How Much Lookahead is Needed to Win Infinite Games?

Joint work with Felix Klein (Saarland University)

Martin Zimmermann

Saarland University

February 13th, 2015
Workshop Automata, Concurrency and Timed Systems
Chennai Mathematical Institute, Chennai, India

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.

- Interaction between players typically described by a graph.
- Simpler setting: realizability / Gale-Stewart games.
  Players I/O alternatingly pick letters $\alpha(i)$ and $\beta(i)$. O wins if $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots$ is in winning condition L.

Introduction

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

- Interaction between players typically described by a graph.
- Simpler setting: realizability / Gale-Stewart games. Players I/O alternatingly pick letters $\alpha(i)$ and $\beta(i)$. O wins if $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots$ is in winning condition L.

But assuming fixed interaction might be too strong in the presence of buffers, asynchronous communication channels, etc.

Introduction

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

- Interaction between players typically described by a graph.
- Simpler setting: realizability / Gale-Stewart games. Players I/O alternatingly pick letters $\alpha(i)$ and $\beta(i)$. O wins if $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots$ is in winning condition L.

But assuming fixed interaction might be too strong in the presence of buffers, asynchronous communication channels, etc.

- Hosch & Landweber ('72), Holtmann, Kaiser & Thomas ('10): allow one player to delay her moves, thereby gain a lookahead on her opponents moves.

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 $i > 0$.

The Delay Game $\Gamma_f(L)$

- Delay function: $f : \mathbb{N} \to \mathbb{N}_+$.
- 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 $i > 0$.

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?

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

  I: b

  No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

  I: b
  O: a

  No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a
O:	a	

No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a, b\} \times \{a, b\})^\omega$, if $\beta(i) = \alpha(i + 2)$.

I:	b	a
O:	a	a

No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a	b
O:	a	a	

  No delay

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a	b
O:	a	a	

No delay: I wins

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a	b
O:	a	a	

  No delay: I wins $\qquad f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\}\times\{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a, b\} \times \{a, b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad$ $f(0) = 3$, $f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad f(0) = 3$, $f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a, b\} \times \{a, b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\quad$ $f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad f(0) = 3,\ f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a, b\} \times \{a, b\})^\omega$, if $\beta(i) = \alpha(i + 2)$.

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

No delay: I wins $\qquad f(0) = 3,\ f(i + 1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a	b					
O:	a	a						

No delay: I wins

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

$f(0) = 3$, $f(i+1) = 1$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

I:	b	a	b					
O:	a	a						

No delay: I wins

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

$f(0) = 3$, $f(i+1) = 1$: O wins

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad$ $f(0) = 3$, $f(i+1) = 1$: O wins

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_2 \subseteq (\{a,b,c\} \times \{a,b,c\})^\omega$, if
  - $\alpha(i) = a$ for every i, or
  - $\beta(0) = \alpha(i)$, where i is minimal with $\alpha(i) \neq a$.

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\}\times\{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $f(0) = 3$, $f(i+1) = 1$: O wins

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_2 \subseteq (\{a,b,c\}\times\{a,b,c\})^\omega$, if
  - $\alpha(i) = a$ for every i, or
  - $\beta(0) = \alpha(i)$, where i is minimal with $\alpha(i) \neq a$.

I: $\overbrace{a \cdots a}^{f(0)}$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\}\times\{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\qquad$ $f(0) = 3$, $f(i+1) = 1$: O wins

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_2 \subseteq (\{a,b,c\}\times\{a,b,c\})^\omega$, if
  - $\alpha(i) = a$ for every i, or
  - $\beta(0) = \alpha(i)$, where i is minimal with $\alpha(i) \neq a$.

$$I: \overbrace{a \cdots a}^{f(0)}$$
$$O: b$$

Examples

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_1 \subseteq (\{a,b\} \times \{a,b\})^\omega$, if $\beta(i) = \alpha(i+2)$.

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

No delay: I wins $\quad$ $f(0) = 3$, $f(i+1) = 1$: O wins

- $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_2 \subseteq (\{a,b,c\} \times \{a,b,c\})^\omega$, if
  - $\alpha(i) = a$ for every i, or
  - $\beta(0) = \alpha(i)$, where i is minimal with $\alpha(i) \neq a$.

$$
\begin{array}{lll}
I: & \overbrace{a \cdots a}^{f(0)} & c \\
O: & b &
\end{array}
$$

I wins for every 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?

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, and 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

Open Questions

- No known (non-trivial) lower bounds on computational complexity and necessary lookahead.
- No results for subclasses of ω-regular conditions.

We consider two subclasses:

Open Questions

- No known (non-trivial) lower bounds on computational complexity and necessary lookahead.
- No results for subclasses of ω-regular conditions.

We consider two subclasses:

Fix $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$

- Reachability acceptance:

$$L_\exists(\mathcal{A}) = \{w \in \Sigma^\omega \mid \mathcal{A} \text{ has run on } w \text{ that visits } F\}$$

- Safety acceptance:

$$L_\forall(\mathcal{A}) = \{w \in \Sigma^\omega \mid \mathcal{A} \text{ has run on } w \text{ that never visits } V \setminus F\}$$

Outline

1. **Lower Bounds on Lookahead**
2. Complexity: Reachability Conditions
3. Complexity: Safety Conditions
4. Complexity: ω-regular Conditions
5. Beyond ω-regularity: WMSO+U conditions
6. Conclusion

Lower Bounds for Reachability Conditions

Theorem

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

- $L_n = L_\exists(\mathcal{A}_n)$ *for 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

- $L_n = L_\exists(\mathcal{A}_n)$ *for 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

- $L_n = L_\exists(\mathcal{A}_n)$ *for 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:

