How Much Lookahead is Needed to Win Infinite Games?

(Partially) joint work with Felix Klein (Saarland University)

Martin Zimmermann

Saarland University

August 26th, 2015

Aalborg University, Aalborg, Denmark

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b

O:

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b

O: a

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L\text{, if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b a

O: a

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b a

O: a a

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b
O:	a	a	

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b a b $\cdots$
O: a a $\cdots$

I wins

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I: b a b $\cdots$
O: a a $\cdots$

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$	I:	b
O:	a	a	$\cdots$		O:	

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a
O:	a	a	$\cdots$			O:		

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a	b
O:	a	a	$\cdots$			O:			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$	I:	b	a	b	
O:	a	a	$\cdots$		O:	b			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a	b	b
O:	a	a	$\cdots$			O:	b			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a	b	b
O:	a	a	$\cdots$			O:	b	b		

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$	I:	b	a	b	b	a
O:	a	a	$\cdots$		O:	b	b			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a	b	b	a
O:	a	a	$\cdots$			O:	b	b	a		

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$		I:	b	a	b	b	a	b
O:	a	a	$\cdots$			O:	b	b	a			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$
O:	a	a	$\cdots$	

I:	b	a	b	b	a	b
O:	b	b	a	b		

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$
O:	a	a	$\cdots$	

I:	b	a	b	b	a	b	a
O:	b	b	a	b			

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$
O:	a	a	$\cdots$	

I:	b	a	b	b	a	b	a
O:	b	b	a	b	a		

I wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

$$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L, \text{ if } \beta(i) = \alpha(i+2) \text{ for every } i$$

I:	b	a	b	$\cdots$				
O:	a	a	$\cdots$					

I wins

I:	b	a	b	b	a	b	a	$\cdots$
O:	b	b	a	b	a	$\cdots$		

O wins

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two:
  Interaction: one player may delay her moves.
  Winning condition: quantitative instead of qualitative.

The Delay Game $\Gamma_f(L)$

- Delay function: $f \colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).

The Delay Game $\Gamma_f(L)$

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).

The Delay Game $\Gamma_f(L)$

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

The Delay Game $\Gamma_f(L)$

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Definition:

- f is constant, if $f(i) = 1$ for every $i > 0$.
- f is bounded, if $f(i) = 1$ for almost all i.

The Delay Game $\Gamma_f(L)$

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Definition:

- f is constant, if $f(i) = 1$ for every $i > 0$.
- f is bounded, if $f(i) = 1$ for almost all i.

Questions we are interested in:

- Given L, is there an f such that O wins $\Gamma_f(L)$?
- How *large* does f have to be?
- How hard is the problem to solve?

Another Example

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Another Example

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Define language L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Another Example

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Define language L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

O wins $\Gamma_f(L_0)$ for every unbounded f:

- If I produces arbitrarily long input blocks, then the lookahead will contain arbitrarily long input blocks.
- Thus, O can produce arbitrarily long output blocks.

Previous Results

Theorem (Hosch & Landweber '72)

The following problem is decidable: Given ω-regular L, does O win $\Gamma_f(L)$ for some constant f?

Previous Results

Theorem (Hosch & Landweber '72)

The following problem is decidable: Given ω-regular L, does O win $\Gamma_f(L)$ for some constant f?

Theorem (Holtmann, Kaiser & Thomas '10)

1. *TFAE for L given by deterministic parity automaton $\mathcal{A}$:*
   - *O wins $\Gamma_f(L)$ for some f.*
   - *O wins $\Gamma_f(L)$ for some constant f with $f(0) \leq 2^{2^{|\mathcal{A}|}}$.*
2. *Deciding whether this is the case is in $2\textsc{ExpTime}$.*

Previous Results

Theorem (Hosch & Landweber '72)

The following problem is decidable: Given ω-regular L, does O win $\Gamma_f(L)$ for some constant f?

