Degrees of Lookahead in Context-free Infinite Games

Joint work with Wladimir Fridman and Christof Löding

Martin Zimmermann

RWTH Aachen University

May 5th, 2011

Gasics Meeting Spring 2011
Mons, Belgium

Motivation

Starting points:

- Walukiewicz: Solving games with deterministic context-free winning conditions in exponential time.

Motivation

Starting points:

- Walukiewicz: Solving games with deterministic context-free winning conditions in exponential time.
- Hosch & Landweber; Holtmann, Kaiser & Thomas: Delay games with regular winning conditions.

Motivation

Starting points:

- Walukiewicz: Solving games with deterministic context-free winning conditions in exponential time.
- Hosch & Landweber; Holtmann, Kaiser & Thomas: Delay games with regular winning conditions.

Here: delay games with deterministic context-free winning conditions.

- Algorithmic properties.
- Bounds on delay.

Outline

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

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player 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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and $f(i) = 2$ for all i

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I: 0 0

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0
O:	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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0
O:	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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0
O:	0	0		

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0
O:	0	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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0
O:	0	0	0			

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0
O:	0	0	0					

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0
O:	0	0	0	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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0
O:	0	0	0	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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0
O:	0	0	0	0	1					

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0	0	1
O:	0	0	0	0	1							

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0	0	1
O:	0	0	0	0	1	1						

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0	0	1	0	0
O:	0	0	0	0	1	1								

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1} \binom{0}{1}^n \binom{1}{*} \binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0	0	1	0	0
O:	0	0	0	0	1	1	1							

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).
- In round i:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

I:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
O:	0	0	0	0	1	1	1									

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:
  - Player I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - Player O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- Player O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Example

$\binom{0}{0}^\omega$ or $\binom{0}{0}^n\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ or $\binom{0}{0}^{n+1}\binom{0}{1}^n\binom{1}{*}\binom{*}{*}^\omega$ and $f(i) = 2$ for all i

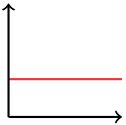
I:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
O:	0	0	0	0	1	1	1	1								

Classifying Delay Functions

1. constant delay function: $f(0) = d$ and $f(i) = 1$ for $i > 0$.

Lookahead
$\sum_{i=0}^{n-1} f(i) - n$

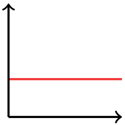
1.)

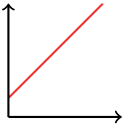


Classifying Delay Functions

1. constant delay function: $f(0) = d$ and $f(i) = 1$ for $i > 0$.
2. linear delay function: $f(i) = k$ for $i \geq 0$.

Lookahead
$$\sum_{i=0}^{n-1} f(i) - n$$

1.) 

2.) 

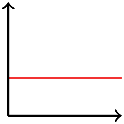
Classifying Delay Functions

1. constant delay function: $f(0) = d$ and $f(i) = 1$ for $i > 0$.
2. linear delay function: $f(i) = k$ for $i \geq 0$.
3. elementary delay function: $[n \mapsto \sum_{i=0}^{n} f(i)] \in \mathcal{O}(\exp_k)$ for some k-fold exponential $\exp_k$.

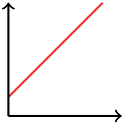
Lookahead

$$\sum_{i=0}^{n-1} f(i) - n$$

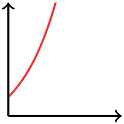
1.)



2.)



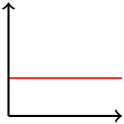
3.)

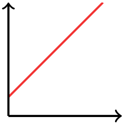


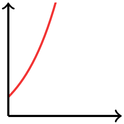
Classifying Delay Functions

1. constant delay function: $f(0) = d$ and $f(i) = 1$ for $i > 0$.
2. linear delay function: $f(i) = k$ for $i \geq 0$.
3. elementary delay function: $[n \mapsto \sum_{i=0}^{n} f(i)] \in \mathcal{O}(\exp_k)$ for some k-fold exponential $\exp_k$.

Lookahead $\sum_{i=0}^{n-1} f(i) - n$

1.) 

2.) 

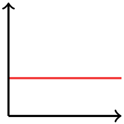
3.) 

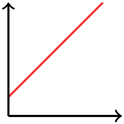
Player O wins the game induced by L with finite (constant, linear, elementary) delay, if there exists an arbitrary (constant, linear, elementary) function f s.t. O has a winning strategy for $\Gamma_f(L)$.