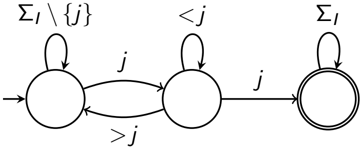
- $\Sigma_I = \Sigma_O = \{1, \ldots, n\}$.
- $w \in \Sigma_I^*$ *contains* *bad j-pair* ($j \in \Sigma_I$) *if there are two occurrences of* j *in* w *such that no* $j' > j$ *occurs in between.*
- $w \in \Sigma_O^*$ *has no bad* j*-pair for any* $j \Rightarrow |w| \leq 2^n - 1$.
- *Exists* $w_n \in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

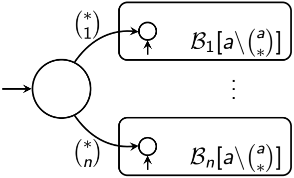
Lower Bounds for Reachability Conditions

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

Lower Bounds for Reachability Conditions

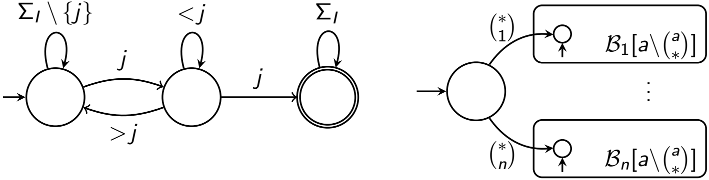
$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.





Lower Bounds for Reachability Conditions

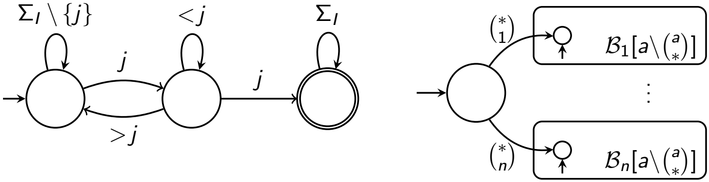
$\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\cdots \in L_n$ iff $\alpha(1)\alpha(2)\cdots$ contains a bad $\beta(0)$-pair.



- 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.

Remarks

- The automata are deterministic.
- Similar construction works for safety, too.

Remarks

- The automata are deterministic.
- Similar construction works for safety, too.
- Alphabet size grows in n.
  - Constant-size alphabets possible using binary encoding.
  - Requires automata of size $(n \log n)$.

Open question: constant-size alphabet and automata of size $\mathcal{O}(n)$ simultaneously achievable.

Outline

A Sufficient Condition

O wins $\Gamma_f(L)$ for some $f \Rightarrow$ projection $\mathrm{pr}_0(L)$ to Σ_I universal.

A Sufficient Condition

O wins $\Gamma_f(L)$ for some $f \Rightarrow$ projection $\mathrm{pr}_0(L)$ to Σ_I universal.

Theorem

Let $L = L_\exists(\mathcal{A})$, where $\mathcal{A}$ is a non-deterministic reachability automaton. 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 $f(0) \leq 2^{|\mathcal{A}|}$.*
3. *$\mathrm{pr}_0(L)$ is universal.*

A Sufficient Condition

O wins $\Gamma_f(L)$ for some $f \Rightarrow$ projection $\mathrm{pr}_0(L)$ to Σ_I universal.

Theorem

Let $L = L_\exists(\mathcal{A})$, where $\mathcal{A}$ is a non-deterministic reachability automaton. 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 $f(0) \leq 2^{|\mathcal{A}|}$.*
3. *$\mathrm{pr}_0(L)$ is universal.*

Corollary

The following problem is PSPACE-complete: Given a non-deterministic reachability automaton $\mathcal{A}$, does O win $\Gamma_f(L_\exists(\mathcal{A}))$ for some f?

Outline

Hardness of Safety Conditions

Theorem

The following problem is $\textsc{ExpTime}$*-hard: Given a deterministic safety automaton* $\mathcal{A}$*, does* O *win* $\Gamma_f(L_\forall(\mathcal{A}))$ *for some* f*?*

Hardness of Safety Conditions

Theorem

The following problem is EXPTIME*-hard: Given a deterministic safety automaton $\mathcal{A}$, does O win $\Gamma_f(L_\forall(\mathcal{A}))$ for some f?*

Proof:

By a reduction from alternating polynomial space Turing machines.

- I produces configurations, picks existential transitions:
  - has to start with initial configuration, and
  - either copies the current configuration
  - or gives a new one.
- O checks copies for correctness, picks universal transitions.

Hardness of Safety Conditions

I:

O:

Hardness of Safety Conditions

I: | N | c_0 $\exists$ |

O:

Hardness of Safety Conditions

I: | N | c_0 $\exists$ | | N | c_1 $\forall$ |

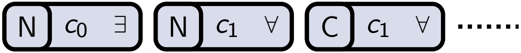
O:

Hardness of Safety Conditions

I: | N | c_0 | $\exists$ | | N | c_1 | $\forall$ | | C | c_1 | $\forall$ |

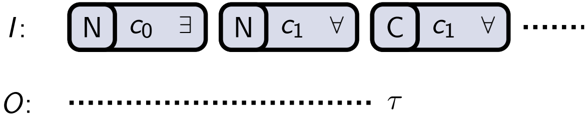
O:

Hardness of Safety Conditions

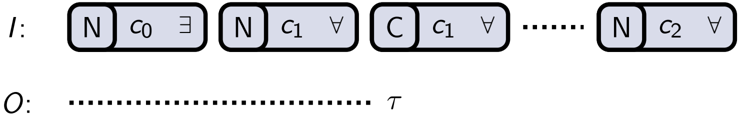
I: 

O:

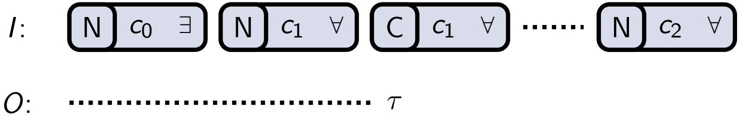
Hardness of Safety Conditions



Hardness of Safety Conditions



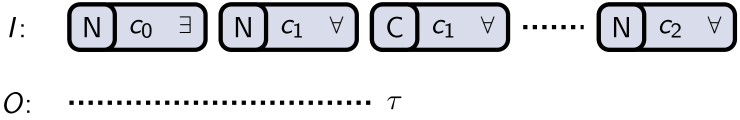
Hardness of Safety Conditions



To prevent I from cheating, O can claim errors:

- an incorrect copy by marking the position in the original.
- an incorrect update by marking the position in the original.