Theorem (Holtmann, Kaiser & Thomas '10)

1. *TFAE for L given by deterministic parity automaton $\mathcal{A}$:*
   - *O wins $\Gamma_f(L)$ for some f.*
   - *O wins $\Gamma_f(L)$ for some constant f with $f(0) \leq 2^{2^{|\mathcal{A}|}}$.*
2. *Deciding whether this is the case is in $2\textsc{ExpTime}$.*

Theorem (Fridman, Löding & Z. '11)

The following problem is undecidable: Given (one-counter, weak, visibly, deterministic) context-free L, does O win $\Gamma_f(L)$ for some f?

Uniformization of Relations

- A strategy σ for O in $\Gamma_f(L)$ induces a mapping

  $$f_\sigma \colon \Sigma_I^\omega \to \Sigma_O^\omega$$

- σ is winning $\Leftrightarrow \{\binom{\alpha}{f_\sigma(\alpha)} \mid \alpha \in \Sigma_I^\omega\} \subseteq L$ (f_σ uniformizes L)

Uniformization of Relations

- A strategy σ for O in $\Gamma_f(L)$ induces a mapping

  $$f_\sigma \colon \Sigma_I^\omega \to \Sigma_O^\omega$$

- σ is winning $\Leftrightarrow \{ \binom{\alpha}{f_\sigma(\alpha)} \mid \alpha \in \Sigma_I^\omega \} \subseteq L$ (f_σ uniformizes L)

Continuity in terms of strategies:

- Strategy without lookahead: i-th letter of $f_\sigma(\alpha)$ only depends on first i letters of α (very strong notion of continuity).

Uniformization of Relations

- A strategy σ for O in $\Gamma_f(L)$ induces a mapping

  $$f_\sigma \colon \Sigma_I^\omega \to \Sigma_O^\omega$$

- σ is winning $\Leftrightarrow \{\binom{\alpha}{f_\sigma(\alpha)} \mid \alpha \in \Sigma_I^\omega\} \subseteq L$ (f_σ uniformizes L)

Continuity in terms of strategies:

- Strategy without lookahead: i-th letter of $f_\sigma(\alpha)$ only depends on first i letters of α (very strong notion of continuity).
- Strategy with constant delay: f_σ Lipschitz-continuous.

Uniformization of Relations

- A strategy σ for O in $\Gamma_f(L)$ induces a mapping

  $$f_\sigma \colon \Sigma_I^\omega \to \Sigma_O^\omega$$

- σ is winning $\Leftrightarrow \{\binom{\alpha}{f_\sigma(\alpha)} \mid \alpha \in \Sigma_I^\omega\} \subseteq L$ (f_σ uniformizes L)

Continuity in terms of strategies:

- Strategy without lookahead: i-th letter of $f_\sigma(\alpha)$ only depends on first i letters of α (very strong notion of continuity).
- Strategy with constant delay: f_σ Lipschitz-continuous.
- Strategy with arbitrary (finite) delay: f_σ (uniformly) continuous.

Uniformization of Relations

- A strategy σ for O in $\Gamma_f(L)$ induces a mapping

  $$f_\sigma \colon \Sigma_I^\omega \to \Sigma_O^\omega$$

- σ is winning $\Leftrightarrow \{\binom{\alpha}{f_\sigma(\alpha)} \mid \alpha \in \Sigma_I^\omega\} \subseteq L$ (f_σ uniformizes L)

Continuity in terms of strategies:

- Strategy without lookahead: i-th letter of $f_\sigma(\alpha)$ only depends on first i letters of α (very strong notion of continuity).
- Strategy with constant delay: f_σ Lipschitz-continuous.
- Strategy with arbitrary (finite) delay: f_σ (uniformly) continuous.