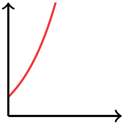
Classifying Delay Functions

1. constant delay function: $f(0) = d$ and $f(i) = 1$ for $i > 0$.
2. linear delay function: $f(i) = k$ for $i \geq 0$.
3. elementary delay function: $[n \mapsto \sum_{i=0}^{n} f(i)] \in \mathcal{O}(\exp_k)$ for some k-fold exponential $\exp_k$.

Lookahead $\sum_{i=0}^{n-1} f(i) - n$

1.) 

2.) 

3.) 

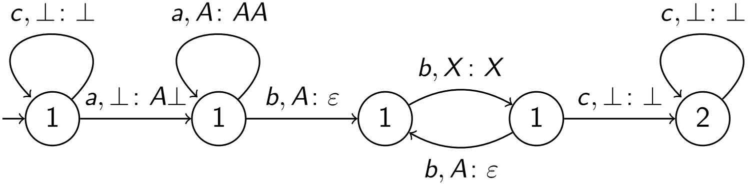
Player O wins the game induced by L with finite (constant, linear, elementary) delay, if there exists an arbitrary (constant, linear, elementary) function f s.t. O has a winning strategy for $\Gamma_f(L)$.

Theorem (HL72, HKT10)

For regular L: Player O wins the game induced by L with finite delay iff she wins it with double-exponential constant delay.

ω-Pushdown Automata

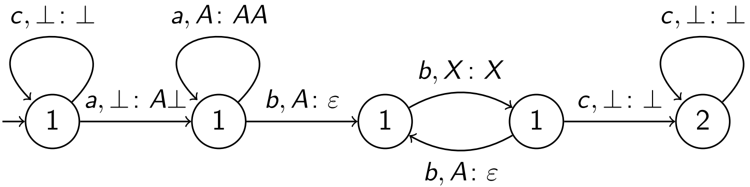
Winning conditions: L recognized by a deterministic ω-pushdown automaton with parity acceptance (parity-DPDA).



Language: $\{c^* a^n b^{2n} c^\omega \mid n > 0\}$.

ω-Pushdown Automata

Winning conditions: L recognized by a deterministic ω-pushdown automaton with parity acceptance (parity-DPDA).



Language: $\{c^* a^n b^{2n} c^\omega \mid n > 0\}$.

Restrictions:

- One-counter: just one stack symbol.
- Visibly: $\Sigma = \underbrace{\Sigma_c}_{Push} \cup \underbrace{\Sigma_r}_{Pop} \cup \underbrace{\Sigma_s}_{Skip}$.

Outline

1. Definitions
2. **Undecidability Results**
3. Lower Bounds on Delay
4. Conclusion

A Decidable Case

Theorem

The following problem is decidable:

Input: *Parity-DPDA $\mathcal{A}$ and f s.t. $\{i \mid f(i) \neq 1\}$ is finite.*

Question: *Does Player O win $\Gamma_f(L(\mathcal{A}))$?*

A Decidable Case

Theorem

The following problem is decidable:

Input: *Parity-DPDA* $\mathcal{A}$ *and* f *s.t.* $\{i \mid f(i) \neq 1\}$ *is finite.*

Question: *Does Player O win* $\Gamma_f(L(\mathcal{A}))$*?*

Proof Idea

- Suppose $f(0) = 3$, $f(1) = 2$, $f(i) = 1$ for $i > 1$.

A Decidable Case

Theorem

The following problem is decidable:

Input: *Parity-DPDA $\mathcal{A}$ and f s.t. $\{i \mid f(i) \neq 1\}$ is finite.*

Question: *Does Player O win $\Gamma_f(L(\mathcal{A}))$?*

Proof Idea

- Suppose $f(0) = 3$, $f(1) = 2$, $f(i) = 1$ for $i > 1$.
- $L' = \{ \binom{\alpha(0)}{\$}\binom{\alpha(1)}{\$}\binom{\alpha(2)}{\beta(0)}\binom{\alpha(3)}{\$}\binom{\alpha(4)}{\beta(1)}\binom{\alpha(5)}{\beta(2)} \cdots \mid \binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\binom{\alpha(2)}{\beta(2)} \cdots \in L(\mathcal{A}) \}$.
- L' deterministic context-free.