Hardness of Safety Conditions



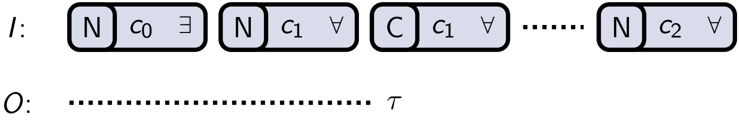
To prevent I from cheating, O can claim errors:

- an incorrect copy by marking the position in the original.
- an incorrect update by marking the position in the original.

Winning condition checks:

- I always picks configurations of length $p(n)$.
- c_0 is initial configuration on w.
- The first error claimed by O is not an actual error.
- Some c_i is accepting.

Hardness of Safety Conditions



To prevent I from cheating, O can claim errors:

- an incorrect copy by marking the position in the original.
- an incorrect update by marking the position in the original.

Winning condition checks:

- I always picks configurations of length $p(n)$.
- c_0 is initial configuration on w.
- The first error claimed by O is not an actual error.
- Some c_i is accepting.

If this is the case, play is not accepted, i.e., I wins.

Outline

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in ExpTime*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in ExpTime*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof Idea:

- Define abstract game $\mathcal{G}(\mathcal{A})$:
  - Define equivalence relation on Σ_I^*: $x \equiv x'$, if x and x' induce the same behavior on projection of $\mathcal{A}$ to Σ_I.
  - In $\mathcal{G}(\mathcal{A})$, Player I picks $\equiv$-equivalence classes, Player O constructs a run of $\mathcal{A}$ on representatives of the picked classes (one move delay).

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in ExpTime*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof Idea:

- Define abstract game $\mathcal{G}(\mathcal{A})$:
  - Define equivalence relation on Σ_I^*: $x \equiv x'$, if x and x' induce the same behavior on projection of $\mathcal{A}$ to Σ_I.
  - In $\mathcal{G}(\mathcal{A})$, Player I picks $\equiv$-equivalence classes, Player O constructs a run of $\mathcal{A}$ on representatives of the picked classes (one move delay).
- $\mathcal{G}(\mathcal{A})$ can be encoded as parity game of exponential size with the same colors as $\mathcal{A}$.
- Such a game can be solved in exponential time in $|\mathcal{A}|$.

Upper Bounds for ω-regular Conditions

Equivalence classes have "short" representatives, as they are recognized by "small" automata.

Upper Bounds for ω-regular Conditions

Equivalence classes have "short" representatives, as they are recognized by "small" automata.

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 $f(0) \leq 2^{(|\mathcal{A}|k)^2+1}$.*

Upper Bounds for ω-regular Conditions

Equivalence classes have "short" representatives, as they are recognized by "small" automata.

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 $f(0) \leq 2^{(|\mathcal{A}|k)^2+1}$.*

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

Outline

1. Lower Bounds on Lookahead
2. Complexity: Reachability Conditions
3. Complexity: Safety Conditions
4. Complexity: ω-regular Conditions
5. **Beyond ω-regularity: WMSO+U conditions**
6. Conclusion

Delay Games with WMSO+U conditions

Two equivalent definitions:

1. **WMSO+U**: weak monadic second-order logic with the unbounding quantifier U. $UX\varphi(X)$: there are arbitrarily large finite sets X s.t. $\varphi(X)$ holds.

Delay Games with WMSO+U conditions

Two equivalent definitions:

1. **WMSO+U**: weak monadic second-order logic with the unbounding quantifier U. $UX\varphi(X)$: there are arbitrarily large finite sets X s.t. $\varphi(X)$ holds.
2. **Max-automata** Deterministic finite automata with counters; actions: incr, reset, max. Acceptance: boolean combination of "counter γ is bounded".

Delay Games with WMSO+U conditions

Two equivalent definitions:

1. **WMSO+U**: weak monadic second-order logic with the unbounding quantifier U. $UX\varphi(X)$: there are arbitrarily large finite sets X s.t. $\varphi(X)$ holds.
2. **Max-automata** Deterministic finite automata with counters; actions: incr, reset, max. Acceptance: boolean combination of "counter γ is bounded".

Example:

$$L = \{\alpha \in \{a, b, c\}^\omega \mid a^n b \text{ infix of } \alpha \text{ for every } n\}$$

Delay Games with WMSO+U conditions

Two equivalent definitions:

1. **WMSO+U**: weak monadic second-order logic with the unbounding quantifier U. $UX\varphi(X)$: there are arbitrarily large finite sets X s.t. $\varphi(X)$ holds.
2. **Max-automata** Deterministic finite automata with counters; actions: incr, reset, max. Acceptance: boolean combination of "counter γ is bounded".