Holtmann, Kaiser, Thomas: for ω-regular L

L uniformizable by continuous function

$\Leftrightarrow$

L uniformizable by Lipschitz-continuous function

Outline

1. **ω-regular Winning conditions**
2. Max-regular Winning Conditions
3. Determinacy
4. Conclusion

Our Results: Regular Winning Conditions

Theorem (Klein & Z. '15)

1. *TFAE for L given by deterministic parity automaton $\mathcal{A}$ with k colors:*
   - *O wins $\Gamma_f(L)$ for some f.*
   - *O wins $\Gamma_f(L)$ for some constant f with $f(0) \leq 2^{|\mathcal{A}| \cdot k}$.*
2. *Deciding whether this is the case is* ExpTime*-complete.*

Our Results: Regular Winning Conditions

Theorem (Klein & Z. '15)

1. *TFAE for L given by deterministic parity automaton $\mathcal{A}$ with k colors:*
   - *O wins $\Gamma_f(L)$ for some f.*
   - *O wins $\Gamma_f(L)$ for some constant f with $f(0) \leq 2^{|\mathcal{A}| \cdot k}$.*
2. *Deciding whether this is the case is* EXPTIME*-complete.*
3. *Matching lower bound on necessary lookahead (already for reachability and safety).*

Our Results: Regular Winning Conditions

Theorem (Klein & Z. '15)

1. *TFAE for L given by deterministic parity automaton $\mathcal{A}$ with k colors:*
   - *O wins $\Gamma_f(L)$ for some f.*
   - *O wins $\Gamma_f(L)$ for some constant f with $f(0) \leq 2^{|\mathcal{A}| \cdot k}$.*
2. *Deciding whether this is the case is* EXPTIME*-complete.*
3. *Matching lower bound on necessary lookahead (already for reachability and safety).*
4. *Solving reachability delay games is* PSPACE*-complete.*

Lower Bounds for Reachability Conditions

Theorem

For every $n > 1$ there is a language L_n such that

- *L_n is recognized by some deterministic reachability automaton $\mathcal{A}_n$ with $|\mathcal{A}_n| \in \mathcal{O}(n)$,*
- *O wins $\Gamma_f(L_n)$ for some constant delay function f, but*
- *I wins $\Gamma_f(L_n)$ for every delay function f with $f(0) \leq 2^n$.*

Lower Bounds for Reachability Conditions

Theorem

For every $n > 1$ there is a language L_n such that

- *L_n is recognized by some deterministic reachability automaton $\mathcal{A}_n$ with $|\mathcal{A}_n| \in \mathcal{O}(n)$,*
- *O wins $\Gamma_f(L_n)$ for some constant delay function f, but*
- *I wins $\Gamma_f(L_n)$ for every delay function f with $f(0) \leq 2^n$.*

Proof:

- $\Sigma_I = \Sigma_O = \{1, \ldots, n\}$.
- $w \in \Sigma_I^*$ contains bad j-pair ($j \in \Sigma_I$) if there are two occurrences of j in w such that no $j' > j$ occurs in between.

Lower Bounds for Reachability Conditions

Theorem

For every $n > 1$ there is a language L_n such that

- *L_n is recognized by some deterministic reachability automaton $\mathcal{A}_n$ with $|\mathcal{A}_n| \in \mathcal{O}(n)$,*
- *O wins $\Gamma_f(L_n)$ for some constant delay function f, but*
- *I wins $\Gamma_f(L_n)$ for every delay function f with $f(0) \leq 2^n$.*

Proof:

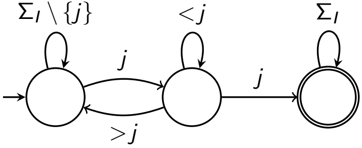
- $\Sigma_I = \Sigma_O = \{1, \ldots, n\}$.
- *$w \in \Sigma_I^*$ contains bad j-pair ($j \in \Sigma_I$) if there are two occurrences of j in w such that no $j' > j$ occurs in between.*
- *$w \in \Sigma_O^*$ has no bad j-pair for any $j \Rightarrow |w| \leq 2^n - 1$.*
- *Exists $w_n \in \Sigma_O^*$ with $|w_n| = 2^n - 1$ and without bad j-pair.*

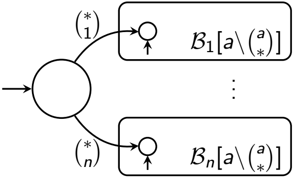
Lower Bounds for Reachability Conditions

$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.

