Recognition of Linear-Slender Context-Free Languages by Real Time One-Way Cellular Automata

Véronique Terrier

GREYC - Université de Caen

Introduction

Objective

To better understand the algorithmic capacity of cellular automata

We will show how one of the simplest types of cellular automata can simulate a restricted variant of context free languages.

Context-free language (CFL)

$\{\mathtt{a}^{\mathtt{n}}\mathtt{b}^{\mathtt{n}+\mathtt{m}}\mathtt{a}^{\mathtt{m}} \colon \mathtt{n} \geq 0, \mathtt{m} \geq 0\}$

$$\begin{aligned} S &\to XY \\ X &\to aXb \mid \varepsilon \\ Y &\to bXa \mid \varepsilon \end{aligned}$$

Linear CFL

$\mathtt{L}_0 = \{\mathtt{a}^{\mathtt{u}}\mathtt{w}\mathtt{b}^{\mathtt{u}} \colon \mathtt{u} > 0, \mathtt{w} = \varepsilon \text{ or } \mathtt{w} \in \mathtt{b}\{\mathtt{a}, \mathtt{b}\}^*\mathtt{a}\}$

$$\begin{aligned} S &\to aSb \mid abTab \mid ab \\ T &\to Ta \mid Tb \mid \varepsilon \end{aligned}$$

Poly-slender and linear-slender CFL

The counting function of a language L

$\sharp_n(L)$: the number of words in L of length n

A language L is

- *k-poly-slender* if $\sharp_n(L)$ is in $\mathcal{O}(n^k)$.
- *linear-slender* if $\sharp_n(L)$ is in $\mathcal{O}(n)$.

Examples

$\{\mathtt{a}^{\mathtt{n}}\mathtt{b}^{\mathtt{n+m}}\mathtt{a}^{\mathtt{m}} : \mathtt{n} \geq 0, \mathtt{m} \geq 0\}$ is a linear-slender CFL

$$\sharp_n(L) = \begin{cases} 0 & \text{if } n \text{ is odd} \\ n/2 - 1 & \text{if } n \text{ is even} \end{cases}$$

$\mathtt{L}_0 = \{\mathtt{a}^{\mathtt{u}}\mathtt{w}\mathtt{b}^{\mathtt{u}} : \mathtt{u} > 0, \mathtt{w} = \varepsilon$ or $\mathtt{w} \in \mathtt{b}\{\mathtt{a}, \mathtt{b}\}^*\mathtt{a}\}$ is not a poly-slender CFL

$\sharp_n(L_0) \sim 2^n$

$\{\mathtt{a}^{\mathtt{n}_1}\mathtt{b}^{\mathtt{n}_2}\mathtt{a}^{\mathtt{n}_3}\mathtt{b}^{\mathtt{n}_4}\mathtt{a}^{\mathtt{n}_5} : \mathtt{n}_1 + \mathtt{n}_3 + \mathtt{n}_5 = \mathtt{n}_2 + \mathtt{n}_4\}$ is a 3-poly-slender CFL

$\sharp_n(L) \sim n^3$

Poly-slender and linear-slender CFL

Characterization in terms of Dyck Loops

k-Dyck Loop (Ilie, Rozenberg and Salomaa)

Given

- a Dyck word on $\{[,]\}$: $z_1 z_2 \cdots z_{2k}$
- some words $y_0, y_1, \cdots, y_{2k}$ and $x_1, \cdots, x_{2k}$
- a map
  $h_{n_1,\cdots,n_k}(y_0 z_1 y_1 z_2 y_2 \cdots z_{2k} y_{2k}) = y_0 x_1^{e_1} y_1 x_2^{e_2} y_2 \cdots x_{2k}^{e_{2k}} y_{2k}$
  where, if z_l and z_r are the i-th matching parenthesis then both exponents e_l and e_r are n_i.