Example:

$$L = \{\alpha \in \{a, b, c\}^\omega \mid a^n b \text{ infix of } \alpha \text{ for every } n\}$$

Theorem

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

Delay Games with WMSO+U conditions

Two equivalent definitions:

1. **WMSO+U**: weak monadic second-order logic with the unbounding quantifier U. $UX\varphi(X)$: there are arbitrarily large finite sets X s.t. $\varphi(X)$ holds.
2. **Max-automata** Deterministic finite automata with counters; actions: incr, reset, max. Acceptance: boolean combination of "counter γ is bounded".

Example:

$$L = \{\alpha \in \{a, b, c\}^\omega \mid a^n b \text{ infix of } \alpha \text{ for every } n\}$$

Theorem

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

But constant delay is not always sufficient.

Outline

Conclusion

Results:

automaton	lookahead	complexity
(non)det. reachability	exponential*	PSPACE-complete

Conclusion

Results:

automaton	lookahead	complexity
(non)det. reachability	exponential*	PSPACE-complete
det. safety	exponential*	EXPTIME-complete
det. parity	exponential*	EXPTIME-complete

Conclusion

Results:

automaton	lookahead	complexity
(non)det. reachability	exponential*	PSPACE-complete
det. safety	exponential*	EXPTIME-complete
det. parity	exponential*	EXPTIME-complete
safety $\cap$ det. reach.	polynomial	Π_2^{P}

*: tight bound.

Conclusion

Results:

automaton	lookahead	complexity
(non)det. reachability	exponential*	PSPACE-complete
det. safety	exponential*	EXPTIME-complete
det. parity	exponential*	EXPTIME-complete
safety $\cap$ det. reach.	polynomial	Π_2^{P}

*: tight bound.

Open questions:

- Consider non-deterministic automata and
- Rabin, Streett, Muller automata.
- Can we determine minimal lookahead that is sufficient to win?
- Weak MSO+U w.r.t. arbitrary delay functions.

Outline

7. Backup Slides

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in $\textsc{ExpTime}$*: Given a deterministic automaton* $\mathcal{A}$*, does* O *win* $\Gamma_f(L(\mathcal{A}))$ *for some* f*?*

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in EXPTIME*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof:

- Extend $\mathcal{A}$ to $\mathcal{C}$ to keep track of maximal color seen during run using states of the form (q, c).
- Note: $L(\mathcal{C}) \neq L(\mathcal{A})$.

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in EXPTIME*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof:

- Extend $\mathcal{A}$ to $\mathcal{C}$ to keep track of maximal color seen during run using states of the form (q, c).
- Note: $L(\mathcal{C}) \neq L(\mathcal{A})$.

I: $\alpha(0)$ ·················· $\alpha(i)$ ···························· $\alpha(j)$

O: $\beta(0)$ ·················· $\beta(i)$

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in EXPTIME*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof:

- Extend $\mathcal{A}$ to $\mathcal{C}$ to keep track of maximal color seen during run using states of the form (q, c).
- Note: $L(\mathcal{C}) \neq L(\mathcal{A})$.

I: $\alpha(0)$ ⋯⋯ $\alpha(i)$ ⋯⋯ $\alpha(j)$

q_0 ⋯ q

O: $\beta(0)$ ⋯⋯ $\beta(i)$

- q: state reached by $\mathcal{A}$ after processing $\binom{\alpha(0)}{\beta(0)} \cdots \binom{\alpha(i)}{\beta(i)}$.

Upper Bounds for ω-regular Conditions

Theorem

The following problem is in EXPTIME*: Given a deterministic automaton $\mathcal{A}$, does O win $\Gamma_f(L(\mathcal{A}))$ for some f?*

Proof:

- Extend $\mathcal{A}$ to $\mathcal{C}$ to keep track of maximal color seen during run using states of the form (q, c).
- Note: $L(\mathcal{C}) \neq L(\mathcal{A})$.

I:	$\alpha(0)$ ········	$\alpha(i)$ ········	$\alpha(j)$
	q_0	q	P
O:	$\beta(0)$ ········	$\beta(i)$	

- q: state reached by $\mathcal{A}$ after processing $\binom{\alpha(0)}{\beta(0)} \cdots \binom{\alpha(i)}{\beta(i)}$.
- P: set of states reachable by $\mathrm{pr}_0(\mathcal{C})$ from $(q, \Omega(q))$ after processing $\alpha(i+1) \cdots \alpha(j)$.

Proof Continued

- $\delta_{\mathcal{P}}$: transition function of powerset automaton of $\mathrm{pr}_0(\mathcal{C})$.

Proof Continued

- $\delta_{\mathcal{P}}$: transition function of powerset automaton of $\mathrm{pr}_0(\mathcal{C})$.
- Let $w \in \Sigma_I^*$: define $r_w^D \colon D \to 2^{Q_{\mathcal{C}}}$ via

$$r_w^D(q, c) = \delta_{\mathcal{P}}^*(\,\{\,(q, \Omega(q))\,\}\,, w)$$

Proof Continued

- $\delta_{\mathcal{P}}$: transition function of powerset automaton of $\mathrm{pr}_0(\mathcal{C})$.
- Let $w \in \Sigma_I^*$: define $r_w^D \colon D \to 2^{Q_\mathcal{C}}$ via

$$r_w^D(q,c) = \delta_{\mathcal{P}}^*(\,\{\,(q,\Omega(q))\,\}\,,w)$$

- w is witness for $r_w^D \Rightarrow$ Language W_r of witnesses.
- $\mathfrak{R} = \{r \mid W_r \text{ infinite}\}$.

Proof Continued

- $\delta_{\mathcal{P}}$: transition function of powerset automaton of $\mathrm{pr}_0(\mathcal{C})$.
- Let $w \in \Sigma_I^*$: define $r_w^D \colon D \to 2^{Q_{\mathcal{C}}}$ via

$$r_w^D(q, c) = \delta_{\mathcal{P}}^*(\{(q, \Omega(q))\}, w)$$

- w is witness for $r_w^D \Rightarrow$ Language W_r of witnesses.
- $\mathfrak{R} = \{r \mid W_r \text{ infinite}\}$.