Lower Bounds for Reachability Conditions

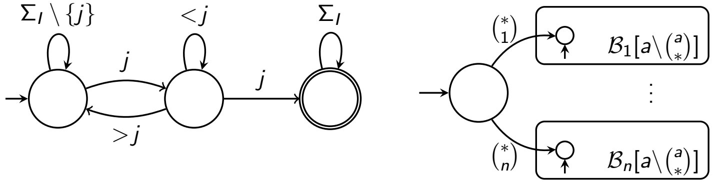
$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.





Lower Bounds for Reachability Conditions

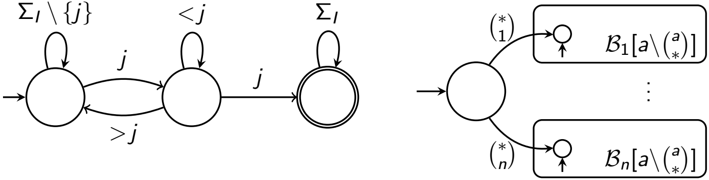
$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.



- O wins $\Gamma_f(L_n)$, if $f(0) > 2^n$: In first round, I picks u_0 s.t. u_0 without its first letter has bad j-pair. O picks j in first round.

Lower Bounds for Reachability Conditions

$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.



- O wins $\Gamma_f(L_n)$, if $f(0) > 2^n$: In first round, I picks u_0 s.t. u_0 without its first letter has bad j-pair. O picks j in first round.
- I wins $\Gamma_f(L_n)$, if $f(0) \leq 2^n$:
  - I picks prefix of $1w_n$ of length $f(0)$ in first round,
  - O answers by some j.
  - I finishes w_n and then picks some $j' \neq j$ ad infinitum.

Solving Delay Games

Theorem

TFAE for L recognized by a parity automaton with k colors:

1. *O wins $\Gamma_f(L)$ for some f.*
2. 0 *wins $\Gamma_f(L)$ for the constant f with $f(0) = 2^{(|\mathcal{A}|k)^2}$.*

Solving Delay Games

Theorem

TFAE for L recognized by a parity automaton with k colors:

1. *O wins $\Gamma_f(L)$ for some f.*
2. 0 *wins $\Gamma_f(L)$ for the constant f with $f(0) = 2^{(|\mathcal{A}|k)^2}$.*
3. *O wins parity game $\mathcal{G}$.*

Furthermore, $\mathcal{G}$ can be constructed and solved in exponential time.

Solving Delay Games

Theorem

TFAE for L recognized by a parity automaton with k colors:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *0 wins $\Gamma_f(L)$ for the constant f with $f(0) = 2^{(|\mathcal{A}|k)^2}$.*
3. *O wins parity game $\mathcal{G}$.*

Furthermore, $\mathcal{G}$ can be constructed and solved in exponential time.

Proof Idea:

Capture behavior of $\mathcal{A}$, i.e., state changes and maximal color seen on run $\Rightarrow$ equivalence relation $\equiv$ over Σ^* of exponential index.

Solving Delay Games

Theorem

TFAE for L recognized by a parity automaton with k colors:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for the constant f with $f(0) = 2^{(|\mathcal{A}|k)^2}$.*
3. *O wins parity game $\mathcal{G}$.*

Furthermore, $\mathcal{G}$ can be constructed and solved in exponential time.

Proof Idea:

Capture behavior of $\mathcal{A}$, i.e., state changes and maximal color seen on run $\Rightarrow$ equivalence relation $\equiv$ over Σ^* of exponential index.

