Delay Games with WMSO+U Winning Conditions

Martin Zimmermann

Saarland University

July 13th, 2015

CSR 2015, Listvyanka, Russia

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	$\cdots$
O:	a	a	$\cdots$	

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

I:	b	a	b	b	a	b
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.

Outline

1. **WMSO with the Unbounding Quantifier**
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. 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

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

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)

Games with WMSO+U winning conditions are decidable.

Max-Automata

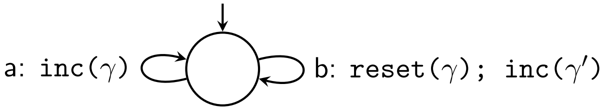
Equivalent automaton model for WMSO+U on infinite words:

- 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".

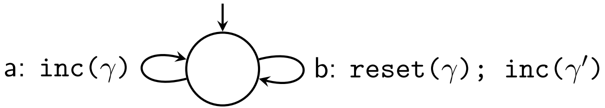


Acceptance condition: γ and γ' unbounded.

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.*

Outline

1. WMSO with the Unbounding Quantifier
2. **Delay Games**
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. Conclusion

Delay Games

The delay game $\Gamma_f(L)$:

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

Delay Games

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 $\beta = v_0 v_1 \cdots$).

Delay Games

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 $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Delay Games

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 unbounded, if $f(i) > 1$ for infinitely many i.

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.

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.

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

For ω-regular L (given by deterministic parity automaton):

Theorem (Hosch & Landweber '72)

"Given L, does O win $\Gamma_f(L)$ for some constant f?" is decidable.

Previous Results

For ω-regular L (given by deterministic parity automaton):

Theorem (Hosch & Landweber '72)

"Given L, does O win $\Gamma_f(L)$ for some constant f?" is decidable.

Theorem (Holtmann, Kaiser & Thomas '10)

1. *O wins $\Gamma_f(L)$ for some $f \Leftrightarrow O$ wins $\Gamma_f(L)$ for some constant f.*
2. *Decision problem in 2EXPTIME.*
3. *Doubly-exponential upper bound on necessary (constant) lookahead.*

Previous Results

For ω-regular L (given by deterministic parity automaton):

Theorem (Hosch & Landweber '72)

"Given L, does O win $\Gamma_f(L)$ for some constant f?" is decidable.

Theorem (Holtmann, Kaiser & Thomas '10)

1. *O wins $\Gamma_f(L)$ for some $f \Leftrightarrow O$ wins $\Gamma_f(L)$ for some constant f.*
2. *Decision problem in 2ExpTime.*
3. *Doubly-exponential upper bound on necessary (constant) lookahead.*

Theorem (Klein, Z. '15)

1. *Decision problem ExpTime-complete.*
2. *Tight exponential bounds on necessary (constant) lookahead.*

Previous Results

For ω-context-free L (given by ω-pushdown automaton):

Theorem (Fridman, Löding & Z. '11)

1. *Decision problem is undecidable.*
2. *Constant lookahead not enough: lookahead has to grow non-elementarily.*

Both results hold already for one-counter, visibly, weak, and deterministic context-free winning conditions.

Previous Results

For ω-context-free L (given by ω-pushdown automaton):

Theorem (Fridman, Löding & Z. '11)

1. *Decision problem is undecidable.*
2. *Constant lookahead not enough: lookahead has to grow non-elementarily.*

Both results hold already for one-counter, visibly, weak, and deterministic context-free winning conditions.

Theorem (Klein, Z. '15)

Delay games with Borel winning conditions are determined.

Outline

1. WMSO with the Unbounding Quantifier

2. Delay Games

3. WMSO+U Delay Games w.r.t. Constant Lookahead

4. Constant Lookahead is not Sufficient

5. Conclusion

Determinacy

Theorem

Delay Games with WMSO+U winning conditions w.r.t fixed delay functions are determined.

Determinacy

Theorem

Delay Games with WMSO+U winning conditions w.r.t fixed delay functions are determined.

Proof idea:

- Winning condition recognized by some automaton $\mathcal{A}$.
- Encode game as parity game in countable arena. States store:
  - Current lookahead (queue over Σ_I)
  - state $\mathcal{A}$ reaches on current play prefix.
  - Current counter values after this run prefix.
  - Maximal counter values seen thus far.
  - Flag marking whether maximum was increased during last transition.
- Thus: counter γ unbounded if, and only if, corresponding flag is raised infinitely often $\Rightarrow$ parity condition.

Capturing Finite Runs of Max-Automata

Theorem

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.

Capturing Finite Runs of Max-Automata

Theorem

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:

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.

Capturing Finite Runs of Max-Automata

Theorem

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:

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 abstract game $\mathcal{G}(\mathcal{A})$:
  - 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 abstract game $\mathcal{G}(\mathcal{A})$:
  - 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 $\mathcal{G}(\mathcal{A})$.

Removing Delay

- In $\mathcal{A}$, project away Σ_O and construct equivalence $\equiv$ over Σ_I^*.
- Define abstract game $\mathcal{G}(\mathcal{A})$:
  - 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 $\mathcal{G}(\mathcal{A})$.

$\mathcal{G}(\mathcal{A})$ 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.

Outline

1. WMSO with the Unbounding Quantifier
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. **Constant Lookahead is not Sufficient**
5. Conclusion

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.

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

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

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.

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.

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.

I:	$\#$	0	0	$\cdots$	0	1	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.

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

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.

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

- 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.

Outline

1. WMSO with the Unbounding Quantifier
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. **Conclusion**

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- Doubly-exponential upper bound on necessary constant lookahead.
- But constant delay is not always sufficient.

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- Doubly-exponential upper bound on necessary constant lookahead.
- But constant delay is not always sufficient.

Current work:

- Tight bounds on necessary lookahead for the case of constant delay functions.
- Solve games w.r.t. arbitrary delay functions.

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- Doubly-exponential upper bound on necessary constant lookahead.
- But constant delay is not always sufficient.

Current work:

- Tight bounds on necessary lookahead for the case of constant delay functions.
- Solve games w.r.t. arbitrary delay functions.

Conjecture

The following are equivalent for L definable in WMSO+U:

1. *O wins $\Gamma_f(L)$ for some f.*
2. *O wins $\Gamma_f(L)$ for every unbounded f s.t. $f(0)$ is "large enough".*