Lemma

Fix domain D. If $|w| \geq 2^{|\mathcal{C}|^2}$, then w is witness of a unique $r \in \mathfrak{R}$ with domain D.

The Game $\mathcal{G}(\mathcal{A})$

Define new game $\mathcal{G}(\mathcal{A})$ between I and O:

- In round 0:
  - I has to pick $r_0 \in \mathfrak{R}$ with $\mathrm{dom}(r_0) = \{q_I^{\mathcal{C}}\}$,
  - O has to pick $q_0 \in \mathrm{dom}(r_0)$ (i.e., $q_0 = q_I^{\mathcal{C}}$).

The Game $\mathcal{G}(\mathcal{A})$

Define new game $\mathcal{G}(\mathcal{A})$ between I and O:

- In round 0:
  - I has to pick $r_0 \in \mathfrak{R}$ with $\mathrm{dom}(r_0) = \{q_I^{\mathcal{C}}\}$,
  - O has to pick $q_0 \in \mathrm{dom}(r_0)$ (i.e., $q_0 = q_I^{\mathcal{C}}$).
- Round $i > 0$ with play prefix $r_0 q_0 \cdots r_{i-1} q_{i-1}$:
  - I has to pick $r_i \in \mathfrak{R}$ with $\mathrm{dom}(r_i) = r_{i-1}(q_{i-1})$,
  - O has to pick $q_i \in \mathrm{dom}(r_i)$.

The Game $\mathcal{G}(\mathcal{A})$

Define new game $\mathcal{G}(\mathcal{A})$ between I and O:

- In round 0:
  - I has to pick $r_0 \in \mathfrak{R}$ with $\mathrm{dom}(r_0) = \{q_I^{\mathcal{C}}\}$,
  - O has to pick $q_0 \in \mathrm{dom}(r_0)$ (i.e., $q_0 = q_I^{\mathcal{C}}$).
- Round $i > 0$ with play prefix $r_0 q_0 \cdots r_{i-1} q_{i-1}$:
  - I has to pick $r_i \in \mathfrak{R}$ with $\mathrm{dom}(r_i) = r_{i-1}(q_{i-1})$,
  - O has to pick $q_i \in \mathrm{dom}(r_i)$.
- Let $q_i = (q_i', c_i)$. O wins play if $c_0 c_1 c_2 \cdots$ satisfies parity condition.

The Game $\mathcal{G}(\mathcal{A})$

Define new game $\mathcal{G}(\mathcal{A})$ between I and O:

- In round 0:
  - I has to pick $r_0 \in \mathfrak{R}$ with $\mathrm{dom}(r_0) = \{q_I^{\mathcal{C}}\}$,
  - O has to pick $q_0 \in \mathrm{dom}(r_0)$ (i.e., $q_0 = q_I^{\mathcal{C}}$).
- Round $i > 0$ with play prefix $r_0 q_0 \cdots r_{i-1} q_{i-1}$:
  - I has to pick $r_i \in \mathfrak{R}$ with $\mathrm{dom}(r_i) = r_{i-1}(q_{i-1})$,
  - O has to pick $q_i \in \mathrm{dom}(r_i)$.
- Let $q_i = (q'_i, c_i)$. O wins play if $c_0 c_1 c_2 \cdots$ satisfies parity condition.

Lemma

O wins $\Gamma_f(L(\mathcal{A}))$ for some f if and only if O wins $\mathcal{G}(\mathcal{A})$.

O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.

$\mathcal{G}$

I:

O:

Γ

I:

O:

O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.

	I:	r_0
$\mathcal{G}$		
	O:	

	I:	
Γ		
	O:	

O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.

$\mathcal{G}$	I:	r_0	
	O:		$q_0 = q_0^{\mathcal{C}}$

Γ	I:	
	O:	

O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

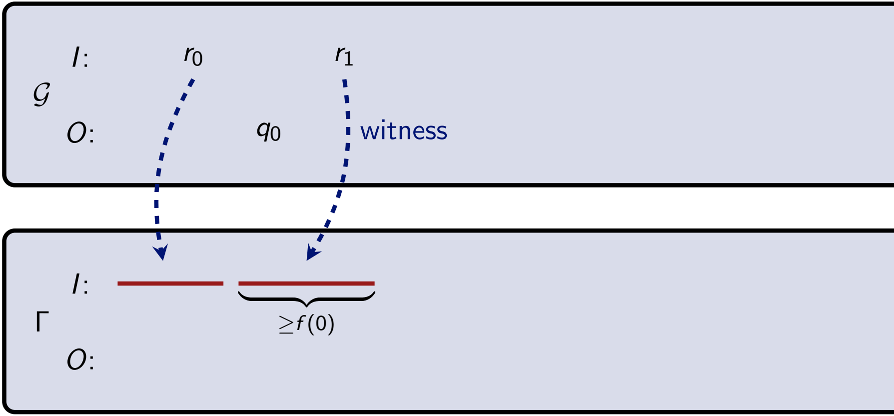
We can assume f to be constant **[HKT10]**.