Lemma

Let $(x_i)_{i \in \mathbb{N}}$ and $(x'_i)_{i \in \mathbb{N}}$ be two sequences of words over Σ^* with $x_i \equiv x'_i$ for all i. Then,

$$x_0 x_1 x_2 \cdots \in L(\mathcal{A}) \Leftrightarrow x'_0 x'_1 x'_2 \cdots \in L(\mathcal{A}).$$

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define parity game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define parity game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some constant f $\Leftrightarrow$ she wins $\mathcal{G}$.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define parity game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some constant f $\Leftrightarrow$ she wins $\mathcal{G}$.

$\mathcal{G}$ is delay-free parity game of exponential size with k colors.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define parity game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some constant f $\Leftrightarrow$ she wins $\mathcal{G}$.

$\mathcal{G}$ is delay-free parity game of exponential size with k colors.

Corollary

Winner can be determined in EXPTIME.

Further Results

Applying both directions of equivalence between $\Gamma_f(L(\mathcal{A}))$ and $\mathcal{G}$ yields upper bound on lookahead.

Corollary

Let $L = L(\mathcal{A})$ where $\mathcal{A}$ is a deterministic parity automaton with k colors. The following are equivalent:

1. *O wins $\Gamma_f(L)$ for some delay function f.*
2. *O wins $\Gamma_f(L)$ for some constant delay function f with $f(0) \leq 2^{(|\mathcal{A}|k)^2}$.*

Further Results

Applying both directions of equivalence between $\Gamma_f(L(\mathcal{A}))$ and $\mathcal{G}$ yields upper bound on lookahead.

Corollary

Let $L = L(\mathcal{A})$ where $\mathcal{A}$ is a deterministic parity automaton with k colors. The following are equivalent:

1. *O wins $\Gamma_f(L)$ for some delay function f.*
2. *O wins $\Gamma_f(L)$ for some constant delay function f with $f(0) \leq 2^{(|\mathcal{A}|k)^2}$.*

Note: $f(0) \leq 2^{2|\mathcal{A}|k+2} + 2$ achievable by direct pumping argument.

Outline

1. ω-regular Winning conditions
2. **Max-regular Winning Conditions**
3. Determinacy
4. Conclusion

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier to (weak) monadic second order logic (WMSO/MSO)

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier to (weak) monadic second order logic (WMSO/MSO)

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

  $L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier to (weak) monadic second order logic (WMSO/MSO)

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

  $L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$

L defined by

$$\forall x \exists y (y > x \wedge P_b(y)) \wedge$$
$$UX\ [\forall x \forall y \forall z (x < y < z \wedge x \in X \wedge z \in X \rightarrow y \in X)$$
$$\wedge\, \forall x (x \in X \rightarrow P_a(x))\ ]$$

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier to (weak) monadic second order logic (WMSO/MSO)

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

  $L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$

L defined by

$$\forall x \exists y (y > x \wedge P_b(y)) \wedge$$
$$UX\ [\forall x \forall y \forall z (x < y < z \wedge x \in X \wedge z \in X \rightarrow y \in X)$$
$$\wedge \forall x (x \in X \rightarrow P_a(x))\ ]$$

Theorem (Bojańczyk '14)

Delay-free games with WMSO+U winning conditions are decidable.

Max-Automata

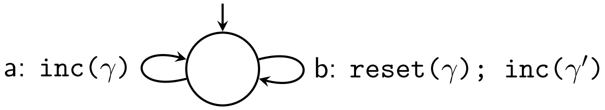
Equivalent automaton model for WMSO+U on infinite words:

- Deterministic finite automata with counters.
- counter actions: `incr`, `reset`, `max`.
- acceptance: boolean combination of "counter γ is bounded".