A k-Dyck loop is

$$\{h_{n_1,\cdots,n_k}(y_0 z_1 y_1 z_2 y_2 \cdots y_{2k-1} z_{2k} y_{2k}) \colon n_i \geq 0\}$$

Poly-slender and linear-slender CFL

Characterization in terms of Dyck Loops

Examples

$\{a^{n_1}b^{n_1+n_2}a^{n_2} : n_1, n_2 \geq 0\}$ is a 2-Dyck loop with Dyck word [][], $y_0 = \cdots = y_4 = \varepsilon$ and $x_1 = x_4 = a, x_2 = x_3 = b$

$\{a^{n_1}b^{n_1+n_2}a^{n_2+n_3}b^{n_3+n_4}a^{n_4} : n_i \geq 0\}$ is a 4-Dyck loop with Dyck word [][][][], $y_i = \varepsilon$ and $x_1 = x_4 = x_5 = x_8 = a, x_2 = x_3 = x_6 = x_7 = b$

$\{a^{n_1+n_2}b^{n_2}a^{n_3}b^{n_1+n_3+n_4}a^{n_4} : n_i \geq 0\}$ is a 4-Dyck loop with Dyck word [[][]][]

Theorem (Ilie, Rozenberg and Salomaa)

For any $k \geq 0$, a CFL is k-poly-slender if and only if it is a finite union of $(k+1)$-Dyck loops.

Real time OCA

A real time OCA is specified by:

- an input alphabet Σ

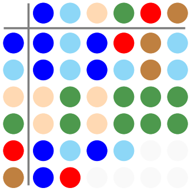
- a finite set of states S $\quad (S \supset \Sigma)$

- a subset of accepting states F

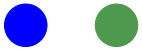
- a transition function $\delta : S \times S \to S$



Real time OCA

A real time OCA is specified by:

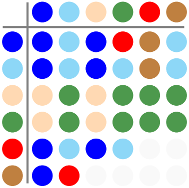
- an input alphabet Σ



- a finite set of states S $(S \supset \Sigma)$

- a subset of accepting states F

- a transition function $\delta : S \times S \to S$



Real time OCA

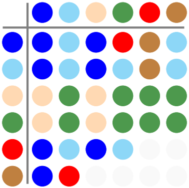
A real time OCA is specified by:

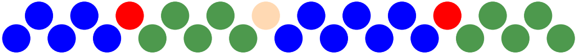
- an input alphabet Σ

- a finite set of states S $(S \supset \Sigma)$

- a subset of accepting states F

- a transition function $\delta : S \times S \to S$





Real time OCA

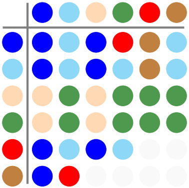
A real time OCA is specified by:

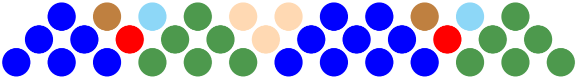
- an input alphabet Σ

- a finite set of states S $\quad (S \supset \Sigma)$

- a subset of accepting states F

- a transition function $\delta : S \times S \to S$





Real time OCA

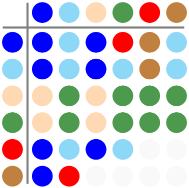
A real time OCA is specified by:

- an input alphabet Σ

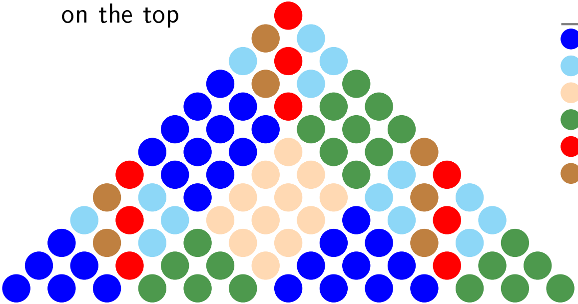
- a finite set of states S $(S \supset \Sigma)$