$\mathcal{G}$	I:	r_0		r_1
	O:		q_0	

Γ	I:	
	O:	

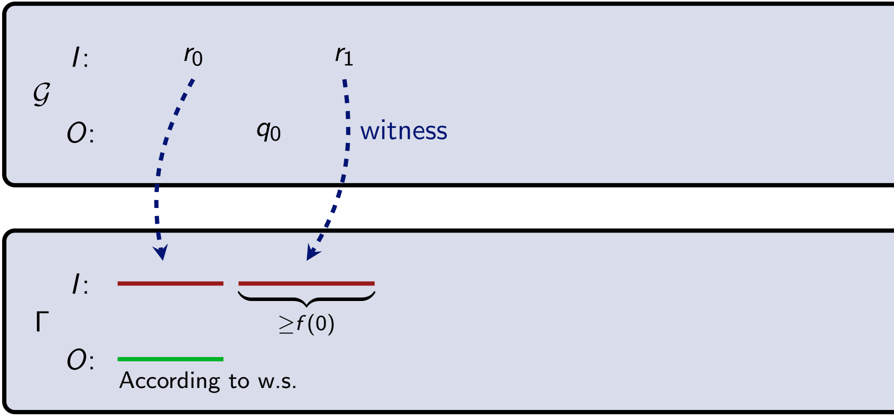
O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.



O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.

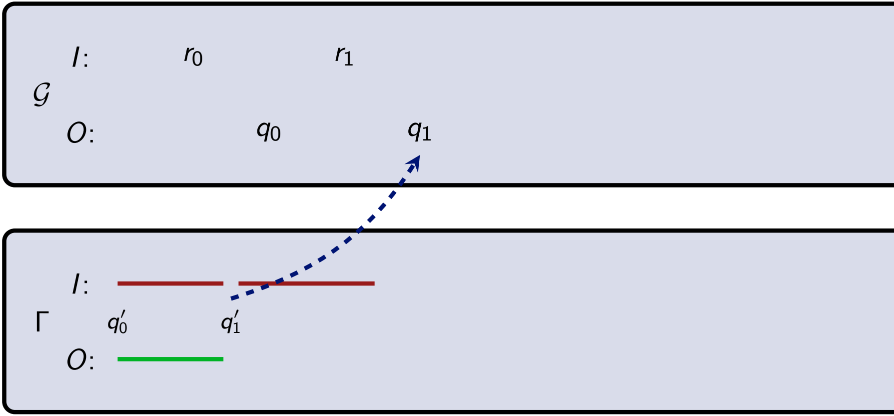


O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.

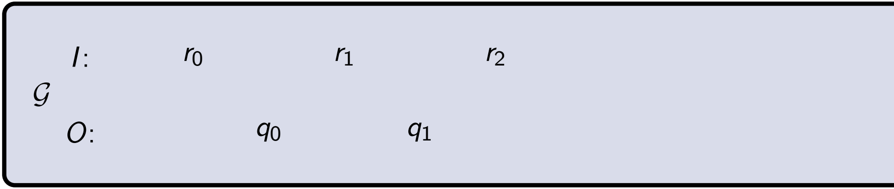
O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.



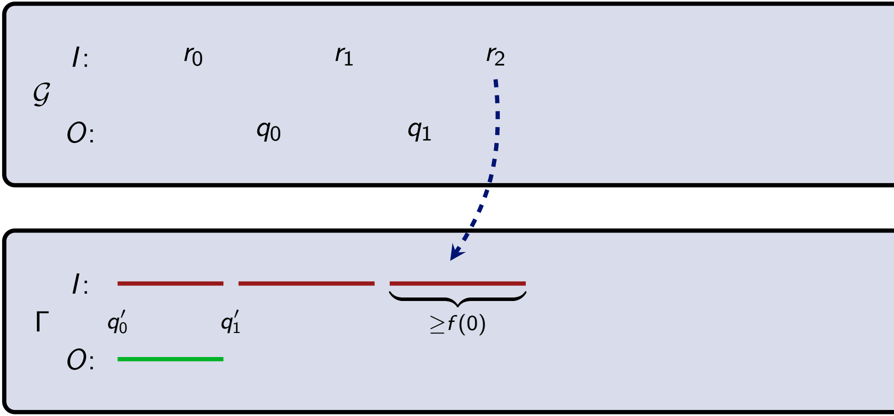
O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.



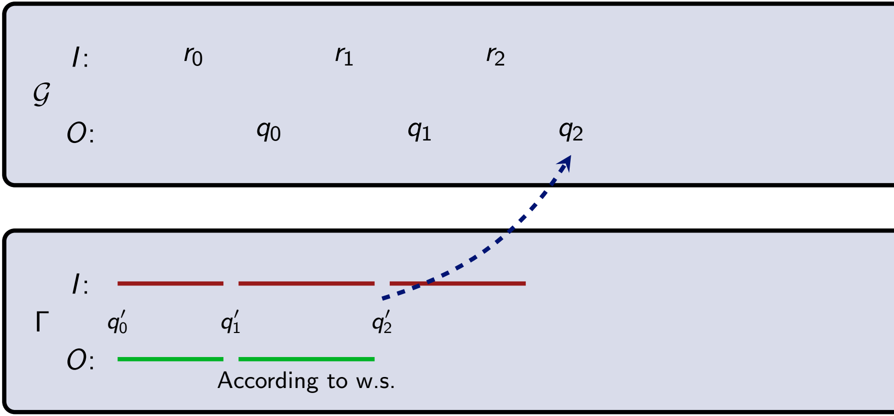
O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.



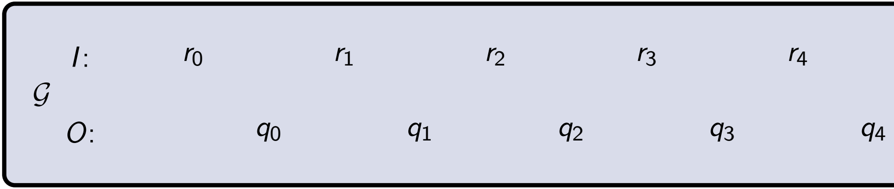
O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

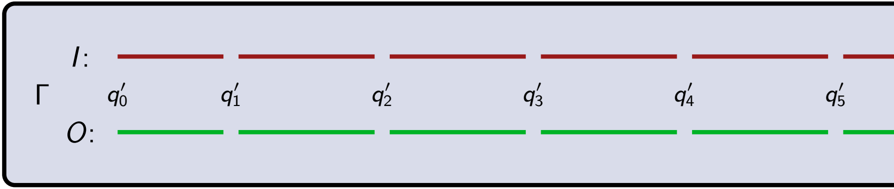
We can assume f to be constant **[HKT10]**.