A Decidable Case

Theorem

The following problem is decidable:

Input: *Parity-DPDA $\mathcal{A}$ and f s.t. $\{i \mid f(i) \neq 1\}$ is finite.*

Question: *Does Player O win $\Gamma_f(L(\mathcal{A}))$?*

Proof Idea

- Suppose $f(0) = 3$, $f(1) = 2$, $f(i) = 1$ for $i > 1$.
- $L' = \{\binom{\alpha(0)}{\$}\binom{\alpha(1)}{\$}\binom{\alpha(2)}{\beta(0)}\binom{\alpha(3)}{\$}\binom{\alpha(4)}{\beta(1)}\binom{\alpha(5)}{\beta(2)}\cdots \mid \binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)}\binom{\alpha(2)}{\beta(2)}\cdots \in L(\mathcal{A})\}$.
- L' deterministic context-free.
- Now we have a game without delay.
- Apply Walukiewicz's Theorem: Games with deterministic context-free winning conditions can be solved effectively.

Undecidability

Theorem

The following problem is undecidable:

Input: *Parity-DPDA $\mathcal{A}$.*

Question: *Does Player O win the game induced by $L(\mathcal{A})$ with finite delay?*

Undecidability

Theorem

The following problem is undecidable:

Input: *Parity-DPDA $\mathcal{A}$.*

Question: *Does Player O win the game induced by $L(\mathcal{A})$ with finite delay?*

Proof Idea

Preliminaries:

- Reduction from halting problem for 2-register machines.
- Encode configuration (ℓ, n_0, n_1) by $\ell a^{n_0} b^{n_1}$.
- $\ell a^{n_0} b^{n_1} \vdash \ell' a^{n'_0} b^{n'_1}$ is checkable by DPDA.

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

```
$  0  $  1  a  $
```

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

```
$   0   $   1   a   $
N
```

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

N: Player O claims no error.

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

$ 0 $ 1 a $ 2 a b $ 3 a
N

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

```
$ 0 $ 1 a $ 2 a b $ 3 a
N -
```

```
0: INC(X0)
1: INC(X1)
2: IF(X1=0) GOTO 5
3: DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$ 0 \$ 1 a \$ 2 a b \$ 3 a b \$ 4 a b \$
N -

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N														

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

N: Player O claims no error.

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N	-													

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N	-	-												

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N	-	-	N											

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

N: Player O claims no error.

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

$	0	$	1	a	$	2	a	b	$	3	a	b	$	4	a	b $
N	-	N	-	-	N	-										

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b	\$
N	-	N	-	-	N	-	-										

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N	-	-	N	-	-	-								

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

Proof Idea

- Player I produces configurations $c_0, c_1, \ldots$.
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b \$
N	-	N	-	-	N	-	-	-	R_0							

