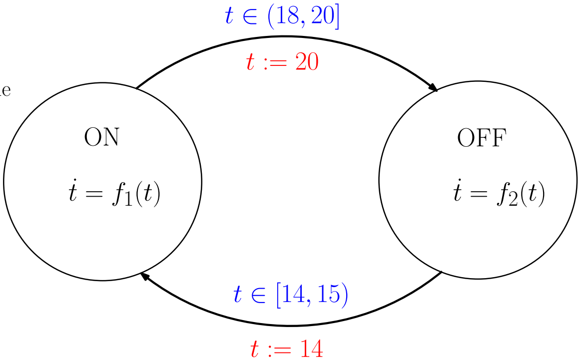


Hybrid Automaton: Thermostat

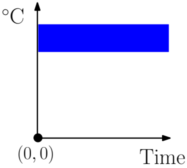


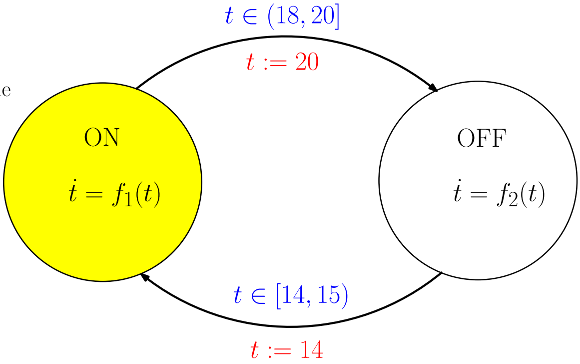
HYBRID AUTOMATON: THERMOSTAT



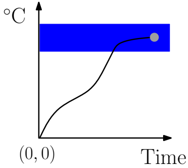


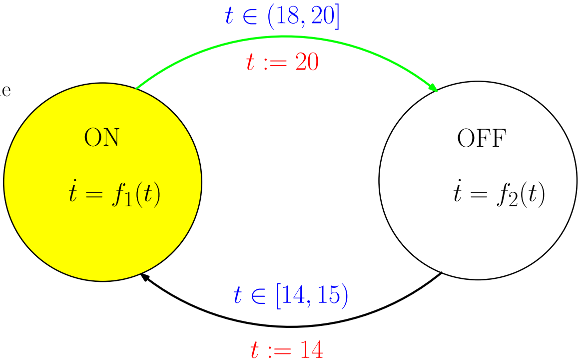
Behaviour



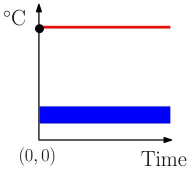


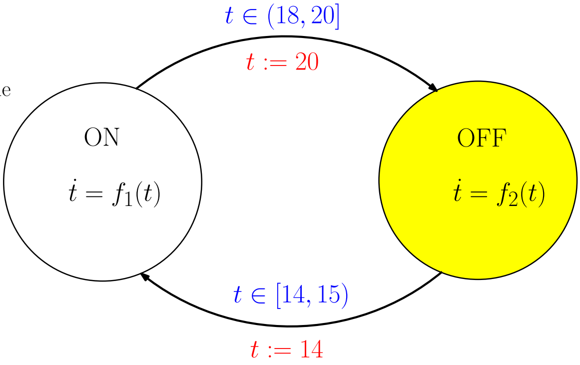
Behaviour



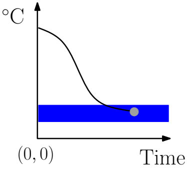


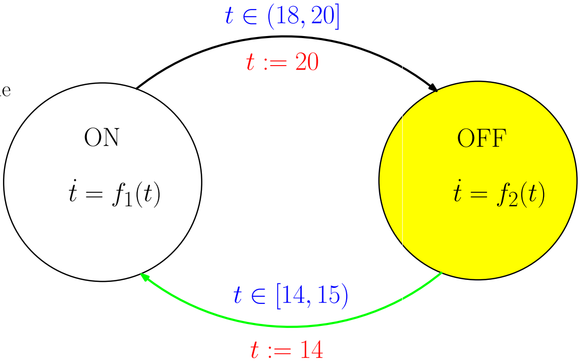
BEHAVIOUR



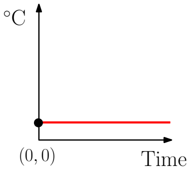


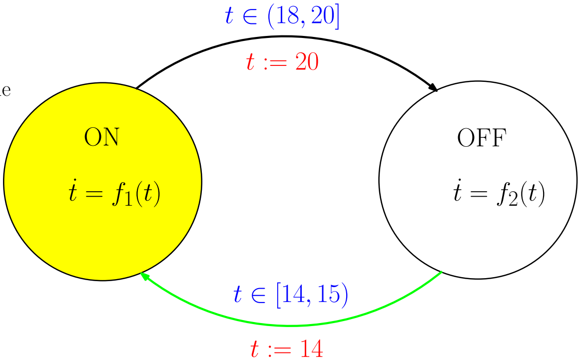
Behaviour



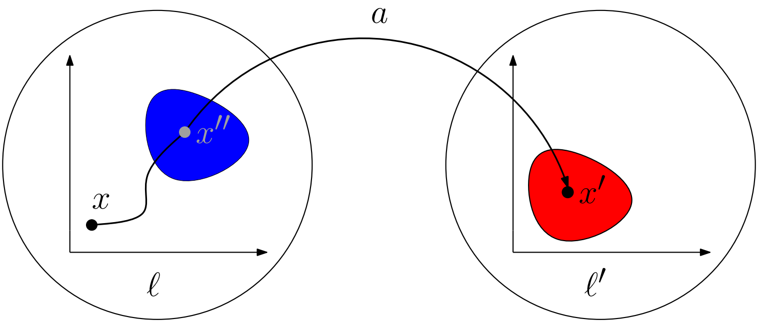


Behaviour





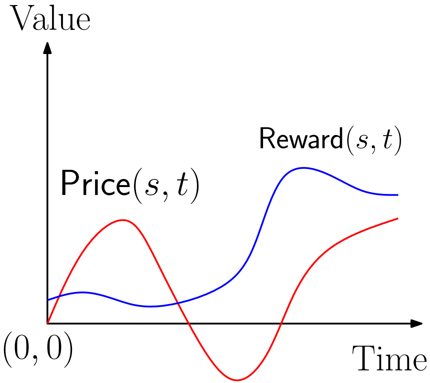
Timed action



$$s \xrightarrow{a}_t s'$$

$$(\ell, x) \xrightarrow{a}_t (\ell', x')$$

Pricing



$$s \xrightarrow{a}_t s'$$

Representation

Definability

- First-order formulae over $\mathscr{R} = \langle \mathbb{R}, \leqslant, +, \cdot, 0, 1 \rangle$ to represent infinite structures.
- Equations/formulae used are first-order expressible in $\mathscr{R}$.