O wins $\Gamma_f(L(\mathcal{A})) \Rightarrow O$ wins $\mathcal{G}(\mathcal{A})$

We can assume f to be constant **[HKT10]**.





Color encoded in q_i is maximal one seen on run from q'_{i-1} to q'_i in play of $\Gamma \Rightarrow$ Play in $\mathcal{G}$ winning for O.

O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.

Γ I:

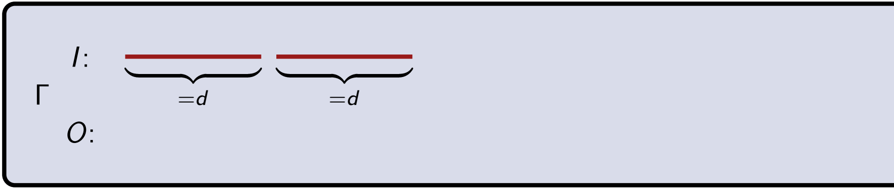
O:

$\mathcal{G}$ I:

O:

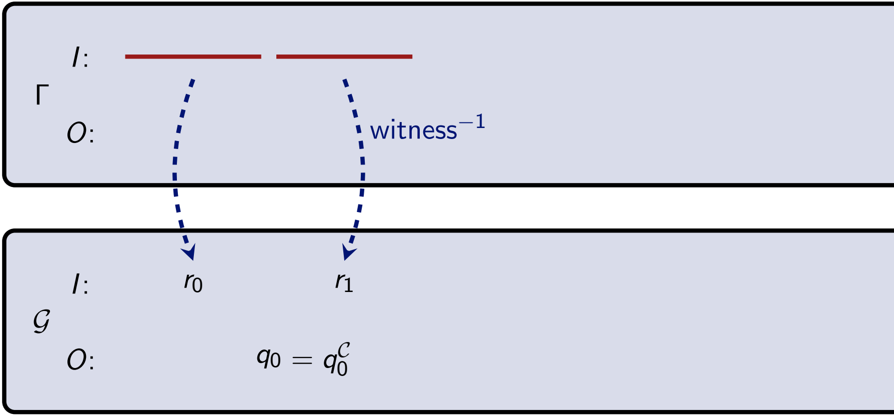
O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.



O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.



O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.

Γ

I:

O:

$\mathcal{G}$

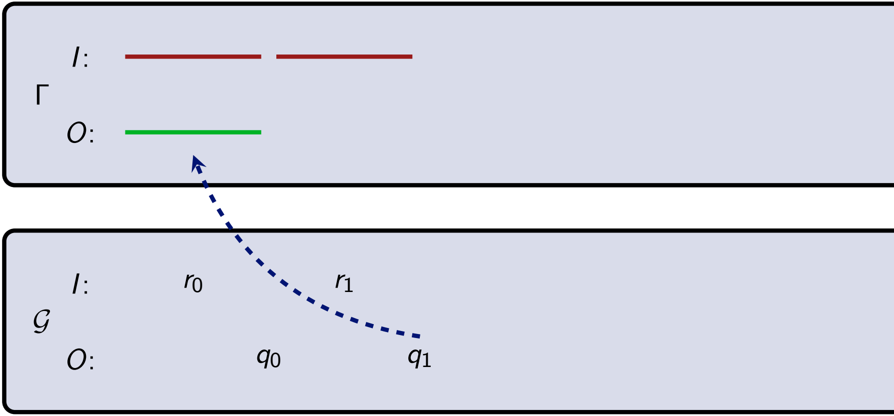
I: r_0 r_1

O: q_0 q_1

According to w.s.

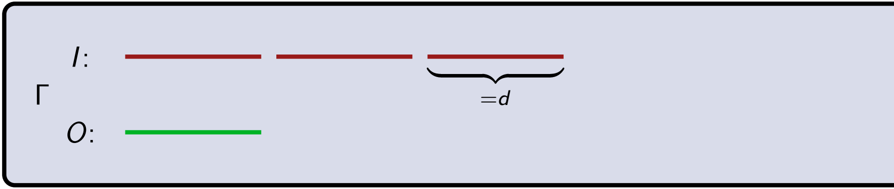
O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

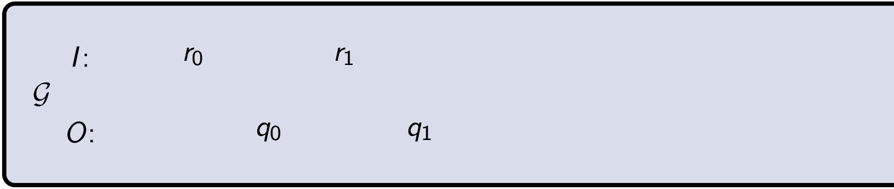
Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.