- a subset of accepting states F

- a transition function $\delta : S \times S \to S$



Result of the computation on the top

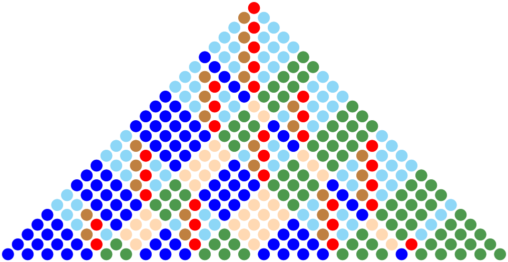


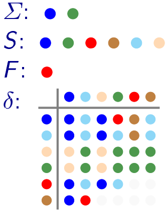
Real time OCA

Language recognition

The language accepted by a real time OCA

The set of all words whose computation ends in an accepting state





accepts

$$L_0 = \{a^u w b^u : u > 0,\ w = \varepsilon \text{ or } w \in b\{a,b\}^*a\}$$

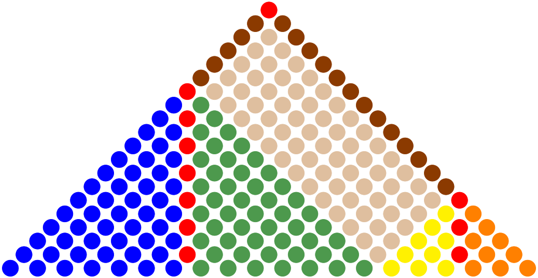
A main feature

The computation of a word contains the computation of all its factors.

Real time OCA

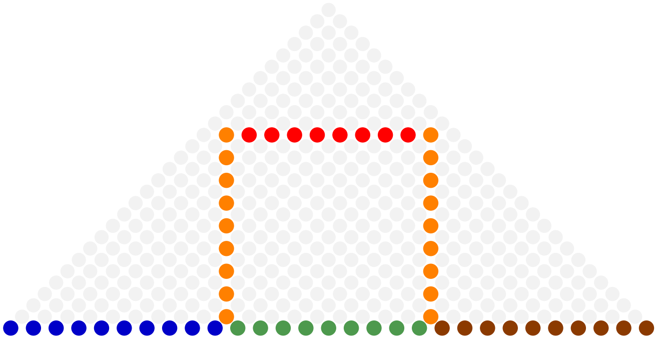
Recognition of

$\{a^n b^n c^m d^m : n \geq 0, m \geq 0\} = \{a^n b^n (c+d)^+ : n \geq 0\} \cap \{(a+b)^+ c^m d^m : m \geq 0\}$



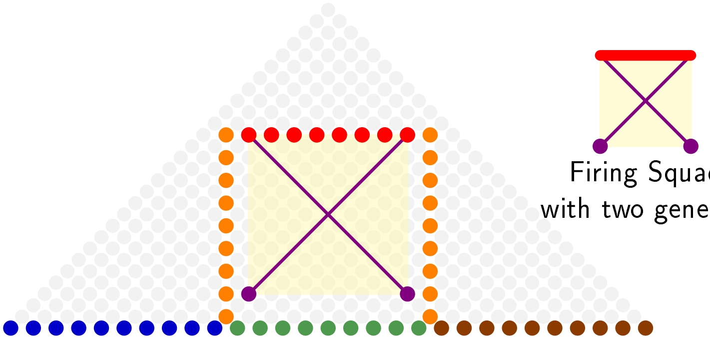
The Čulík algorithm

Recognition of $\{\mathtt{a}^{\mathtt{n}}\mathtt{b}^{\mathtt{n+m}}\mathtt{d}^{\mathtt{m}} : \mathtt{n} \geq 0, \mathtt{m} \geq 0\}$



The Čulík algorithm

Recognition of $\{a^n b^{n+m} d^m : n \geq 0, m \geq 0\}$



Firing Squad
with two generals

Real time OCA

A robust class

The real time OCA class

- is closed under boolean operations
- is not closed under morphism, concatenation
- contains all linear CFL

Characterization in terms of linear conjunctive grammar

A linear CF grammar broaden with a conjunctive operation &

Theorem (Okhotin 2004)

A language L is recognized in real time by an OCA if and only if L is generated by a linear conjunctive grammar.

Relationship between CFL and real time OCA

CFL and real time OCA are incomparable

CFL do not contain real time OCA

$\{a^n b^n c^n : n > 0\}$ is a real time OCA language but not a CFL.

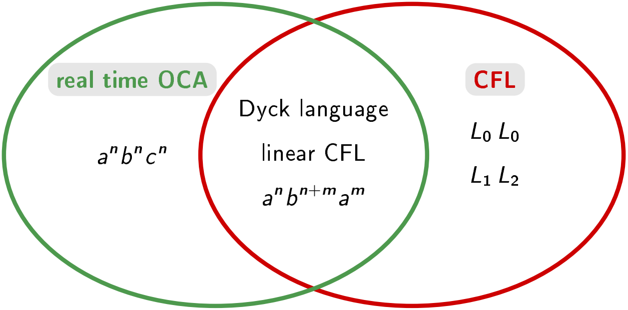
Real time OCA do not contain CFL

$L_0 = \{a^u w b^u : u > 0, w = \varepsilon \text{ or } w \in b\{a,b\}^* a\}$
L_0 is a linear CFL and so a real time OCA language.
$L_0 L_0$ is a CFL but not a real time OCA language.

