Games on graphs

Séminaire Thésards

Nathanaël Fijalkow

LIAFA, CNRS Université Denis Diderot - Paris 7, France

November 10th, 2010

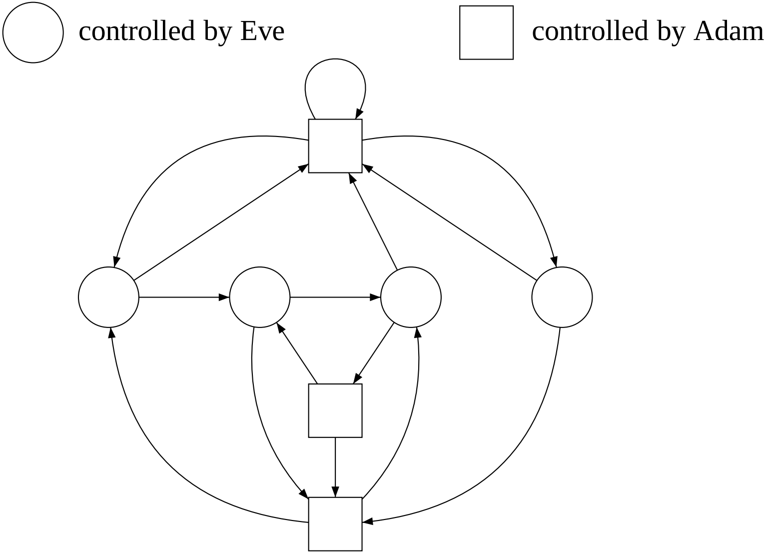
Outline

Outline

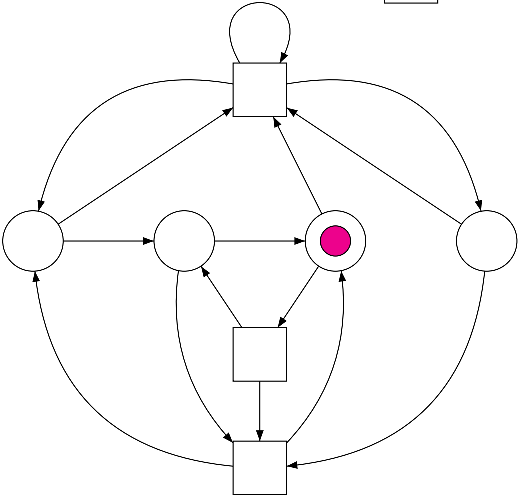
Players

Two players: **Eve** and **Adam**.