Max-Automata

Equivalent automaton model for WMSO+U on infinite words:

- Deterministic finite automata with counters.
- counter actions: `incr`, `reset`, `max`.
- acceptance: boolean combination of "counter γ is bounded".

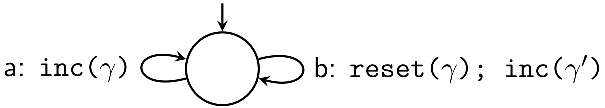


Acceptance condition: γ and γ' unbounded.

Max-Automata

Equivalent automaton model for WMSO+U on infinite words:

- Deterministic finite automata with counters.
- counter actions: `incr`, `reset`, `max`.
- acceptance: boolean combination of "counter γ is bounded".



Acceptance condition: γ and γ' unbounded.

Theorem (Bojańczyk '09)

The following are (effectively) equivalent:

1. *L WMSO+U-definable.*
2. *L recognized by max-automaton.*

The Case of Bounded Lookahead

Theorem (Z. '15)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

The Case of Bounded Lookahead

Theorem (Z. '15)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

Proof Idea:

Analogously to the parity case: capture behavior of $\mathcal{A}$, i.e., state changes and evolution of counter values:

- Transfers from counter γ to γ'.
- Existence of increments, but not how many.

$\Rightarrow$ equivalence relation $\equiv$ over Σ^* of exponential index.

The Case of Bounded Lookahead

Theorem (Z. '15)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

Proof Idea:

Analogously to the parity case: capture behavior of $\mathcal{A}$, i.e., state changes and evolution of counter values:

- Transfers from counter γ to γ'.
- Existence of increments, but not how many.

$\Rightarrow$ equivalence relation $\equiv$ over Σ^* of exponential index.

Lemma

Let $(x_i)_{i\in\mathbb{N}}$ and $(x_i')_{i\in\mathbb{N}}$ be two sequences of words over Σ^* with $\sup_i |x_i| < \infty$, $\sup_i |x_i'| < \infty$, and $x_i \equiv x_i'$ for all i. Then,

$$x_0x_1x_2\cdots \in L(\mathcal{A}) \Leftrightarrow x_0'x_1'x_2'\cdots \in L(\mathcal{A}).$$

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some constant f $\Leftrightarrow$ she wins $\mathcal{G}$.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define game $\mathcal{G}$:
  - I picks equivalence classes,
  - O constructs run on representatives (always one step behind to account for delay).
  - O wins, if run is accepting.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some constant $f \Leftrightarrow$ she wins $\mathcal{G}$.

$\mathcal{G}$ is delay-free with WMSO+U winning condition.

- Can be solved effectively by reduction to satisfiability problem for WMSO+U with path quantifiers over infinite trees.
- Doubly-exponential upper bound on necessary constant lookahead.

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

I: $\#$ 0 0 $\cdots$ 0

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

I:	$\#$	0	0	$\cdots$	0
O:	0				

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

$$\begin{array}{rcccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 \\ O: & 0 & & & & & \end{array}$$

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

$$\begin{array}{rcccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 \\ O: & 0 & * & & & & \end{array}$$

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

$$\begin{array}{rcccccccc} I\colon & \# & 0 & 0 & \cdots & 0 & 1 & 1 \\ O\colon & 0 & * & & & & & \end{array}$$

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

I:	$\#$	0	0	$\cdots$	0	1	1
O:	0	$*$	$*$				

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

$$\begin{array}{rcccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1 \\ O: & 0 & * & * & & & & & \end{array}$$

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

I:	$\#$	0	0	$\cdots$	0	1	1	1	$\cdots$
O:	0	$*$	$*$	$\cdots$					

Constant Lookahead is not Sufficient

Recall: O wins $\Gamma_f(L_0)$ for every unbounded f.