Real time OCA do not contain deterministic CFL (and even *LL*(1) languages) (Okhotin 2014)

$L_1 = \{c^m b a^{l_1} b \cdots a^{l_{m-1}} b : m > 0, l_i \geq 0\}$
$L_2 = \{a^n b w d^n : n > 0, w \in \{a,b\}^*\}$
L_1 and L_2 are linear CFL and so real time OCA languages.
$L_1 L_2$ is a deterministic CFL (and even a *LL*(1) language) but not a real time OCA language.

Relationship between CFL and real time OCA



Question

What languages do they have in common?

Poly-slender CFL and rt OCA language

Conjecture

The poly-slender CFL are real time OCA languages.

Here we only prove the minimal statement:

Claim

The linear-slender CFL are real time OCA languages.

Recall that the linear-slender CFL correspond to the finite unions of 2-Dyck loops.
$\Rightarrow$ We have to show that the 2-Dyck loops are recognizable by real time OCA.

Poly-slender CFL and rt OCA language

The real question: the closure under concatenation

Theorem (Ginsburg and Spanier)

The family of poly-slender CFL is the smallest family which contains all finite languages and is closed under the following operations:

1. union
2. catenation
3. $(x, y)^\star L = \bigcup_{n \geq 0} x^n L y^n$ for any x, y words

The real time OCA languages

- include all finite languages
- are closed under union and $\star$ operation
- are not closed under concatenation

But, the known witnesses for non-closure under concatenation are not languages inside the family of poly-slender CFL.

2-Dyck loops are real time OCA

Shapes of 2-Dyck loops

Case 1. The underlying Dyck word is [[]]

$$\{y_0 x_1^{n_1} y_1 x_2^{n_2} y_2 x_3^{n_2} y_3 x_4^{n_1} y_4 : n_1, n_2 \geq 0\}$$

Case 2. The underlying Dyck word is [][]

$$\{y_0 x_1^{n_1} y_1 x_2^{n_1} y_2 x_3^{n_2} y_3 x_4^{n_2} y_4 : n_1, n_2 \geq 0\}$$

In the following, we will deal with simplified version of 2-Dyck loops by setting $y_0 = \cdots = y_4 = \varepsilon$.
It does not change the essential arguments.

2-Dyck loops are real time OCA

Shapes of 2-Dyck loops

Case 1. The underlying Dyck word is [[]]
$\{y_0 x_1^{n_1} y_1 x_2^{n_2} y_2 x_3^{n_2} y_3 x_4^{n_1} y_4 : n_1, n_2 \geq 0\}$

Case 2. The underlying Dyck word is [][]
$\{y_0 x_1^{n_1} y_1 x_2^{n_1} y_2 x_3^{n_2} y_3 x_4^{n_2} y_4 : n_1, n_2 \geq 0\}$

In the following, we will deal with simplified version of 2-Dyck loops by setting $y_0 = \cdots = y_4 = \varepsilon$.
It does not change the essential arguments.

Case 1. The underlying Dyck word is [[]]
$\{x_1^{n_1} x_2^{n_2} x_3^{n_2} x_4^{n_1} : n_1, n_2 \geq 0\}$

Case 2. The underlying Dyck word is [][]
$\{x_1^{n_1} x_2^{n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$

2-Dyck loops are real time OCA

Case 1. The underlying Dyck word is [[]]

The corresponding Dyck loop is $\{\mathtt{x}_1^{\mathtt{n}_1}\mathtt{x}_2^{\mathtt{n}_2}\mathtt{x}_3^{\mathtt{n}_2}\mathtt{x}_4^{\mathtt{n}_1} : \mathtt{n}_1, \mathtt{n}_2 \geq 0\}$.
It is a linear CFL language and so it is a real time OCA language.

The closure under $\star$ operation $\left((x, y)^\star L = \bigcup_{n \geq 0} x^n L y^n\right)$ is here involved.

2-Dyck loops are real time OCA

Case 2. The underlying Dyck word is [][]

The corresponding Dyck loop is $\{x_1^{n_1} x_2^{n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$.