Computability

Using decidability of the first-order theory of $\mathscr{R}$ we get:

- Approximability of definable values
- Computability of almost optimal solutions

REPRESENTATION

DEFINABILITY

- First-order formulae over $\mathscr{R} = \langle \mathbb{R}, \leqslant, +, \cdot, 0, 1 \rangle$ to represent infinite structures.
- Equations/formulae used are first-order expressible in $\mathscr{R}$.

COMPUTABILITY

Using decidability of the first-order theory of $\mathscr{R}$ we get:

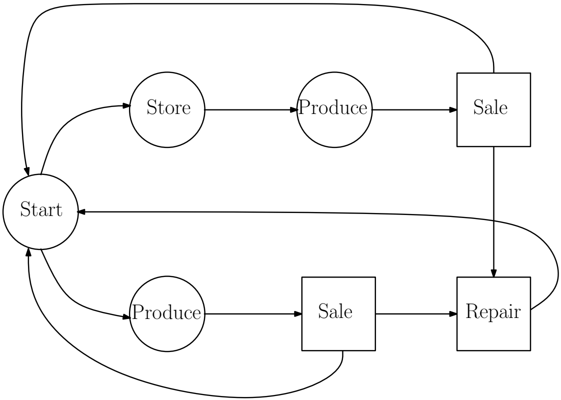
- Approximability of definable values
- Computability of almost optimal solutions