- Input block: $\#w$ with $w \in \{0,1\}^+$.
- Output block: $\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)}$
- Winning condition L_0: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Claim: I wins $\Gamma_f(L_0)$ for every constant f.

$$\begin{array}{rcccccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1 & \cdots \\ O: & 0 & * & * & \cdots & & & & & \end{array}$$

- Lookahead contains only input blocks of length $f(0)$.
- I can react to O's declaration at beginning of an output block to bound size of output blocks while producing arbitrarily large input blocks.

Any Unbounded Lookahead is Sufficient

Theorem

TFAE for L recognized by a max automaton with k counters:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for every unbounded f with $f(0) \geq 2^{2^{(|\mathcal{A}|k)^2}}$.*

Any Unbounded Lookahead is Sufficient

Theorem

TFAE for L recognized by a max automaton with k counters:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for every unbounded f with $f(0) \geq 2^{2^{(|\mathcal{A}|k)^2}}$.*
3. *O wins $\mathcal{G}$.*

Any Unbounded Lookahead is Sufficient

Theorem

TFAE for L recognized by a max automaton with k counters:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for every unbounded f with $f(0) \geq 2^{2^{(|\mathcal{A}|k)^2}}$.*
3. *O wins $\mathcal{G}$.*

Analogously to the case: of bounded lookahead:

- Define $\equiv_m$ as $\equiv$, but capture behavior up to m increments.
- I picks $\equiv_m$ classes for m tending to infinity.

Any Unbounded Lookahead is Sufficient

Theorem

TFAE for L recognized by a max automaton with k counters:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for every unbounded f with $f(0) \geq 2^{2^{(|\mathcal{A}|k)^2}}$.*
3. *O wins $\mathcal{G}$.*

Analogously to the case: of bounded lookahead:

- Define $\equiv_m$ as $\equiv$, but capture behavior up to m increments.
- I picks $\equiv_m$ classes for m tending to infinity.

$\mathcal{G}$ is infinite state $\Rightarrow$ cannot solve it to determine winner of delay game w.r.t. unbounded delay functions.

Outline

1. ω-regular Winning conditions
2. Max-regular Winning Conditions
3. **Determinacy**
4. Conclusion

Borel Determinacy for Delay Games

- A game is determined, if one of the players has a winning strategy.
- Borel hierarchy: family of languages constructed from *open* languages $K \cdot \Sigma^\omega$ with $K \subseteq \Sigma^*$ via countable union and complementation.
- Contains all regular and max-regular languages (and much more).

Borel Determinacy for Delay Games

- A game is determined, if one of the players has a winning strategy.
- Borel hierarchy: family of languages constructed from *open* languages $K \cdot \Sigma^\omega$ with $K \subseteq \Sigma^*$ via countable union and complementation.
- Contains all regular and max-regular languages (and much more).

Theorem (Martin '75)

Every delay-free game with Borel winning condition is determined.

Borel Determinacy for Delay Games

- A game is determined, if one of the players has a winning strategy.
- Borel hierarchy: family of languages constructed from *open* languages $K \cdot \Sigma^\omega$ with $K \subseteq \Sigma^*$ via countable union and complementation.
- Contains all regular and max-regular languages (and much more).

Theorem (Martin '75)

Every delay-free game with Borel winning condition is determined.

Theorem (Klein & Z. '15)

Every delay game with Borel winning condition is determined.

Outline

1. ω-regular Winning conditions
2. Max-regular Winning Conditions
3. Determinacy
4. **Conclusion**

Conclusion

Results:

- Tight results for ω-regular conditions
- First results for max-regular conditions, but decidability and exact complexity open.
- Borel determinacy.

Conclusion

Results:

- Tight results for ω-regular conditions
- First results for max-regular conditions, but decidability and exact complexity open.
- Borel determinacy.

Open problems:

- Results for other acceptance conditions (Rabin, Streett Muller), non-deterministic or alternating automata.
- Decidability of max-regular delay games w.r.t. unbounded delay functions.
- What are strategies in delay games, e.g., do they have to depend on the delay function under consideration?