Boundedness Games

Séminaire du LIGM, April 16th, 2013

Nathanaël Fijalkow

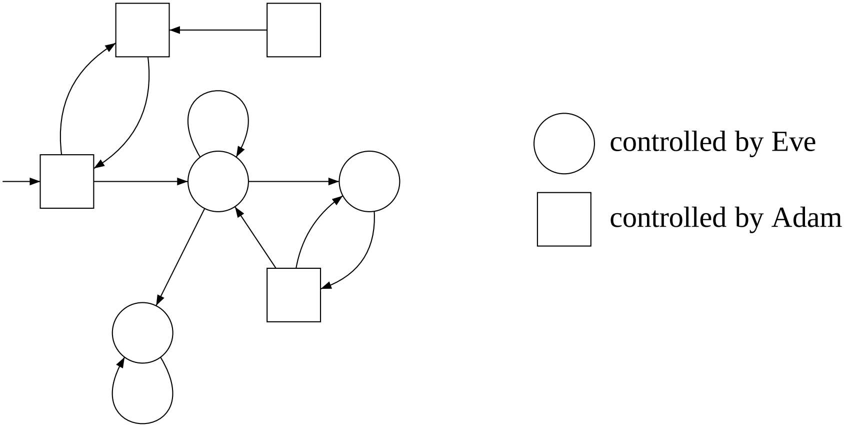
Institute of Informatics, Warsaw University – Poland

LIAFA, Université Paris 7 Denis Diderot – France

(based on joint works with Krishnendu Chatterjee, Thomas Colcombet, Florian Horn and Martin Zimmermann)

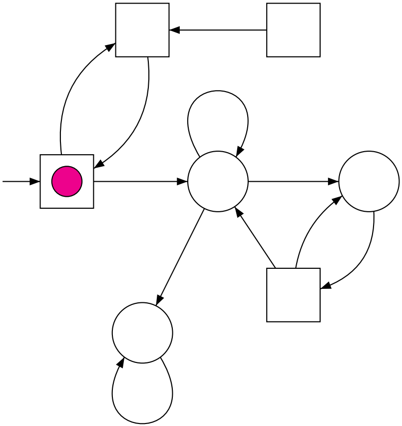
Definition of ωB-games

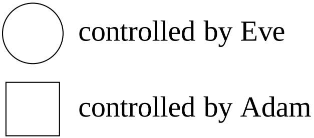
Two-player turn-based games over **finite** or **infinite** graphs



Definition of ωB-games

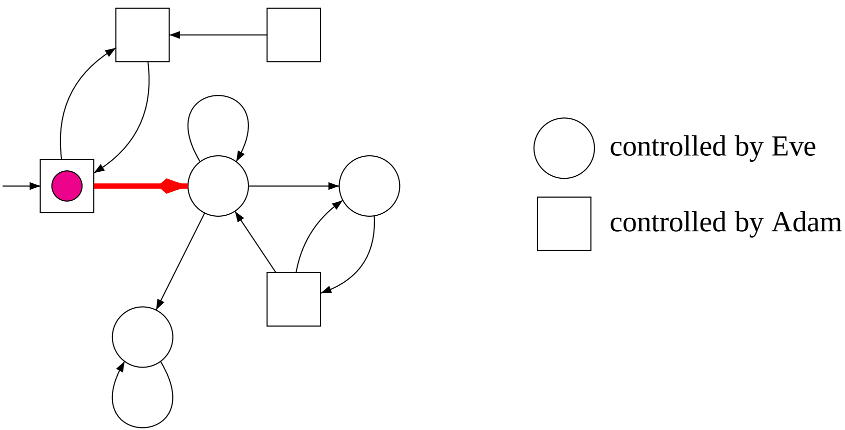
Two-player turn-based games over **finite** or **infinite** graphs





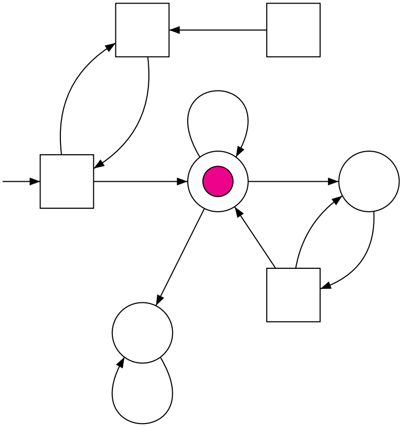
Definition of ωB-games

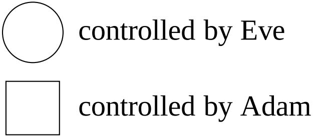
Two-player turn-based games over **finite** or **infinite** graphs