```
0:  INC(X0)
1:  INC(X1)
2:  IF(X1=0) GOTO 5
3:  DEC(X0)
...
```

R_0: Player O claims error in `X0`.

Player O wins:
$(3, 1, 1) \not\vdash (4, 1, 1)$

Proof Idea

- Player I produces configurations c_0, c_1, ...
- Player O can check once whether $c_i \vdash c_{i+1}$ holds.
- If $c_i \vdash c_{i+1}$, Player I wins, otherwise Player O wins.

Example

\$	0	\$	1	a	\$	2	a	b	\$	3	a	b	\$	4	a	b	\$
N	-	N	-	-	N	-	-	-	R_0								

- If machine halts, Player I has to cheat. Player O can detect this with linear delay and wins.
- If machine does not halt, Player I can play forever without cheating and wins.

More Undecidability

Corollary

The following problems are undecidable:

- **Input:** *Parity-DPDA $\mathcal{A}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with constant delay?*

More Undecidability

Corollary

The following problems are undecidable:

- **Input:** *Parity-DPDA $\mathcal{A}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with constant delay?*
- **Input:** *Parity-DPDA $\mathcal{A}$ and $k \in \mathbb{N}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with linear delay k?*

More Undecidability

Corollary

The following problems are undecidable:

- **Input:** *Parity-DPDA $\mathcal{A}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with constant delay?*
- **Input:** *Parity-DPDA $\mathcal{A}$ and $k \in \mathbb{N}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with linear delay k?*
- **Input:** *Parity-DPDA $\mathcal{A}$.*
  Question: *Does Player O win the game induced by $L(\mathcal{A})$ with linear delay?*

More Undecidability

Corollary

The following problems are undecidable:

- **Input:** *Parity-DPDA* $\mathcal{A}$.
  Question: *Does Player O win the game induced by* $L(\mathcal{A})$ *with constant delay?*
- **Input:** *Parity-DPDA* $\mathcal{A}$ *and* $k \in \mathbb{N}$.
  Question: *Does Player O win the game induced by* $L(\mathcal{A})$ *with linear delay* k*?*
- **Input:** *Parity-DPDA* $\mathcal{A}$.
  Question: *Does Player O win the game induced by* $L(\mathcal{A})$ *with linear delay?*

Undecidability results hold for visibly one-counter winning conditions: let Player O control the stack.

Outline

1. Definitions
2. Undecidability Results
3. **Lower Bounds on Delay**
4. Conclusion

Lower Bounds on Delay

Theorem

There exists a parity-DPDA $\mathcal{A}$ such that Player O wins the game induced by $L(\mathcal{A})$ with finite delay, but for any elementary delay function f, the game $\Gamma_f(L(\mathcal{A}))$ is won by Player I.

Lower Bounds on Delay

Theorem

There exists a parity-DPDA $\mathcal{A}$ such that Player O wins the game induced by $L(\mathcal{A})$ with finite delay, but for any elementary delay function f, the game $\Gamma_f(L(\mathcal{A}))$ is won by Player I.

Proof Idea

Preliminaries:

- Adapt idea from undecidability proof.
- Player I produces blocks on which a successor relation is defined (which can be checked by a DPDA).
- Player I has to cheat at some point.
- Player O wins if she catches Player I.

Proof Idea

- 0-th block: $w_0 = 0$.
- $(n+1)$-st block: $w_{n+1} = \$\,0\,\$\,00\,\$\,0000\,\$\,\cdots\,\$\,0^{2^{|w_n|}}\,\$$.
- $|w_{n+1}| > \sum_{i=0}^{|w_n|} 2^i = 2^{|w_n|+1} - 1$.

Proof Idea

- 0-th block: $w_0 = 0$.
- $(n+1)$-st block: $w_{n+1} = \$\,0\,\$\,00\,\$\,0000\,\$\,\cdots\,\$\,0^{2^{|w_n|}}\,\$$.
- $|w_{n+1}| > \sum_{i=0}^{|w_n|} 2^i = 2^{|w_n|+1} - 1$.

Now, both players have to announce errors:

- Copy error: $|w_{n+1}|_\$ \neq |w_n| + 1$.
- Doubling error: infix $\$0^m\$0^n\$$ s.t. $n \neq 2m$.
- Both errors can be checked by a DPDA.

Proof Idea

- 0-th block: $w_0 = 0$.
- $(n+1)$-st block: $w_{n+1} = \$\,0\,\$\,00\,\$\,0000\,\$\ \cdots\ \$\,0^{2^{|w_n|}}\,\$$.
- $|w_{n+1}| > \sum_{i=0}^{|w_n|} 2^i = 2^{|w_n|+1} - 1$.

Now, both players have to announce errors:

- Copy error: $|w_{n+1}|_{\$} \neq |w_n| + 1$.
- Doubling error: infix $\$0^m\$0^n\$$ s.t. $n \neq 2m$.
- Both errors can be checked by a DPDA.

Player O wins if:

- she claims an existing error before Player I does, or
- Player I claims an non-existing error, or
- Player I never claims an error.

Proof Idea

- 0-th block: $w_0 = 0$.
- $(n+1)$-st block: $w_{n+1} = \$\,0\,\$\,00\,\$\,0000\,\$\,\cdots\,\$\,0^{2^{|w_n|}}\,\$$.
- $|w_{n+1}| > \sum_{i=0}^{|w_n|} 2^i = 2^{|w_n|+1} - 1$.

Now, both players have to announce errors:

- Copy error: $|w_{n+1}|_{\$} \neq |w_n| + 1$.
- Doubling error: infix $\$0^m\$0^n\$$ s.t. $n \neq 2m$.
- Both errors can be checked by a DPDA.

Player O wins if:

- she claims an existing error before Player I does, or
- Player I claims an non-existing error, or
- Player I never claims an error.

Player O needs non-elementary lookahead to win this game.

Outline

1. Definitions
2. Undecidability Results
3. Lower Bounds on Delay
4. **Conclusion**

Conclusion

Delay games with context-free winning conditions.

- Determining the winner is undecidable.
- This holds even for restricted classes of winning conditions.

Conclusion

Delay games with context-free winning conditions.

- Determining the winner is undecidable.
- This holds even for restricted classes of winning conditions.
- Non-elementary lower bounds on delay.
- Again, also for restricted classes of winning conditions.

Conclusion

Delay games with context-free winning conditions.

- Determining the winner is undecidable.
- This holds even for restricted classes of winning conditions.
- Non-elementary lower bounds on delay.
- Again, also for restricted classes of winning conditions.

Open questions:
Undecidability and non-elementary lower bounds, if Player O controls the stack.

- What if Player I controls the stack?
- Linear delay necessary in this case.