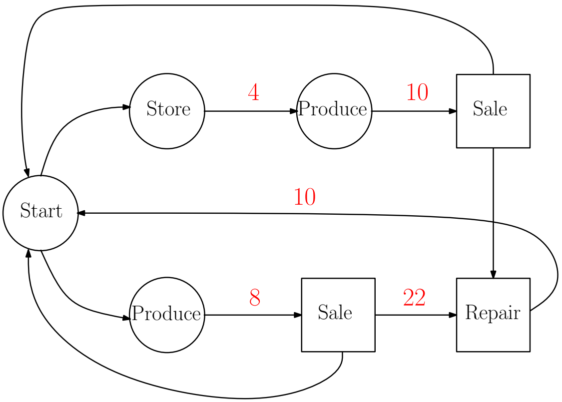
1 Hybrid Automata

2 Average Price-per-Reward Games

3 Solving Hybrid Games

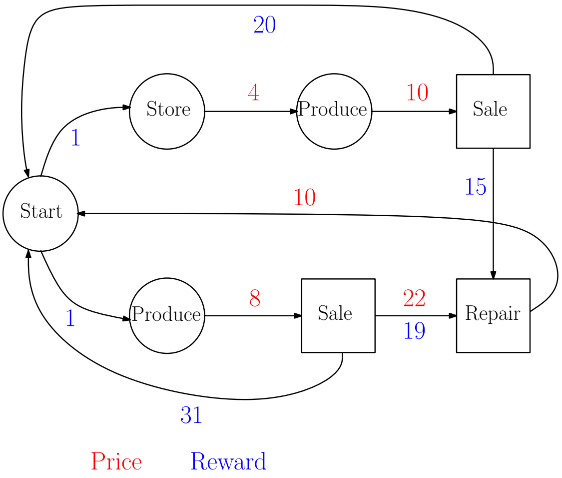
Average Price-per-Reward



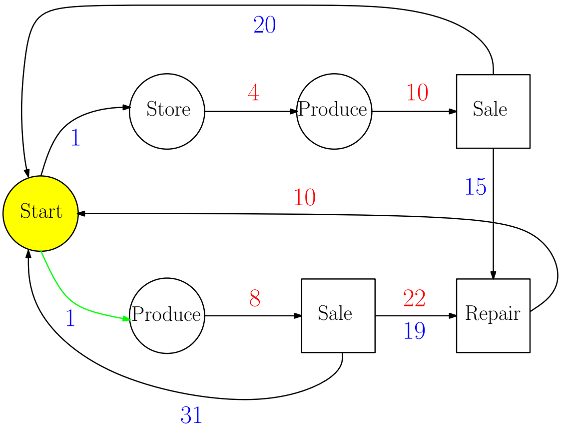


Price

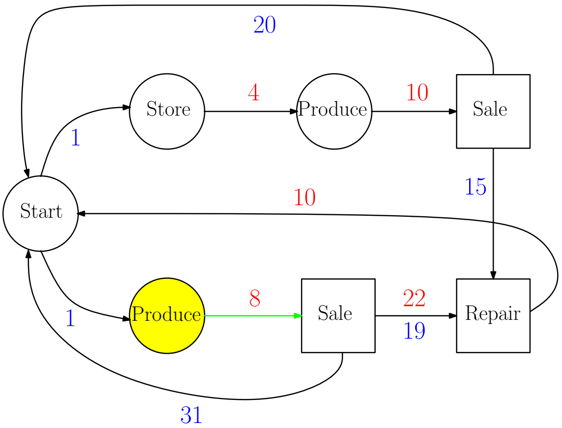
Average Price-per-Reward



PLAYING THE GAME

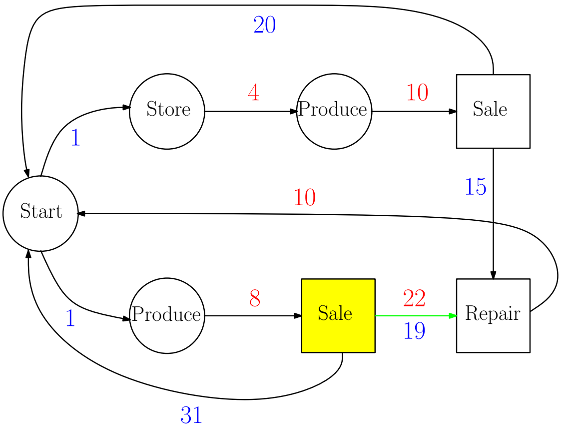


Average Price-per-Reward: 0

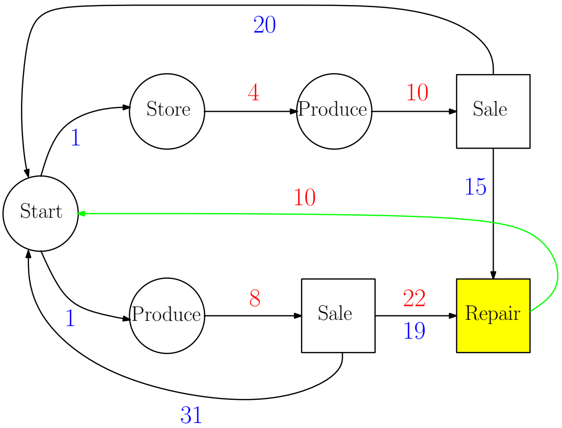


Average Price-per-Reward: $\frac{0}{1} = 0$

Playing the game

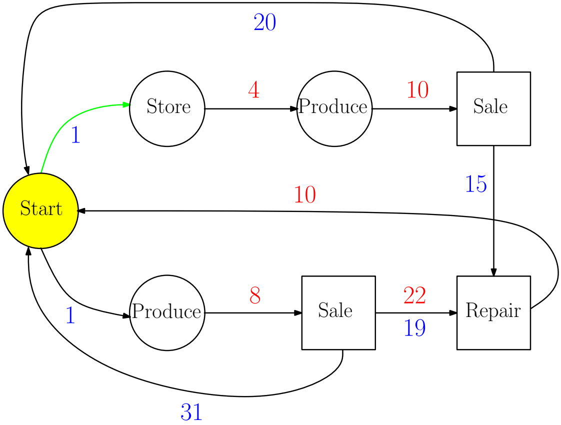


Average Price-per-Reward: $\frac{0+8}{1+0} = 8$



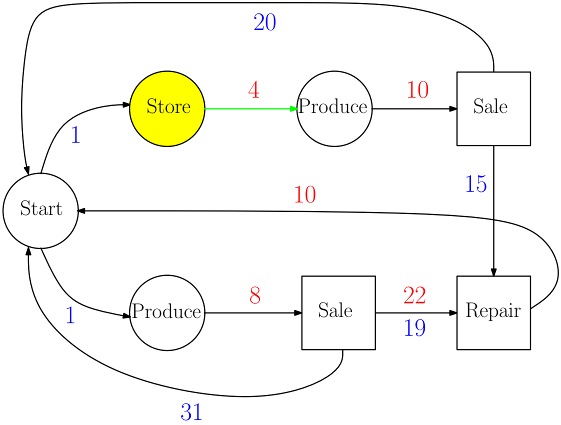
Average Price-per-Reward: $\frac{8+22}{1+19} = \frac{3}{2}$

Playing the game



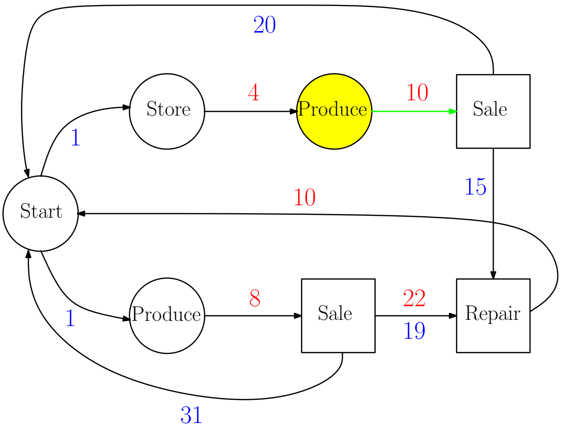
Average Price-per-Reward: $\frac{30+10}{20+0} = 2$

Playing the game



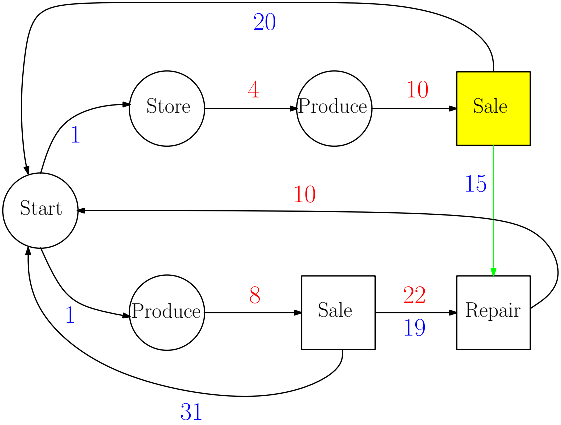
Average Price-per-Reward: $\frac{40+0}{20+1} = \frac{40}{21}$