Definition of ωB-games

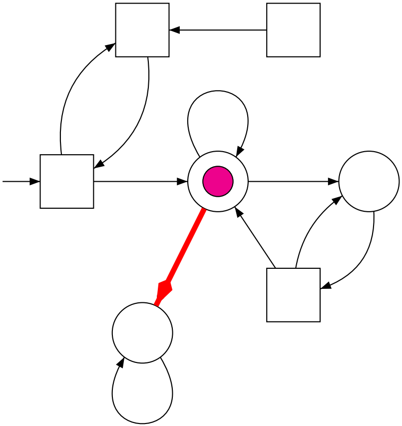
Two-player turn-based games over **finite** or **infinite** graphs

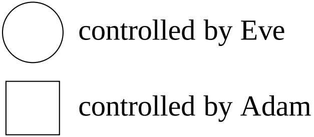




Definition of ωB-games

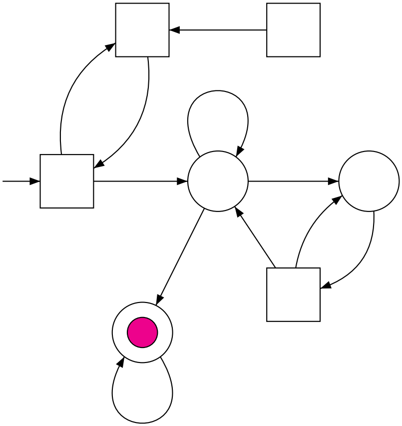
Two-player turn-based games over **finite** or **infinite** graphs

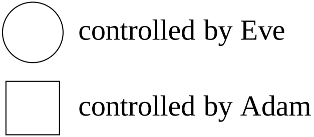




Definition of ωB-games

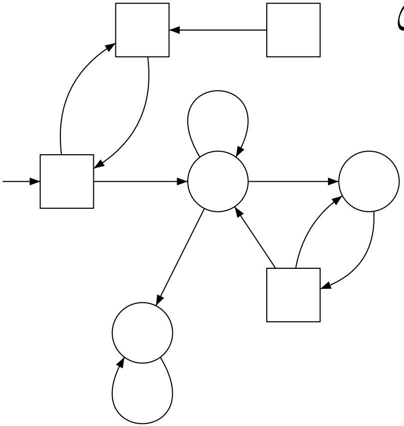
Two-player turn-based games over **finite** or **infinite** graphs





Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs

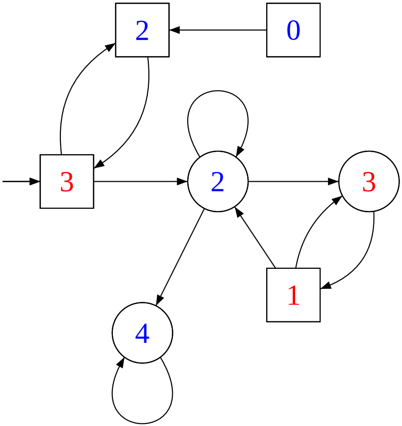


ωB winning condition:

parity
and
all counters
are bounded

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs

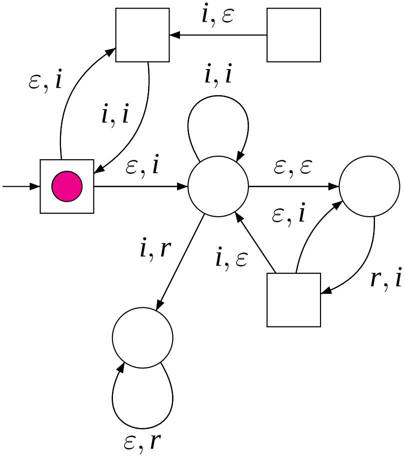


parity condition:

the minimal priority seen infinitely often is even

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$$c_1 = 0$$
$$c_2 = 0$$

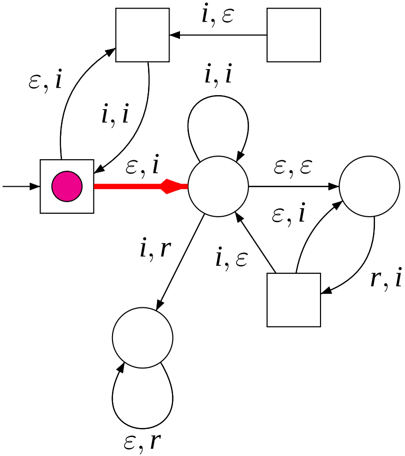
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$$c_1 = 0$$
$$c_2 = 0$$