Plays



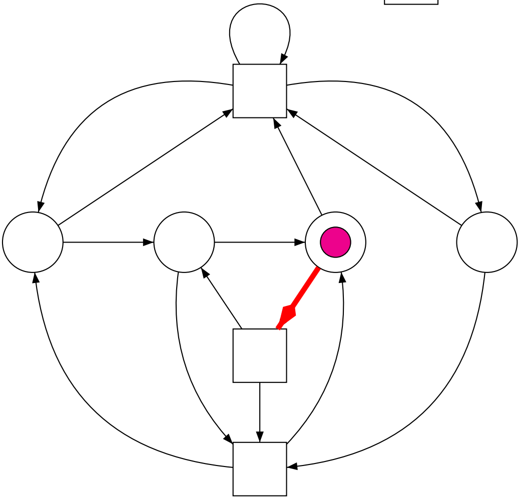
Plays



Plays

controlled by Eve

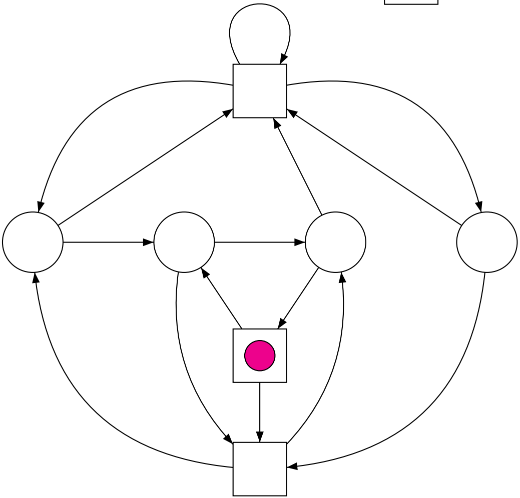
controlled by Adam



Plays

controlled by Eve

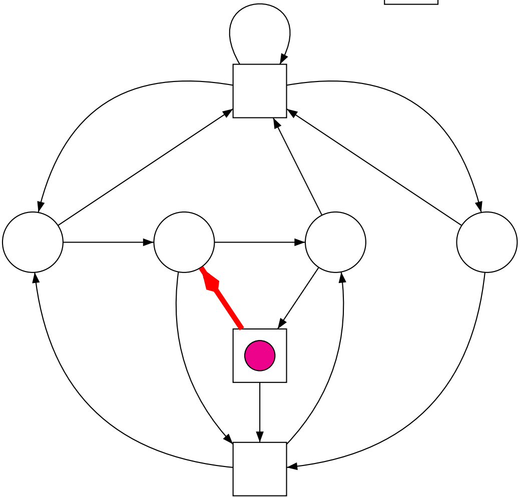
controlled by Adam



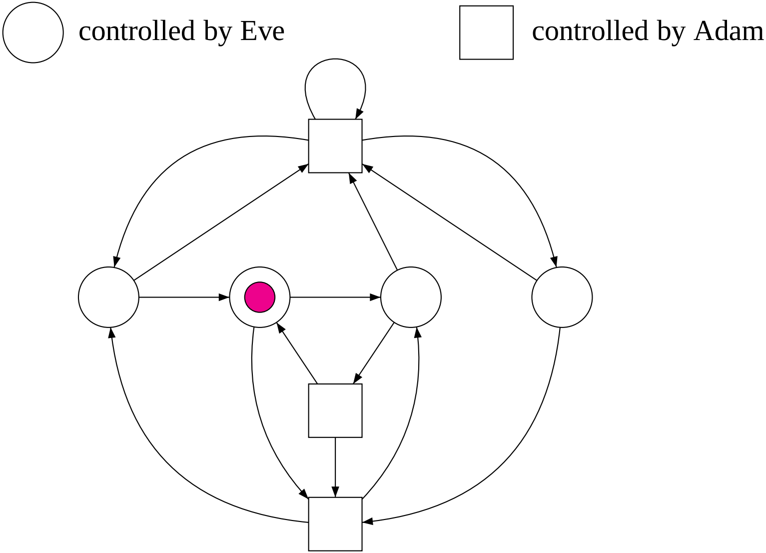
Plays

controlled by Eve

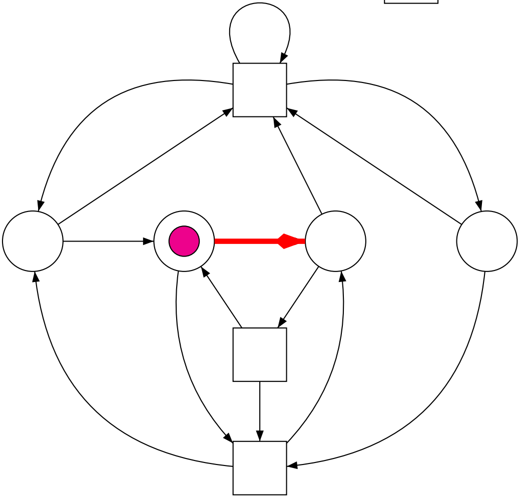
controlled by Adam



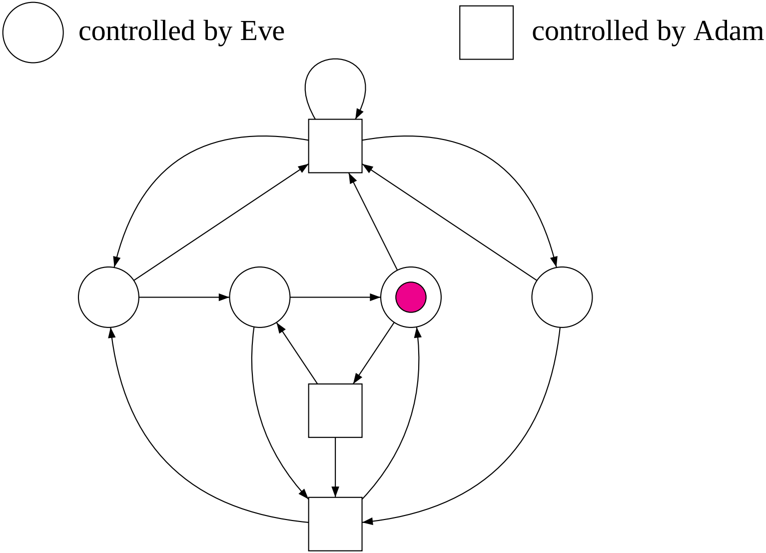
Plays



Plays



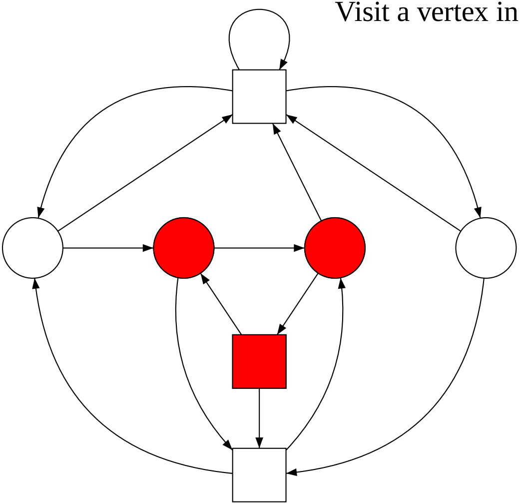
Plays



Outline

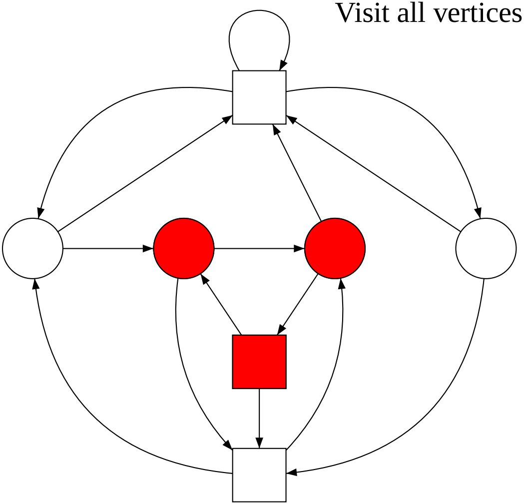
Reachability and Büchi objectives

Visit a vertex in F



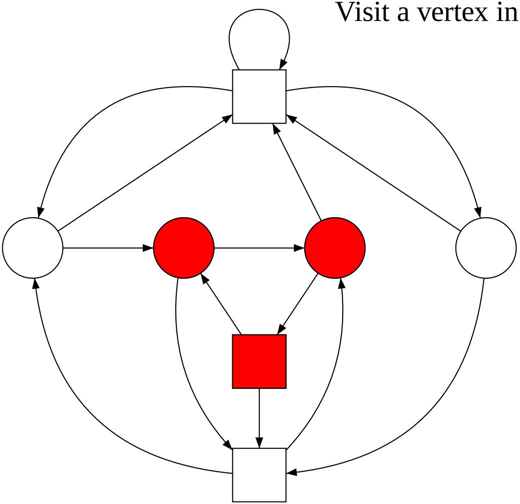
Reachability and Büchi objectives

Visit all vertices in F



Reachability and Büchi objectives

Visit a vertex in F infinitely often



Outline

Outline

1. Games
   - Playing
   - Winning
2. Motivations
   - Deciding tree automata membership
   - Deciding logics
3. Problems and tools
   - Deciding a winner
   - Strategy complexity
   - Stochastic games
   - Concurrent games

Theory

- A tree automaton $\mathcal{A}$ accepts a tree t if all branches of t satisfy a condition;

Theory

- A tree automaton $\mathcal{A}$ accepts a tree t if all branches of t satisfy a condition;
- Let us play on $\mathcal{A}$: Eve chooses transitions, Adam chooses nodes.
- Eve and Adam take turn to build a labeled branch: Eve wins if the ensued branch is accepting.
- $\mathcal{A}$ accepts t if and only if Eve wins the game.

Theory

- A tree automaton $\mathcal{A}$ accepts a tree t if all branches of t satisfy a condition;
- Let us play on $\mathcal{A}$: Eve chooses transitions, Adam chooses nodes.
- Eve and Adam take turn to build a labeled branch: Eve wins if the ensued branch is accepting.
- $\mathcal{A}$ accepts t if and only if Eve wins the game.

Assume we can construct finitely representable winning strategies for Adam. Then we can use them as witnesses to complement automata.