Playing the game

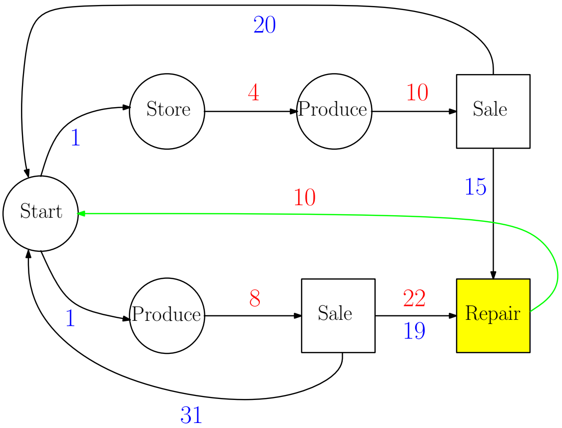


Average Price-per-Reward: $\frac{40+4}{21+0} = \frac{44}{21}$

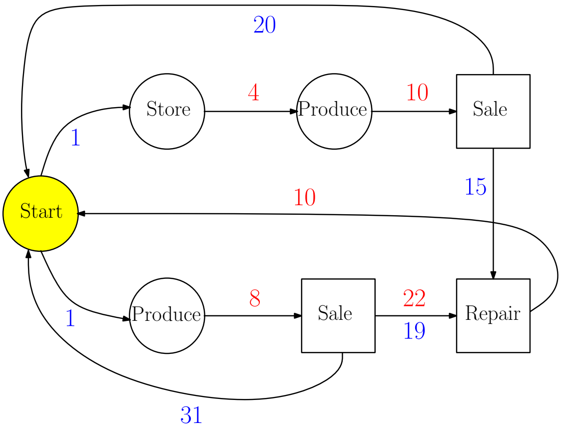
Playing the game



Average Price-per-Reward: $\frac{44+10}{21+0} = \frac{18}{7}$

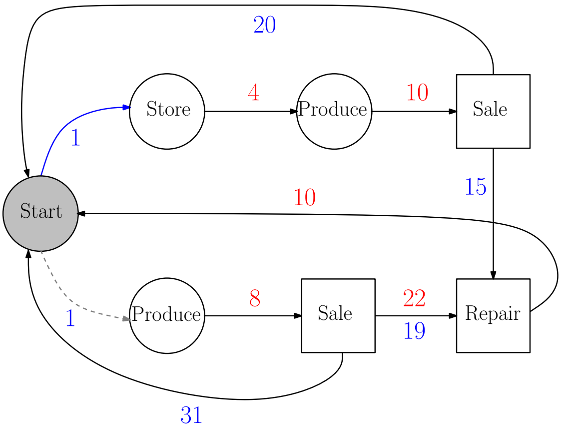


Average Price-per-Reward: $\frac{54+0}{21+15} = \frac{3}{2}$

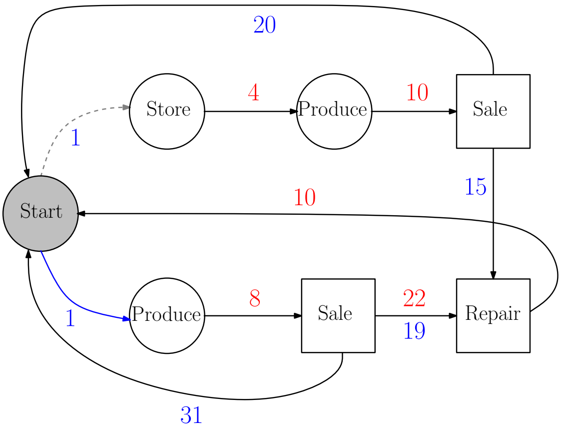


Average Price-per-Reward: $\frac{54+10}{36+0} = \frac{16}{9}$

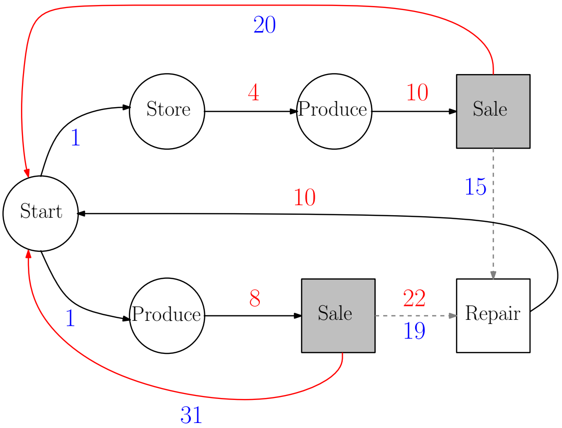
STRATEGIES OF PLAYER MIN: μ



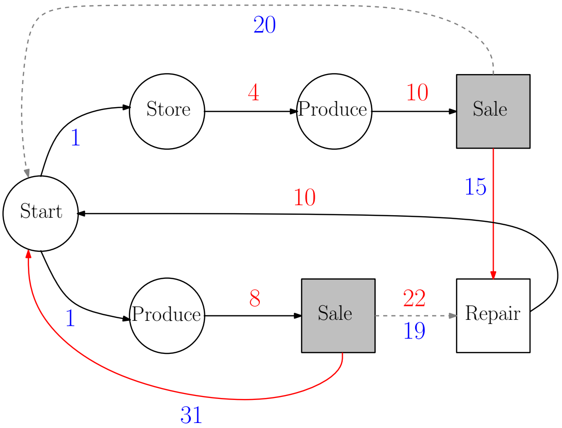
STRATEGIES OF PLAYER MIN: μ



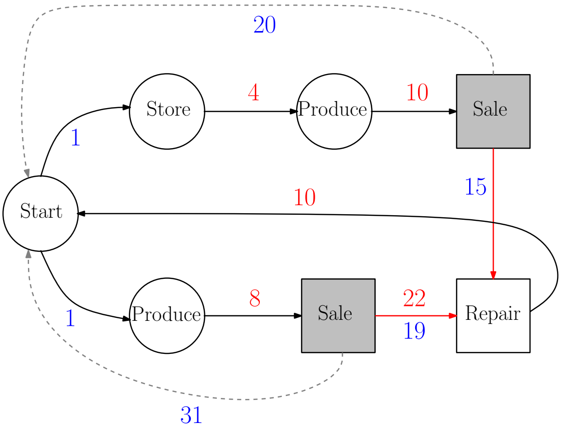
Strategies of player Max: χ



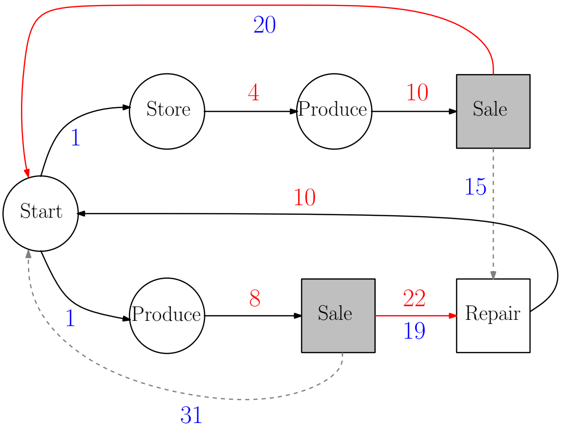
STRATEGIES OF PLAYER MAX: χ



Strategies of player Max: χ



STRATEGIES OF PLAYER MAX: χ



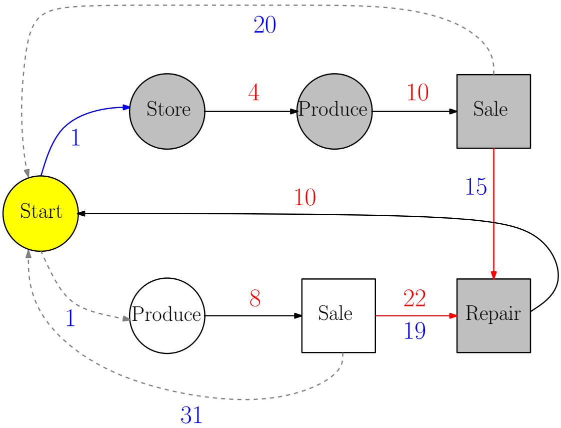
Runs and Payoffs

We are given

- Start state: s
- Strategy of player Min: μ
- Strategy of player Max: χ

This gives us

- A unique run: $\mathsf{Run}(s, \mu, \chi)$,
- for which we can compute the payoff: $\mathsf{AvgPpR}(\mathsf{Run}(s, \mu, \chi))$



Average Price-per-Reward: $\frac{0+4+10+0+10}{1+0+0+15+0} = \frac{3}{2}$

GAME VALUE

PLAYER MIN CAN GUARANTEE:

$$\mathsf{Val}^*(s) = \inf_{\mu} \sup_{\chi} \mathsf{AvgPpR}(\mathsf{Run}(s, \mu, \chi))$$

PLAYER MAX CAN GUARANTEE:

$$\mathsf{Val}_*(s) = \sup_{\chi} \inf_{\mu} \mathsf{AvgPpR}(\mathsf{Run}(s, \mu, \chi))$$

GAME VALUE (DETERMINACY)

- We always have: $\mathsf{Val}^*(s) \geqslant \mathsf{Val}_*(s)$.