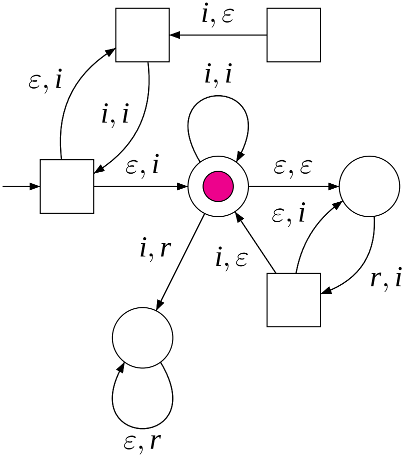
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$$c_1 = 0$$
$$c_2 = 1$$

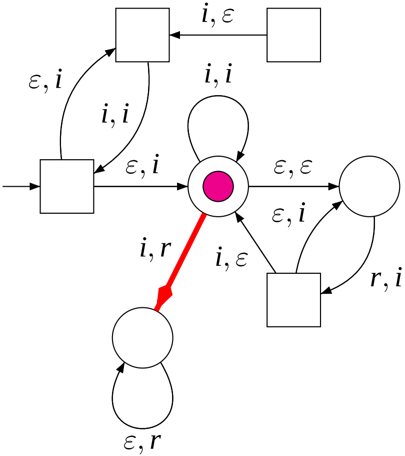
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$$c_1 = 0$$
$$c_2 = 1$$

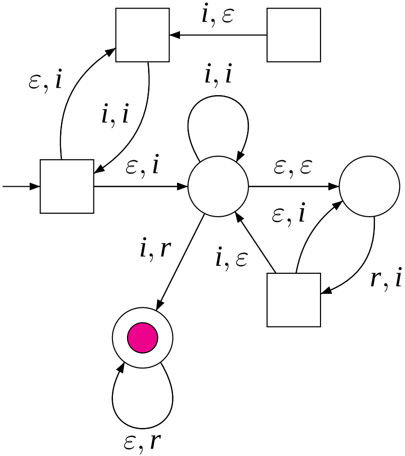
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$c_1 = 1$

$c_2 = 0$

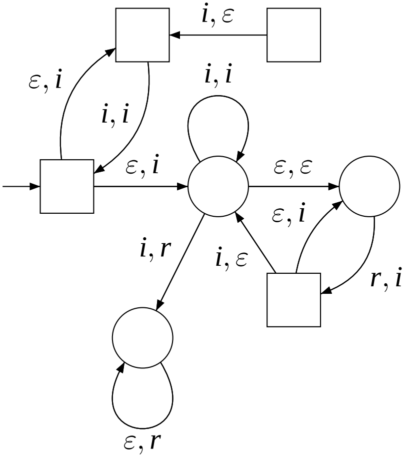
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



$$c_1 = 1$$
$$c_2 = 0$$

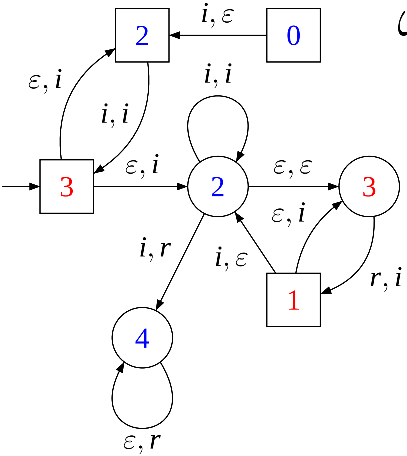
ε : nothing

i : increment

r : reset

Definition of ωB-games

Two-player turn-based games over **finite** or **infinite** graphs



ωB winning condition:

parity
and
all counters
are bounded

Strategy (for Eve)

General form

$$\sigma : V^+ \to V$$

Strategy (for Eve)

General form

$$\sigma : V^+ \to V$$

Positional or memoryless

$$\sigma : V \to V$$

Strategy (for Eve)

General form

$$\sigma : V^+ \to V$$

Positional or memoryless

$$\sigma : V \to V$$

Theorem (Müller and Schupp)

In parity games, both players have memoryless winning strategies.

Strategy (for Eve)

General form

$$\sigma : V^+ \to V$$

Positional or memoryless

$$\sigma : V \to V$$

Theorem (Müller and Schupp)

In parity games, both players have memoryless winning strategies.

What about ωB games?

Strategy (for Eve)

General form

$$\sigma : V^+ \to V$$

Positional or memoryless

$$\sigma : V \to V$$

Theorem (Müller and Schupp)

In parity games, both players have memoryless winning strategies.