O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

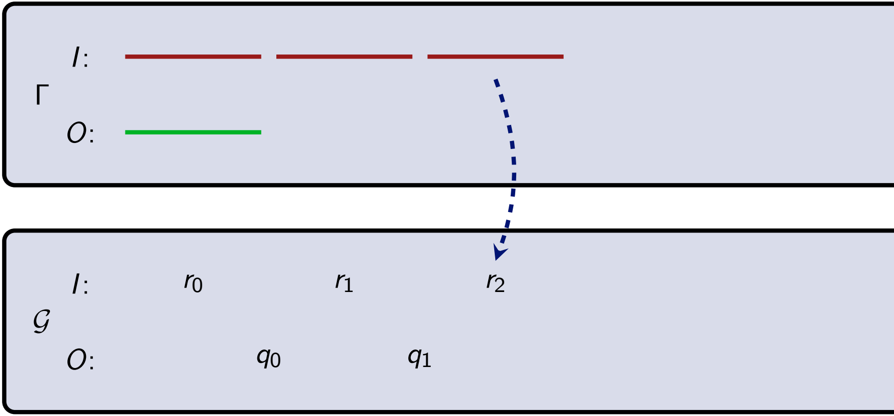
Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.





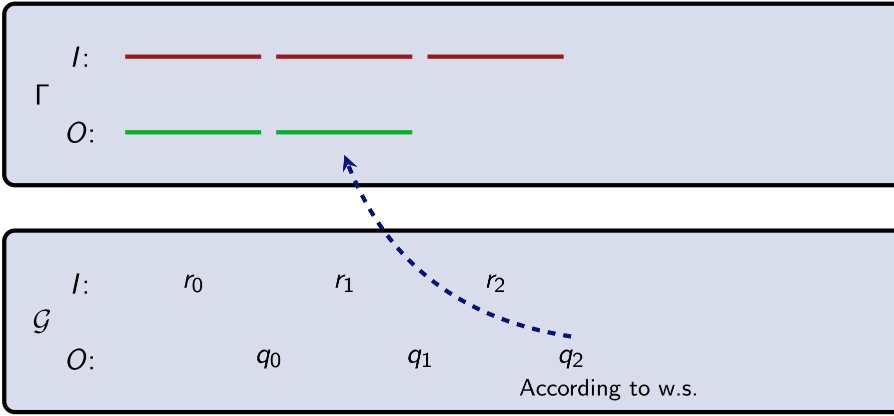
O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.



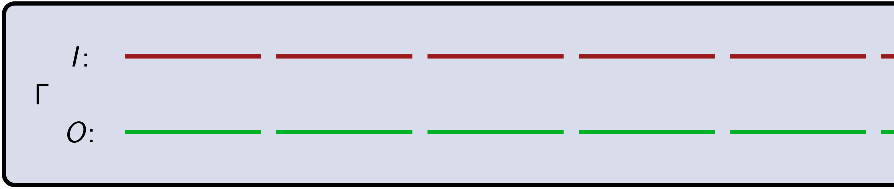
O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

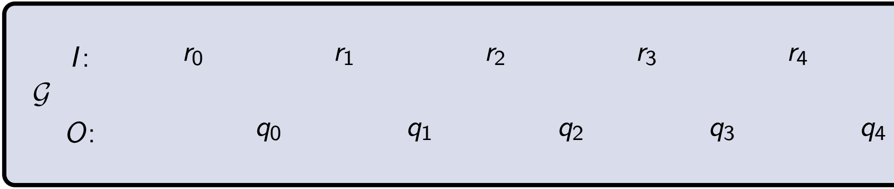
Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.



O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

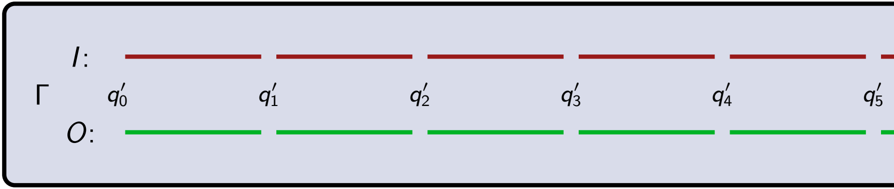
Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.

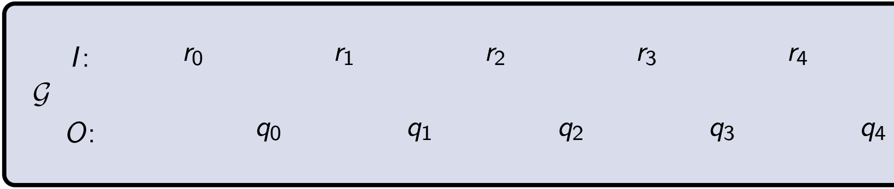




O wins $\Gamma_f(L(\mathcal{A})) \Leftarrow O$ wins $\mathcal{G}(\mathcal{A})$

Let $d = 2^{|\mathcal{C}|^2}$ and $f(0) = 2d$.





Color encoded in q_i is maximal one seen on run from q'_{i-1} to q'_i in play of $\Gamma \Rightarrow$ Play in Γ winning for O.

Finishing the Proof

- $\mathcal{G}(\mathcal{A})$ can be encoded as parity game of exponential size with the same colors as $\mathcal{A}$.
- Such a game can be solved in exponential time in $|\mathcal{A}|$.

Finishing the Proof

- $\mathcal{G}(\mathcal{A})$ can be encoded as parity game of exponential size with the same colors as $\mathcal{A}$.
- Such a game can be solved in exponential time in $|\mathcal{A}|$.

Applying both directions of equivalence between $\Gamma_f(L(\mathcal{A}))$ and $\mathcal{G}(\mathcal{A})$ 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 $f(0) \leq 2^{(|\mathcal{A}|k)^2+1}$.*

Finishing the Proof

- $\mathcal{G}(\mathcal{A})$ can be encoded as parity game of exponential size with the same colors as $\mathcal{A}$.
- Such a game can be solved in exponential time in $|\mathcal{A}|$.

Applying both directions of equivalence between $\Gamma_f(L(\mathcal{A}))$ and $\mathcal{G}(\mathcal{A})$ 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 $f(0) \leq 2^{(|\mathcal{A}|k)^2+1}$.*

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