- We want to prove: $\mathsf{Val}_*(s) \geqslant \mathsf{Val}^*(s)$ i.e. existence of $\mathsf{Val}(s)$.

Player Min can guarantee:

$$\mathsf{Val}^*(s) = \inf_{\mu} \sup_{\chi} \mathsf{AvgPpR}(\mathsf{Run}(s,\mu,\chi))$$

Player Max can guarantee:

$$\mathsf{Val}_*(s) = \sup_{\chi} \inf_{\mu} \mathsf{AvgPpR}(\mathsf{Run}(s,\mu,\chi))$$

Game value (determinacy)

- We always have: $\mathsf{Val}^*(s) \geqslant \mathsf{Val}_*(s)$.
- We want to prove: $\mathsf{Val}_*(s) \geqslant \mathsf{Val}^*(s)$ i.e. existence of $\mathsf{Val}(s)$.

Game value

Player Min can guarantee:

$$\mathsf{Val}^*(s) = \inf_{\mu} \sup_{\chi} \mathsf{AvgPpR}(\mathsf{Run}(s, \mu, \chi))$$

Player Max can guarantee:

$$\mathsf{Val}_*(s) = \sup_{\chi} \inf_{\mu} \mathsf{AvgPpR}(\mathsf{Run}(s, \mu, \chi))$$

Game Value (Determinacy)

- We always have: $\mathsf{Val}^*(s) \geqslant \mathsf{Val}_*(s)$.
- We want to prove: $\mathsf{Val}_*(s) \geqslant \mathsf{Val}^*(s)$ i.e. existence of $\mathsf{Val}(s)$.

Optimality equations

State of player Min

$$
\begin{aligned}
G(s) &= \min_{e}\{G(s')\} \\
B(s) &= \min_{e}\{\text{Price}(e) - \text{Reward}(e) \cdot G(s') + B(s') \ : \ G(s) = G(s')\}
\end{aligned}
$$

State of player Max

$$
\begin{aligned}
G(s) &= \max_{e}\{G(s')\} \\
B(s) &= \max_{e}\{\text{Price}(e) - \text{Reward}(e) \cdot G(s') + B(s') \ : \ G(s) = G(s')\}
\end{aligned}
$$

Optimality equations

State of player Min

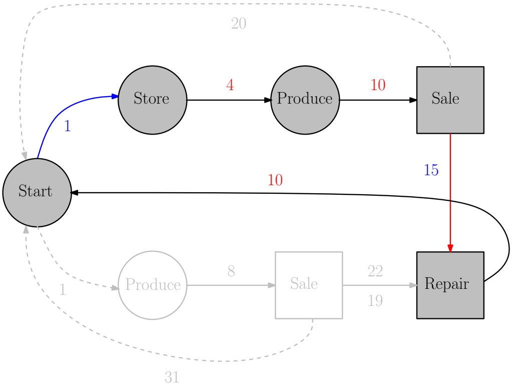
$$G(s) = \min_e \{G(s')\}$$

$$B(s) = \min_e \{\text{Price}(e) - \text{Reward}(e) \cdot G(s') + B(s') \ : \ G(s) = G(s')\}$$