What about ωB games?

Finite-memory

$$\begin{cases} \sigma : V \times M \to V \\ \mu : M \times E \to M \end{cases}$$

Why proving the existence of finite-memory strategies?

Thomas Colcombet's habilitation:

le fait 2.22 et en déduire que la domination entre formules de la logique monadique de coût est décidable sur les arbres infinis. Ainsi, la conjecture 9.2 implique la conjecture 9.1.

En fait, il est possible de pointer avec encore plus de précision où se trouve la difficulté. Si l'on cherche à démontrer la conjecture 9.2, tout comme dans le cas des arbres finis, le point crucial est l'existence de stratégies gagnantes à mémoire finie. Il suffirait d'établir la conjecture suivante.

Conjecture 9.3. *Les objectifs* `hB` $\wedge$ `parité` *et* $\neg$`B` $\wedge$ `parité` *sont à* $\approx$*-mémoire finie, sur toutes les arènes/sur les arènes chronologiques/sur les arènes «arborescentes».*

Existence of finite-memory strategies in (some) boundedness games
$\Longrightarrow$ Decidability of cost MSO over infinite trees
$\Longrightarrow$ Decidability of the index of the non-deterministic Mostowski's hierarchy!

Quantification

Eve wins means:

$\exists\sigma$ (strategy for Eve),
$\forall\pi$ (paths),
$\exists N \in \mathbb{N}$,

$\exists\sigma$ (strategy for Eve),
$\exists N \in \mathbb{N}$,
$\forall\pi$ (paths),

π satisfies parity and each counter is bounded by N.

Quantification

Eve wins means:

$\exists \sigma$ (strategy for Eve),
$\forall \pi$ (paths),
$\exists N \in \mathbb{N}$,

$\exists \sigma$ (strategy for Eve),
$\exists N \in \mathbb{N}$,
$\forall \pi$ (paths),

π satisfies parity and each counter is bounded by N.

Is this:

1. Equivalent?
2. Decidable?

Quantification

Eve wins means:

$\exists \sigma$ (strategy for Eve),
$\forall \pi$ (paths),
$\exists N \in \mathbb{N}$,

$\exists \sigma$ (strategy for Eve),
$\exists N \in \mathbb{N}$,
$\forall \pi$ (paths),

π satisfies parity and each counter is bounded by N.

Is this:

1. Equivalent? Sometimes ...
2. Decidable? Not always ...

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.
   ↬ Cost-parity games [F. and Zimmermann, 2012].

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.
   ↬ Cost-parity games [F. and Zimmermann, 2012].
2. Over pushdown graphs: decide the winner and construct finite-memory strategies.

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.
   $\looparrowright$ Cost-parity games [F. and Zimmermann, 2012].
2. Over pushdown graphs: decide the winner and construct finite-memory strategies.
   $\looparrowright$ Pushdown finitary games [Chatterjee and F., 2013].

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.
   $\looparrowright$ Cost-parity games [F. and Zimmermann, 2012].
2. Over pushdown graphs: decide the winner and construct finite-memory strategies.
   $\looparrowright$ Pushdown finitary games [Chatterjee and F., 2013].
3. Over infinite graphs: prove the existence of finite-memory winning strategies.

The questions I am interested in

Boundedness games:

1. Over finite graphs: decide the winner efficiently and construct small finite-memory strategies.
   ↬ Cost-parity games [F. and Zimmermann, 2012].
2. Over pushdown graphs: decide the winner and construct finite-memory strategies.
   ↬ Pushdown finitary games [Chatterjee and F., 2013].
3. Over infinite graphs: prove the existence of finite-memory winning strategies.
   ↬ Ongoing work with Thomas Colcombet and Florian Horn.

Outline

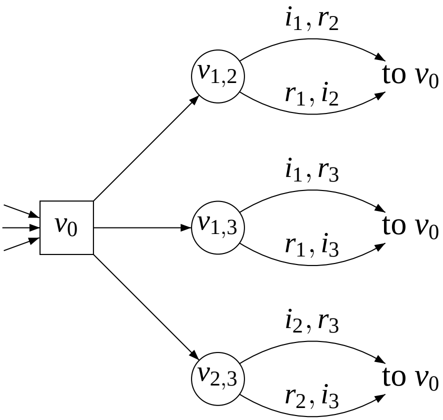
1. Finite-memory strategies
   - Some examples
   - Worst-case strategies
   - Finitary conditions

2. Equivalence for pushdown games
   - The case of finite graphs
   - The case of pushdown graphs
   - Application: ωB-games with max