Outline

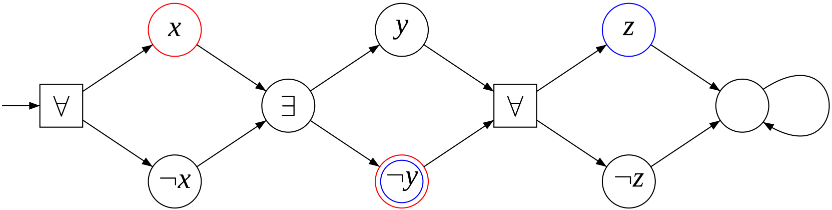
Deciding QBF

$$\phi = \forall x\, \exists y\, \forall z\ \ (x \vee \neg y) \wedge (\neg y \vee z)$$

Deciding QBF

$$\phi = \forall x\, \exists y\, \forall z\ (x \vee \neg y) \wedge (\neg y \vee z)$$

$$F_1 = \{x, \neg y\} \qquad F_2 = \{\neg y, z\}$$



Outline

Outline

The winner problem

Given a game and an initial vertex, who has a winning strategy?

The winner problem

Given a game and an initial vertex, who has a winning strategy?
Using crazy winning conditions we can even have non-determined games: neither Eve nor Adam has a winning strategy.

The winner problem

Given a game and an initial vertex, who has a winning strategy?

Using crazy winning conditions we can even have non-determined games: neither Eve nor Adam has a winning strategy.

Hence we focus on ω-regular winning conditions, that is given by a Büchi automaton.

For these games, the winner problem is decidable.

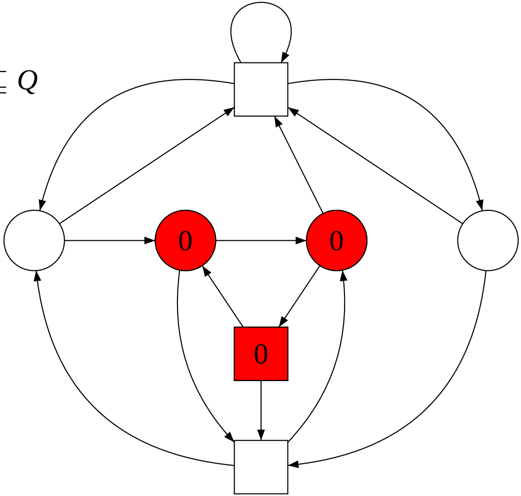
A sketchy sketch of a proof

Let $\mathbb{G}$ an arena and $\mathcal{A}$ a deterministic Büchi automaton.

- First compute the synchronous product $\mathbb{G} \times \mathcal{A}$. A play is winning if its second component is accepted by $\mathcal{A}$;
- The resulting game is a Büchi game.
- We are now left to solve Büchi games. This can be done efficiently (quadratic time in the size of the arena), using the following ideas:

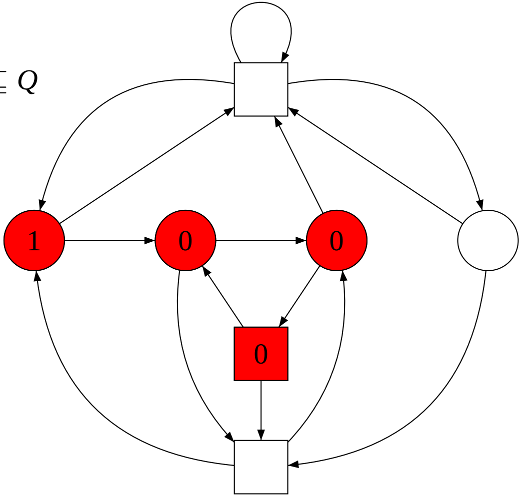
Solving reachability games: attractor computation

Given $F \subseteq Q$



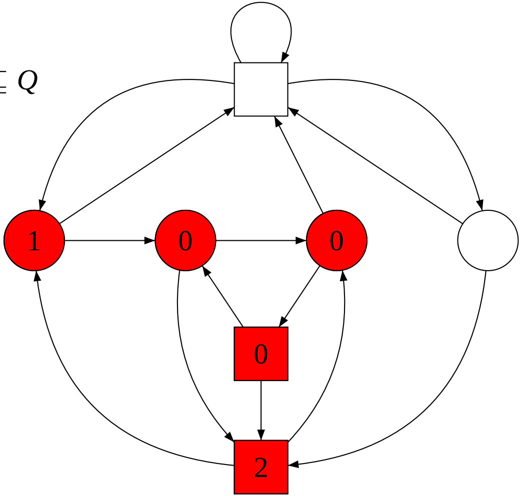
Solving reachability games: attractor computation