State of player Max

$$G(s) = \max_e \{G(s')\}$$

$$B(s) = \max_e \{\text{Price}(e) - \text{Reward}(e) \cdot G(s') + B(s') \ : \ G(s) = G(s')\}$$



Average P-per-R: $\frac{0+4+10+0+10}{1+0+0+15+0} = \frac{3}{2}$

$$
\begin{aligned}
B(\mathsf{Start}) &= 0 - 1 \cdot g + B(\mathsf{Store}) \\
B(\mathsf{Store}) &= 4 - 0 \cdot g + B(\mathsf{Produce}) \\
B(\mathsf{Produce}) &= 10 - 0 \cdot g + B(\mathsf{Sale}) \\
B(\mathsf{Sale}) &= 0 - 15 \cdot g + B(\mathsf{Repair}) \\
+ \quad B(\mathsf{Repair}) &= 10 - 0 \cdot g + B(\mathsf{Start})
\end{aligned}
$$

$$(1+0+0+15+0) \cdot g = (0+4+10+0+10)$$

Summary

Results

Fact 1 If G, B satisfy optimality equations then: $G = \mathsf{Val}$.

Fact 2 G, B satisfying optimality equations can be computed.

Proof ideas

Proof 1 Use the "intuition" to prove that Min can guarantee at most G and player Max at least G.

Proof 2 Use strategy improvement algorithms

SUMMARY

RESULTS

FACT 1 If G, B satisfy optimality equations then: $G = \mathsf{Val}$.

FACT 2 G, B satisfying optimality equations can be computed.

PROOF IDEAS

PROOF 1 Use the "intuition" to prove that Min can guarantee at most G and player Max at least G.

PROOF 2 Use strategy improvement algorithms

1. Hybrid Automata

2. Average Price-per-Reward Games

3. Solving Hybrid Games

MAIN RESULT

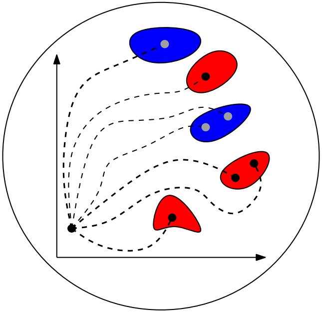
THEOREM

Hybrid average price-per-reward games are determined.

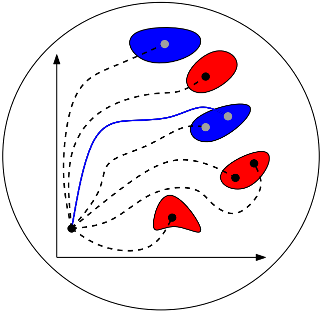
CONSEQUENCES

- We can approximate game values because optimality equations are first order expressible
- We can compute optimal strategies because they are definable.

Controller vs. Environment

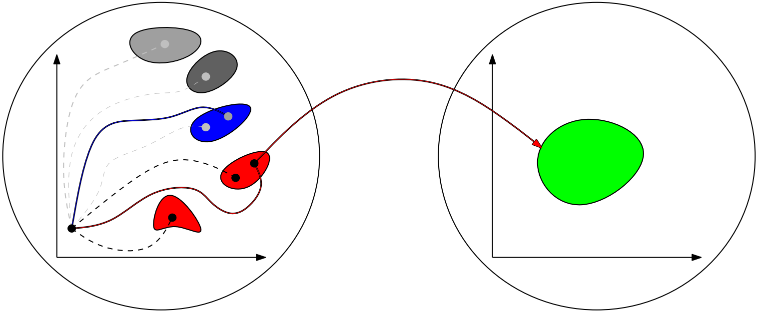


CONTROLLER VS. ENVIRONMENT



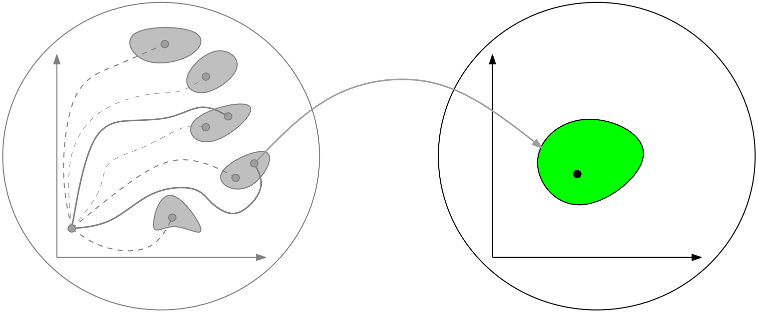
1. *Controller* chooses his timed action $\tau \in \mathsf{M}^{\mathrm{Min}}(s)$

Controller vs. Environment



1. *Controller* chooses his timed action $\tau \in \mathsf{M}^{\text{Min}}(s)$
2. *Environment* complies or chooses one of his s timed actions $\tau' \in \mathsf{M}^{\text{Max}}(s, \tau)$

Controller vs. Environment



1. *Controller* chooses his timed action $\tau \in \mathsf{M}^{\mathrm{Min}}(s)$
2. *Environment* complies or chooses one of his s timed actions $\tau' \in \mathsf{M}^{\mathrm{Max}}(s, \tau)$
3. *Environment* resets the variables, by choosing $s' \in \mathsf{R}(a)$, where $\tau' = (a, t)$

Summary

Solution idea

- Finite equivalence on the state space,
- Separation of continuous and discrete choices,
- Reduction to a problem on a "finite" graph,
- Optimality equations to prove correctness.

Context

- Average-price hybrid games are a special case,
- Simple reduction to finite games doesn't work,
- Protocol is the cause of complexity.

Solution - New model

"Finite" graphs, where prices (rewards) of transitions are determined in the course of a two player game.

Summary

Solution idea

- Finite equivalence on the state space,
- Separation of continuous and discrete choices,
- Reduction to a problem on a “finite” graph,
- Optimality equations to prove correctness.

Context

- Average-price hybrid games are a special case,
- Simple reduction to finite games doesn’t work,
- Protocol is the cause of complexity.