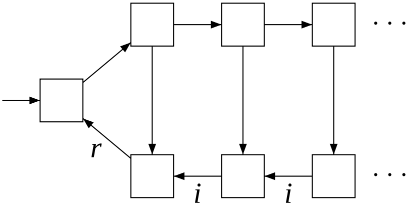
Outline

Outline

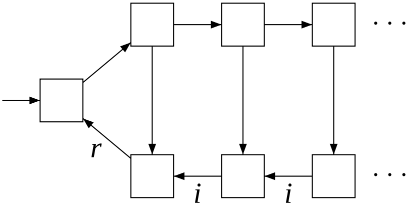
Eve needs some memory

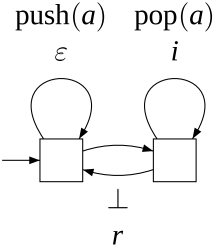


Adam needs infinite memory

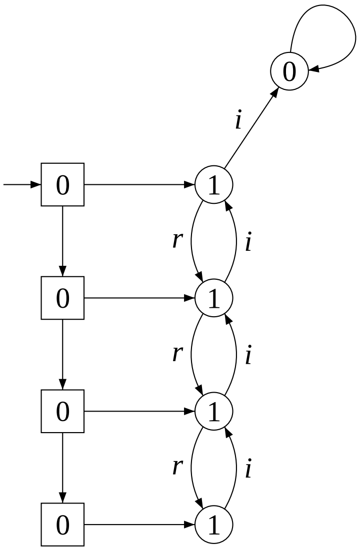


Adam needs infinite memory

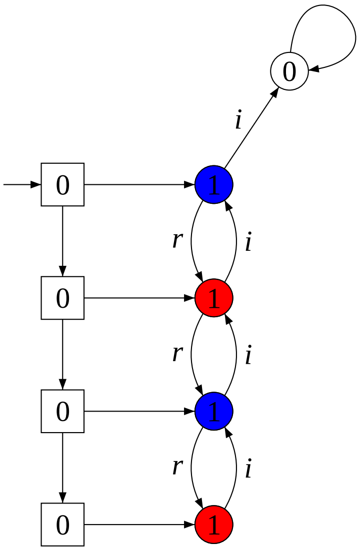




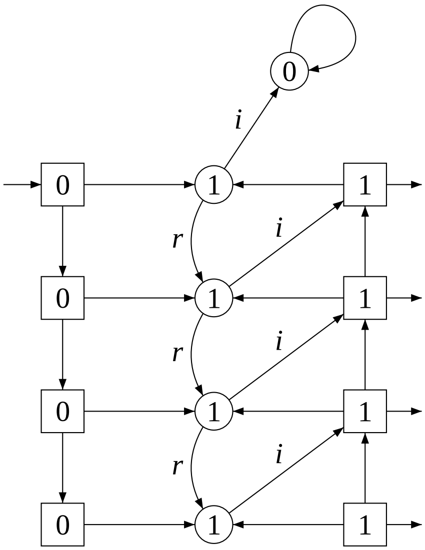
Eve needs infinite memory



Eve needs infinite memory



Eve needs infinite memory



Outline

Playing for the worst, hoping for the best

Condition: the only counter is bounded by N.

Observation

For every vertex v, there is a maximal $N(v) \leq N$ such that Eve wins.

Idea: play from v as you would with counter value $N(v)$.

Playing for the worst, hoping for the best

Condition: the only counter is bounded by N.

Observation

For every vertex v, there is a maximal $N(v) \leq N$ such that Eve wins.

Idea: play from v as you would with counter value $N(v)$.

Bottom-line: possible scenarios are *linearly ordered*.

Playing for the worst, hoping for the best

Condition: the only counter is bounded by N.

Observation

For every vertex v, there is a maximal $N(v) \leq N$ *such that Eve wins.*

Idea: play from v as you would with counter value $N(v)$.

Bottom-line: possible scenarios are *linearly ordered*.

Lemma

Eve has positional winning strategies for the condition "the counter is bounded by N*".*

Outline

Finitary Büchi

Condition: there exists N such that Büchi vertices are visited every N steps.

Finitary Büchi

Condition: there exists N such that Büchi vertices are visited every N steps.
(roughly speaking: actions i and r, no ε)

Finitary Büchi

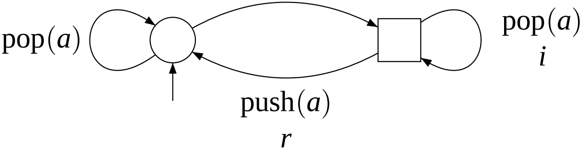
Condition: there exists N such that Büchi vertices are visited every N steps.
(roughly speaking: actions i and r, no ε)

Theorem

- *Eve has positional winning strategies in finitary Büchi games.*
- *Eve has finite-memory winning strategies in finitary parity games.*

Outline

Some more examples (1)



Eve should maintain a low stack.