The closure under concatenation of 1-Dyck loops is now involved.

Case 2.a. x_1 and x_2 have the same primitive root (or x_3 and x_4)

A degenerated case

$x_1 = z^r, x_2 = z^s$ for some z, r, s

$\{x_1^{n_1} x_2^{n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\} = \{z^{(r+s)n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$ is a linear CFL

Case 2.b. x_2 and x_3 do not have the same primitive root

A type of marked concatenation

The Dyck loop is the intersection of two linear CFL

$$\{x_1^{n_1} x_2^{n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\} = \{x_1^{n_1} x_2^{n_1} x_3^{m} x_4^{p} : n_1, m, p \geq 0\} \cap \{x_1^{m} x_2^{p} x_3^{n_2} x_4^{n_2} : n_2, m, p \geq 0\}$$

Case 2.c. x_2 and x_3 have the same primitive root but not x_1, x_2 and x_3, x_4

The critical case

$\{x_1^{n_1} x_2^{n_1} x_3^{n_2} x_4^{n_2} : n_1, n_2 \geq 0\} = \{x_1^{n_1} z^{r\,n_1 + s\,n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$

2-Dyck loops are real time OCA

Dyck loops of shape $\{x_1^{n_1} z^{r\,n_1+s\,n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$ with x_1 and z having distinct primitive roots as well as z and x_4

- The Čulík's OCA recognizes in real time $\{a^{n_1} b^{n_1+n_2} c^{n_2} : n_1, n_2 \geq 0\}$ with a, b, c distinct letters
- By way of geometric transformations of the Čulík's OCA, we construct real time OCAs which recognize the slight variants: $\{a^{n_1} b^{r\,n_1+s\,n_2} c^{n_2} : n_1, n_2 \geq 0\}$
- Making use of Okhotin's equivalence, we translate these OCAs in terms of linear conjunctive grammars
- Let h be the homomorphism $h(a) = x_1$, $h(b) = z$, $h(c) = x_4$. Providing that $h(a)$ and $h(b)$, as well as $h(b)$ and $h(c)$, have distinct primitive roots, we can modify the previous linear conjunctive grammars to generate $h\left(\{a^{n_1} b^{r\,n_1+s\,n_2} c^{n_2} : n_1, n_2 \geq 0\}\right) = \{x_1^{n_1} z^{r\,n_1+s\,n_2} x_4^{n_2} : n_1, n_2 \geq 0\}$

Conclusion

Linear-slender CFL are real time OCA languages

Three ingredients:

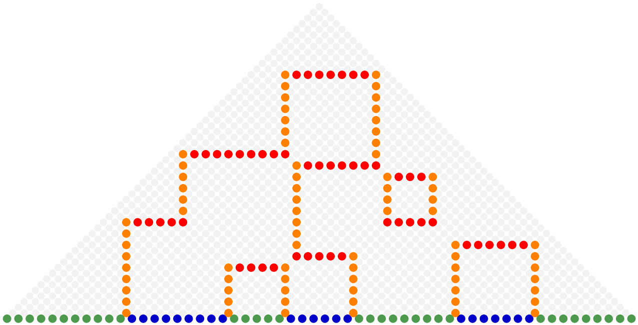
- the characterization of poly-slender CFL in terms of Dyck loops given by Ilie, Rozenberg and Salomaa
- the Čulík's OCA which recognizes in real time the language $\{a^n b^{n+m} a^m : n, m \geq 0\}$
- the Okhotin's characterization of real time OCA by linear conjunctive grammars

Conclusion

Question: Are poly-slender CFL real time OCA languages?

A preliminary step:

> How to extend the Čulík's algorithm for the bounded versions of the language of words whose the number of a's equals the number of b's?



Bibliography

Karel Čulík II.
Variations of the firing squad problem and applications.
Information Processing Letters, 30(3):152 – 157, 1989.

Lucian Ilie, Grzegorz Rozenberg, and Arto Salomaa.
A characterization of poly-slender context-free languages.
Theoretical Informatics and Applications, 34(1):77–86, 2000.

Alexander Okhotin.
On the equivalence of linear conjunctive grammars and trellis automata.
RAIRO Informatique Théorique et Applications, 38(1):69–88, 2004.