Solution - New model

“Finite” graphs, where prices (rewards) of transitions are determined in the course of a two player game.

Summary

Solution idea

- Finite equivalence on the state space,
- Separation of continuous and discrete choices,
- Reduction to a problem on a “finite” graph,
- Optimality equations to prove correctness.

Context

- Average-price hybrid games are a special case,
- Simple reduction to finite games doesn’t work,
- Protocol is the cause of complexity.

Solution - New model

“Finite” graphs, where prices (rewards) of transitions are determined in the course of a two player game.

Discrete-Continuous separation

Protocol

Each step of the protocol consists of a discrete and a continuous component:

1. Min: Proposes an action and a delay.
2. Max: Chooses the actual action and a delay.
3. Max: Resets the variables

Equivalence

Two states are equivalent if they admit the same discrete choices for both players.

Separation

1. Min and Max make the discrete choices. In step 3 Max chooses an equivalence class.
2. Min and Max make the corresponding continuous choices. In step 3 Max chooses an element of the chosen equivalence class.

Discrete-Continuous separation

Protocol

Each step of the protocol consists of a discrete and a continuous component:

1. Min: Proposes an action and a delay.
2. Max: Chooses the actual action and a delay.
3. Max: Resets the variables

Equivalence

Two states are equivalent if they admit the same discrete choices for both players.

Separation

1. Min and Max make the discrete choices. In step 3 Max chooses an equivalence class.
2. Min and Max make the corresponding continuous choices. In step 3 Max chooses an element of the chosen equivalence class.

Discrete-Continuous separation

Protocol

Each step of the protocol consists of a discrete and a continuous component:

1. Min: Proposes an action and a delay.
2. Max: Chooses the actual action and a delay.
3. Max: Resets the variables

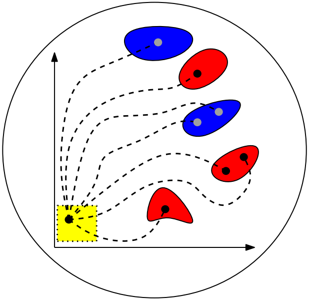
Equivalence

Two states are equivalent if they admit the same discrete choices for both players.

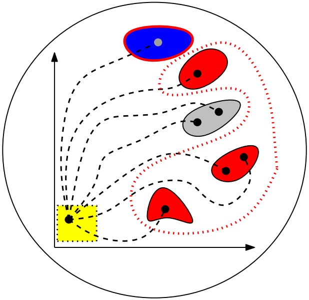
Separation

1. Min and Max make the discrete choices. In step 3 Max chooses an equivalence class.
2. Min and Max make the corresponding continuous choices. In step 3 Max chooses an element of the chosen equivalence class.

Protocol Abstraction

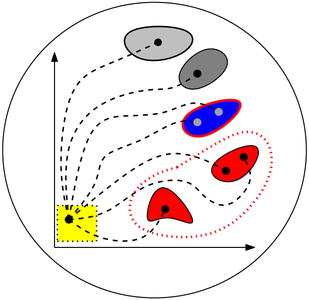


Protocol Abstraction



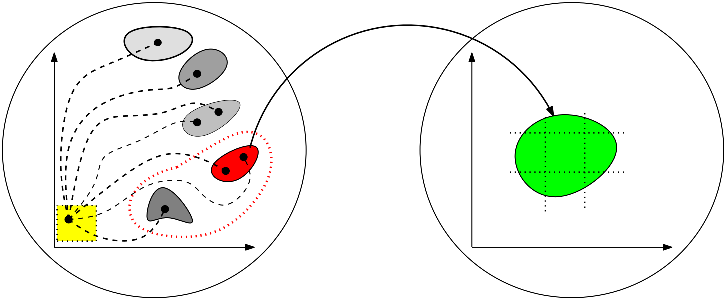
1. *Controller* chooses his action and a set of Environment's actions $(a, A) \in \mathsf{A} \times 2^{\mathsf{A}}$

PROTOCOL ABSTRACTION

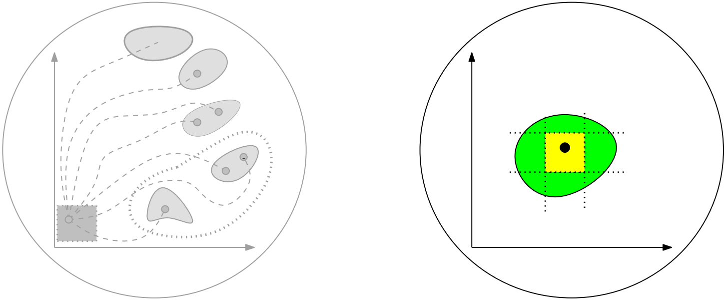


1. *Controller* chooses his action and a set of Environment's actions $(a, A) \in \mathsf{A} \times 2^{\mathsf{A}}$

PROTOCOL ABSTRACTION



1. *Controller* chooses his action and a set of Environment's actions $(a, A) \in \mathsf{A} \times 2^{\mathsf{A}}$
2. *Environment* chooses an action $a' \in \{a\} \cup A$



1. *Controller* chooses his action and a set of Environment's actions $(a, A) \in \mathsf{A} \times 2^{\mathsf{A}}$

2. *Environment* chooses an action $a' \in \{a\} \cup A$

3. *Environment* chooses $Q \in\sim$ such that $Q' \subseteq \mathsf{R}(a')$

Reduction

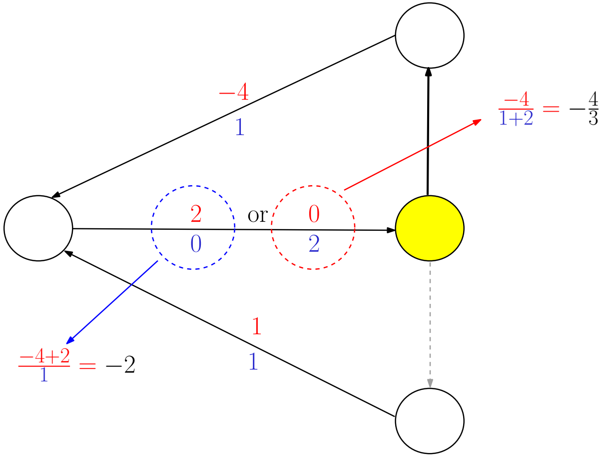
Finite graph

- The discrete choices are encoded in a finite graph
- Prices (rewards) depend on the continuous choices.