Objective

Theorem

For all pushdown games, the following are equivalent:

- $\exists\sigma$ *(strategy for Eve)*, $\forall\pi$ *(paths)*, $\exists N \in \mathbb{N}$,
  π *satisfies parity and each counter is bounded by* N.
- $\exists\sigma$ *(strategy for Eve)*, $\exists N \in \mathbb{N}$, $\forall\pi$ *(paths)*,
  π *satisfies parity and* ***eventually*** *each counter is bounded by* N.

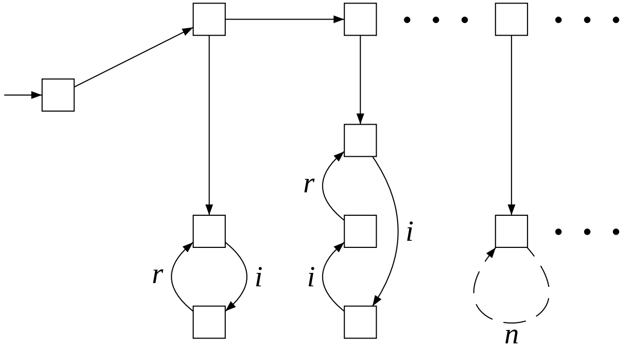
Objective

Theorem

For all pushdown games, the following are equivalent:

- $\exists\sigma$ *(strategy for Eve),* $\forall\pi$ *(paths),* $\exists N \in \mathbb{N}$,
  π *satisfies parity and each counter is bounded by* N.
- $\exists\sigma$ *(strategy for Eve),* $\exists N \in \mathbb{N}$, $\forall\pi$ *(paths),*
  π *satisfies parity and* ***eventually*** *each counter is bounded by* N.

Counter-example for the general case



Eve wins but she does not know the bound!

Outline

A simple proof for the case of finite graphs

Condition: parity and all counters are bounded.

Define:

- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

A simple proof for the case of finite graphs

Condition: parity and all counters are bounded.

Define:

- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

Lemma

1. $\mathcal{W}_E(0) \subseteq \mathcal{W}_E(1) \subseteq \cdots \subseteq \mathcal{W}_E(N) \subseteq \mathcal{W}_E(N+1) \subseteq \cdots \subseteq \mathcal{W}_E$.

A simple proof for the case of finite graphs

Condition: parity and all counters are bounded.

Define:

- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

Lemma

1. $\mathcal{W}_E(0) \subseteq \mathcal{W}_E(1) \subseteq \cdots \subseteq \mathcal{W}_E(N) \subseteq \mathcal{W}_E(N+1) \subseteq \cdots \subseteq \mathcal{W}_E$.
2. *There exists N such that $\mathcal{W}_E(N) = \mathcal{W}_E(N+1) = \cdots$.*

A simple proof for the case of finite graphs

Condition: parity and all counters are bounded.

Define:

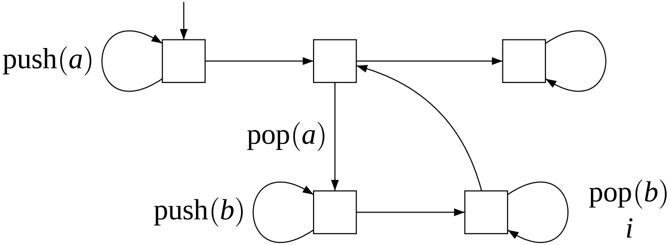
- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

Lemma

1. $\mathcal{W}_E(0) \subseteq \mathcal{W}_E(1) \subseteq \cdots \subseteq \mathcal{W}_E(N) \subseteq \mathcal{W}_E(N+1) \subseteq \cdots \subseteq \mathcal{W}_E$.
2. *There exists N such that $\mathcal{W}_E(N) = \mathcal{W}_E(N+1) = \cdots$.*
3. *For such N, Adam wins from $V \setminus \mathcal{W}_E(N)$, hence $\mathcal{W}_E = \mathcal{W}_E(N)$.*

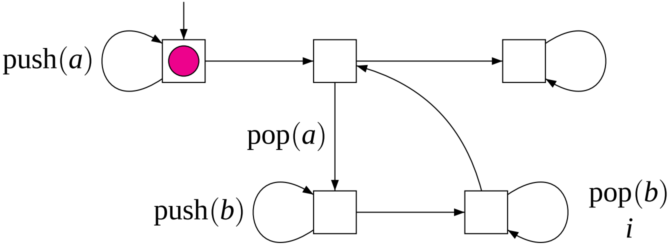
Outline

Some more examples (2)