Given $F \subseteq Q$



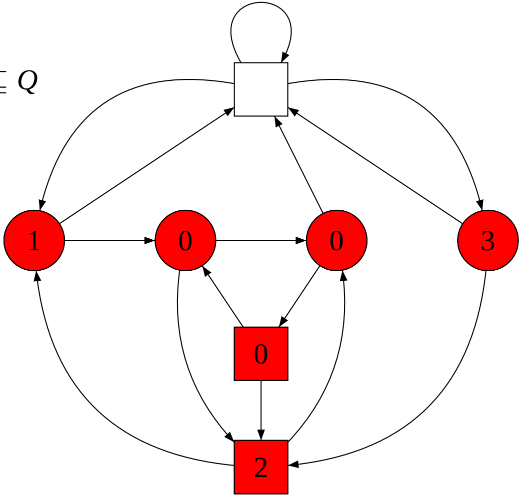
Solving reachability games: attractor computation

Given $F \subseteq Q$



Solving reachability games: attractor computation

Given $F \subseteq Q$



Solving Büchi games

If Eve can ensure to visit F a great number of times, then she can ensure to visit F infinitely often. Then we design a recursive algorithm to solve k-repeated reachability games.

Solving Büchi games

If Eve can ensure to visit F a great number of times, then she can ensure to visit F infinitely often. Then we design a recursive algorithm to solve k-repeated reachability games.
Informally, to ensure to reach F at least $k+1$ times, you have to ensure to reach it k times and then to ensure at least one visit.

Solving Büchi games

If Eve can ensure to visit F a great number of times, then she can ensure to visit F infinitely often. Then we design a recursive algorithm to solve k-repeated reachability games.
Informally, to ensure to reach F at least $k+1$ times, you have to ensure to reach it k times and then to ensure at least one visit.
We can show that for $k \geq n+1$, the k-repeated reachability games and Büchi games are equivalent.

Questions

- What about non-deterministic automata?
- Do these proofs allow to construct winning strategies?

Outline

1. Games
   - Playing
   - Winning
2. Motivations
   - Deciding tree automata membership
   - Deciding logics
3. Problems and tools
   - Deciding a winner
   - Strategy complexity
   - Stochastic games
   - Concurrent games

Different models of strategy

A strategy for a player in the most general form is a function that maps any finite history of a play to a chosen transition.

Different models of strategy

A strategy for a player in the most general form is a function that maps any finite history of a play to a chosen transition.
We are interested in finitely representable strategies that are sufficient to win the games we consider.

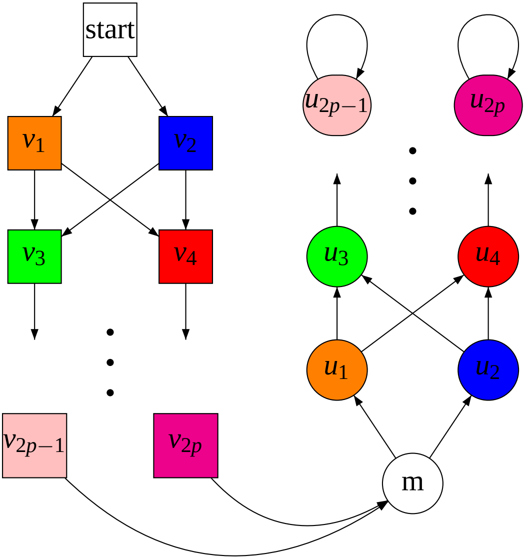
Different models of strategy

A strategy for a player in the most general form is a function that maps any finite history of a play to a chosen transition.

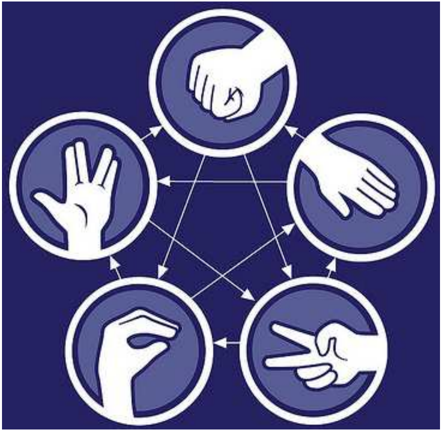
We are interested in finitely representable strategies that are sufficient to win the games we consider.

Strategies can use memory or randomization.

Why memory?



Why randomization?



Outline

Outline

1. Games
   - Playing
   - Winning
2. Motivations
   - Deciding tree automata membership
   - Deciding logics
3. Problems and tools
   - Deciding a winner
   - Strategy complexity
   - Stochastic games
   - Concurrent games

The end.

Thank for your attention!