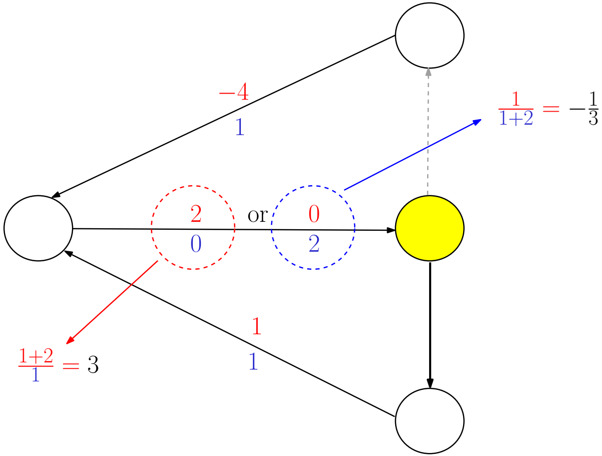
Issues

- In the special case of *average-price* games the discrete part of the protocol determined the optimal continuous part.
- In *average price-per-reward* optimal continuous choices depend on how the game is played.
- We resolve this problem by adding *edge games* to the graph. The price (reward) of an edge is determined in a two player game.
- The strategies in that game are exactly the continuous choices that players make in the protocol.

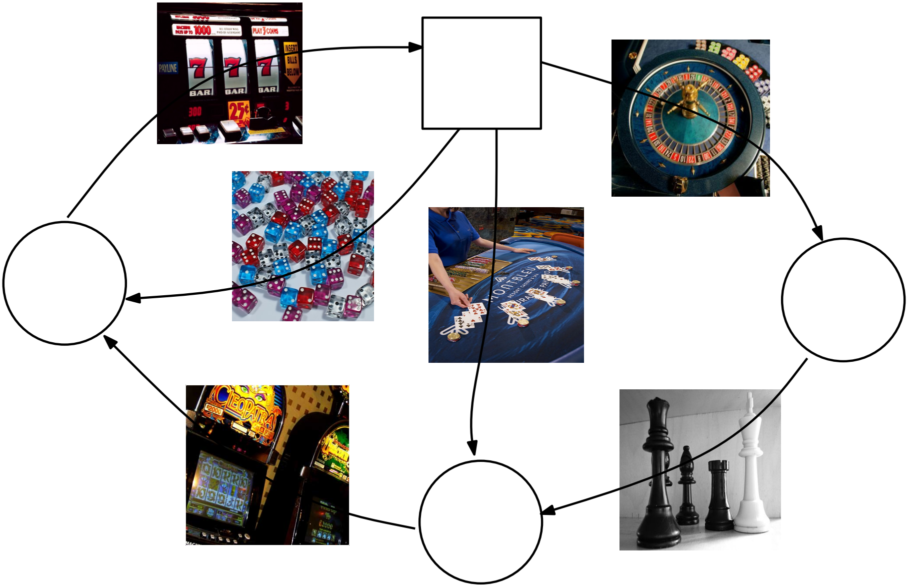
Example



Example



Graphs with edge games



REDUCTION

CORRECTNESS

- In both cases optimality equations characterise game values.
- In fact their solutions coincide.
- We want to show existence of solutions in our new model.

STRATEGY IMPROVEMENT

- We show that strategy improvement is correct, however, we can't guarantee termination.
- It is sufficient to show the existence of solutions for graphs of out-degree one.

GRAPHS OF OUT-DEGREE ONE

- **Assumptions:** price and reward are bounded, and the game is reward divergent.
- **Consequence:** *Lower* and *Upper* values are bounded.

REDUCTION

CORRECTNESS

- In both cases optimality equations characterise game values.
- In fact their solutions coincide.
- We want to show existence of solutions in our new model.

STRATEGY IMPROVEMENT

- We show that strategy improvement is correct, however, we can't guarantee termination.
- It is sufficient to show the existence of solutions for graphs of out-degree one.

GRAPHS OF OUT-DEGREE ONE

- **Assumptions:** price and reward are bounded, and the game is reward divergent.
- **Consequence:** *Lower* and *Upper* values are bounded.

REDUCTION

CORRECTNESS

- In both cases optimality equations characterise game values.
- In fact their solutions coincide.
- We want to show existence of solutions in our new model.

STRATEGY IMPROVEMENT

- We show that strategy improvement is correct, however, we can't guarantee termination.
- It is sufficient to show the existence of solutions for graphs of out-degree one.

GRAPHS OF OUT-DEGREE ONE

- **Assumptions:** price and reward are bounded, and the game is reward divergent.
- **Consequence:** *Lower* and *Upper* values are bounded.

DETERMINACY

THEOREM

Average price-per-reward game on graphs of out-degree one are determined.

PROOF SKETCH

By analysis of optimality equations we show that for every g either:

$$\mathsf{Val}_*(s) \geqslant g \text{ or } \mathsf{Val}^*(s) \leqslant g$$

Using "binary search" we prove that $\mathsf{Val}_* \geqslant \mathsf{Val}^*$ (recall that $\mathsf{Val}_* \leqslant \mathsf{Val}^*$).

COROLLARY

Determinacy extends to all graphs from correctness of strategy improvement.

CONCLUSIONS

RESULTS

- We have showed that *Average price-per-reward games* on hybrid systems with strong resets are determined.
- To prove this we reduced to a new class of finite graphs, that feature games on edges, and proved determinacy of respective games.
- Definability of hybrid systems and first-order expressibility of optiomality equations allows us to approximate game values.

Thank You