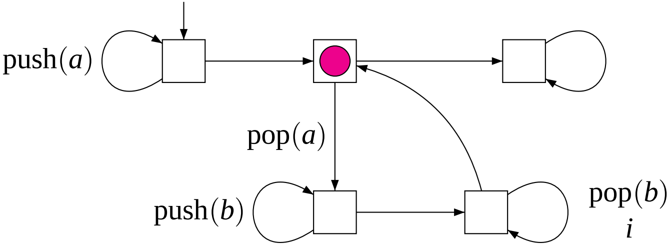
Adam may use the stack as “credit”.

Some more examples (2)



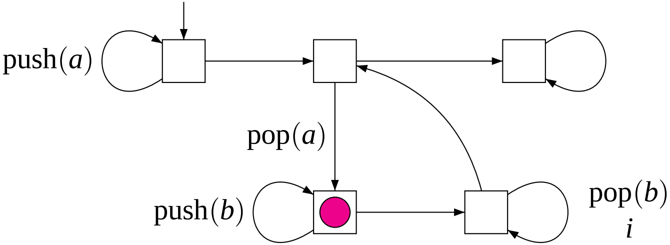
Adam may use the stack as “credit”.

Some more examples (2)



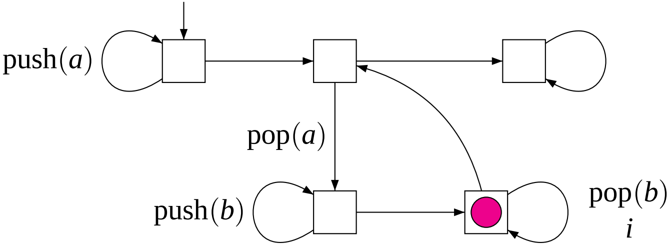
Adam may use the stack as “credit”.

Some more examples (2)



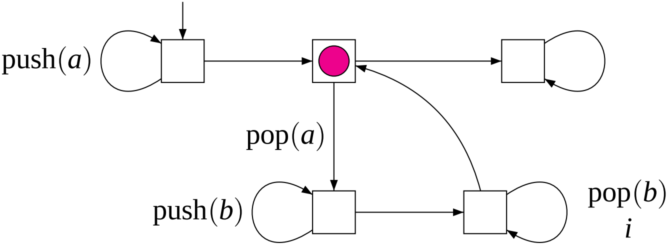
Adam may use the stack as “credit”.

Some more examples (2)



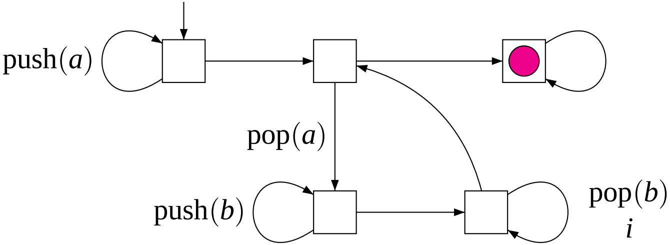
Adam may use the stack as “credit”.

Some more examples (2)



Adam may use the stack as “credit”.

Some more examples (2)



Adam may use the stack as “credit”.

Proof sketch

Condition: parity and all counters are bounded.

Define:

- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N **in the limit**.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

Proof sketch

Condition: parity and all counters are bounded.

Define:

- $\mathcal{W}_E(N)$ the set of vertices where Eve wins for the bound N **in the limit**.
- $\mathcal{W}_E$ the set of vertices where Eve wins for some (non-uniform) bound.

Proposition

1. $\mathcal{W}_E(0) \subseteq \mathcal{W}_E(1) \subseteq \cdots \subseteq \mathcal{W}_E(N) \subseteq \mathcal{W}_E(N+1) \subseteq \cdots \subseteq \mathcal{W}_E$.
2. *There exists N such that $\mathcal{W}_E(N) = \mathcal{W}_E(N+1) = \cdots$.*
3. *For such N, Adam wins from $V \setminus \mathcal{W}_E(N)$, hence $\mathcal{W}_E = \mathcal{W}_E(N)$.*

Why is 2. true?

Regularity of the winning regions

Theorem (derived from Serre)

*For all N, $\mathcal{W}_E(N)$ is a regular set of configurations, recognized by an alternating automaton of size **independent of** N.*

Decidability

Theorem

For all pushdown games, the following are equivalent:

- $\exists\sigma$ *(strategy for Eve),* $\forall\pi$ *(paths),* $\exists N \in \mathbb{N}$,
  π *satisfies parity and each counter is bounded by* N.
- $\exists\sigma$ *(strategy for Eve),* $\exists N \in \mathbb{N}$, $\forall\pi$ *(paths),*
  π *satisfies parity and* ***eventually*** *each counter is bounded by* N.

Decidability

Theorem

For all pushdown games, the following are equivalent:

- $\exists\sigma$ *(strategy for Eve),* $\forall\pi$ *(paths),* $\exists N \in \mathbb{N}$,
  π *satisfies parity and each counter is bounded by* N.
- $\exists\sigma$ *(strategy for Eve),* $\exists N \in \mathbb{N}$, $\forall\pi$ *(paths),*
  π *satisfies parity and* ***eventually*** *each counter is bounded by* N.

Corollary

Determining the winner in a pushdown ωB*-game is decidable.*

Decidability

Theorem

For all pushdown games, the following are equivalent:

- $\exists \sigma$ *(strategy for Eve),* $\forall \pi$ *(paths),* $\exists N \in \mathbb{N}$*,*
  π *satisfies parity and each counter is bounded by* N*.*
- $\exists \sigma$ *(strategy for Eve),* $\exists N \in \mathbb{N}$*,* $\forall \pi$ *(paths),*
  π *satisfies parity and* ***eventually*** *each counter is bounded by* N*.*

Corollary

Determining the winner in a pushdown ωB*-game is decidable.*

Remark: one can show that the collapse bound is doubly-exponential!

Outline

The max operator

We add a new feature for counters: $\gamma \leftarrow \max(\gamma_1, \gamma_2)$.

Theorem (derived from Bojańczyk and Toruńczyk)

Deterministic max-automata are equivalent to Weak MSO + $\mathbb{U}$.

We consider ωB-games with max.

Co-determinisation

$$
\begin{tikzcd}
0 \arrow[r, "i"] & 1 \arrow[r, "r"] & 0 \arrow[r] & 0 \arrow[r, "\max(\gamma_1, \gamma_2)"] & 2 \arrow[r, "i"] & 3 \\
0 \arrow[r] & 0 \arrow[r, "i"] & 1 \arrow[r, "\max(\gamma_2, \gamma_3)"] & 2 \arrow[r, "i"] \arrow[ur, dashed] & 3 \arrow[r, "r"] \arrow[dr, dashed] & 0 \\
0 \arrow[r, "i"] & 1 \arrow[r, "i"] & 2 \arrow[r, "i"] \arrow[ur, dashed] & 3 \arrow[r, "r"] & 0 \arrow[r, "\max(\gamma_1, \gamma_2)"'] & 3
\end{tikzcd}
$$

Co-determinisation

$$
\begin{tikzcd}
0 \arrow[r, "i"] & 1 \arrow[r, "r"] & 0 \arrow[r] & 0 \arrow[r, "\max(\gamma_1,\gamma_2)"] & 2 \arrow[r, "i"] & 3 \\
0 \arrow[r] & 0 \arrow[r, "i"] & 1 \arrow[r, "\max(\gamma_2,\gamma_3)"] & 2 \arrow[r, "i", red] \arrow[ur, dashed] & 3 \arrow[r, "r"] \arrow[dr, red] & 0 \\
0 \arrow[r, "i", red] & 1 \arrow[r, "i", red] & 2 \arrow[r, "i"] \arrow[ur, red] & 3 \arrow[r, "r"] & 0 \arrow[r, "\max(\gamma_1,\gamma_2)"'] & 3
\end{tikzcd}
$$

To prove that a counter value is high, one can count backwards!

Reduction

We reduce ωB-games with max to pushdown ωB-games (without max).

Reduction

We reduce ωB-games with max to pushdown ωB-games (without max).

Idea: simulate the game and store the play in the stack.

Whenever he wants, Adam can declare “this counter has a very large value”: from there, play backwards using the stack until a reset is met.

Reduction

We reduce ωB-games with max to pushdown ωB-games (without max).

Idea: simulate the game and store the play in the stack.

Whenever he wants, Adam can declare “this counter has a very large value”: from there, play backwards using the stack until a reset is met.

Theorem

Determining the winner in an ωB-game with max is decidable.

The end.

Thank you!

Outline

1. Finite-memory strategies
   - Some examples
   - Worst-case strategies
   - Finitary conditions

2. Equivalence for pushdown games
   - The case of finite graphs
   - The case of pushdown graphs
   - Application: ωB-games with max