



**FACULTY  
OF INFORMATION  
TECHNOLOGY  
CTU IN PRAGUE**

# Bispecial factors in DOL systems

Journées combinatoires de Bordeaux

*Herman Goulet-Ouellet*, Karel Klouda and Štěpán Starosta

03.02.2025

Part 1



**DOL systems**

*"In spite of the simplicity of the basic definitions, the theory of DOL systems is at present very rich and challenging. Apart from providing applications to formal languages and biology, this theory has also shed new light on the very basic mathematical notion of an endomorphism defined on a free monoid."*

A. Salomaa and G. Rozenberg

*The mathematical theory of L systems, 1980*

*"In spite of the simplicity of the basic definitions, the theory of DOL systems is at present very rich and challenging. Apart from providing applications to **formal languages** and biology, this theory has also shed new light on the very basic mathematical notion of an endomorphism defined on a free monoid."*

A. Salomaa and G. Rozenberg

*The mathematical theory of L systems, 1980*



ALPHABET Finite set  $A$  of symbols (letters).



**ALPHABET** Finite set  $A$  of symbols (letters).

**FREE MONOID** Set  $A^*$  of all finite words over  $A$  (strings).



**ALPHABET** Finite set  $A$  of symbols (letters).

**FREE MONOID** Set  $A^*$  of all finite words over  $A$  (strings).

**LENGTH** The length of a word  $u \in A^*$  is denoted  $|u|$ .



**ALPHABET** Finite set  $A$  of symbols (letters).

**FREE MONOID** Set  $A^*$  of all finite words over  $A$  (strings).

**LENGTH** The length of a word  $u \in A^*$  is denoted  $|u|$ .

**EMPTY WORD** The unique word  $\varepsilon \in A^*$  of length 0.





**ALPHABET** Finite set  $A$  of symbols (letters).

**FREE MONOID** Set  $A^*$  of all finite words over  $A$  (strings).

**LENGTH** The length of a word  $u \in A^*$  is denoted  $|u|$ .

**EMPTY WORD** The unique word  $\varepsilon \in A^*$  of length 0.

**CONCATENATION** Write  $uv$  for the concatenation of  $u, v \in A^*$ .

**ALPHABET** Finite set  $A$  of symbols (letters).

**FREE MONOID** Set  $A^*$  of all finite words over  $A$  (strings).

**LENGTH** The length of a word  $u \in A^*$  is denoted  $|u|$ .

**EMPTY WORD** The unique word  $\varepsilon \in A^*$  of length 0.

**CONCATENATION** Write  $uv$  for the concatenation of  $u, v \in A^*$ .

**MORPHISM** Map  $\varphi: A^* \rightarrow B^*$  such that  $\varphi(uv) = \varphi(u)\varphi(v)$ .

D O L  
DETERMINISTIC CONTEXT-FREE LINDENMAYER<sup>1</sup>

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- $A$  is an **alphabet**.
- $\varphi: A^* \rightarrow A^*$  is a **morphism**.
- $w \in A^*$  is a word (**axiom**).

---

<sup>1</sup>Aristid Lindenmayer, 1925–1989, Hungarian biologist.

## Example 1 (Fibonacci)

$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$

## Example 1 (Fibonacci)

$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$

## Example 2 (Thue–Morse)

$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$

## Example 1 (Fibonacci)

$$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$$

## Example 2 (Thue–Morse)

$$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$$

## Example 3 (Quaternary system)

$$(\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$$

## Example 1 (Fibonacci)

$$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$$

## Example 2 (Thue–Morse)

$$(\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$$

## Example 3 (Quaternary system)

$$(\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$$

## Example 4 (Ternary system)

$$(\{a, b, c\}, \varphi: a \mapsto abaca, b \mapsto aba, c \mapsto aba, a)$$

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

SEQUENCE  $(\varphi^n(w))_{n \in \mathbb{N}}$ .

LANGUAGE  $\mathbf{L}(S) = \{u \in A^* \mid \exists n \in \mathbb{N}, \exists w \in W, u \leq_{\text{fac}} \varphi^n(w)\}$ .



$$S = (\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$$

SEQUENCE  $a, ab, aba, abaab, abaababa, abaababaabaab, \dots$

LANGUAGE  $L(S) = \{\varepsilon, a, b, ab, ba, aa, aab, aba, baa, bab, \dots\}$

$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$

**SEQUENCE**  $a, baa, adcbaabaa,$   
 $baaadcdcadcbaabaaadcbaabaa, \dots$

**LANGUAGE**  $L(S) = \{\epsilon, a, b, aa, ab, ad, ba, ca, cb, cd, dc,$   
 $aaa, aab, aad, aba, adc, baa, cad, cba, cdc, \dots\}$

$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$

**SEQUENCE**  $a, baa, adcbaabaa,$   
 $baaadcdcadcbaabaaadcbabaa, \dots$

**LANGUAGE**  $L(S) = \{\varepsilon, a, b, aa, ab, ad, ba, ca, cb, cd, dc,$   
 $aaa, aab, aad, aba, adc, baa, cad, cba, cdc, \dots\}$

- $L(S)$  is computable (proof by SageMath).

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?
- What is the structure of  $\mathbf{L}(S)$ ?

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?
- What is the structure of  $\mathbf{L}(S)$ ?
- How many factors of length  $n$  are in  $\mathbf{L}(S)$ ?

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?
- What is the structure of  $\mathbf{L}(S)$ ?
- How many factors of length  $n$  are in  $\mathbf{L}(S)$ ?
- Where and how do words occur in  $\mathbf{L}(S)$ ?

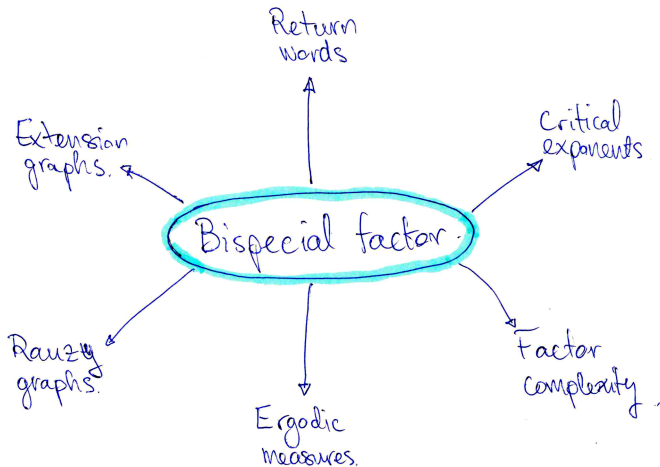
$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?
- What is the structure of  $\mathbf{L}(S)$ ?
- How many factors of length  $n$  are in  $\mathbf{L}(S)$ ?
- Where and how do words occur in  $\mathbf{L}(S)$ ?
- Can words be extended in  $\mathbf{L}(S)$  in a predictable way?



$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- What kinds of infinite words are generated by  $S$  (if any)?
- What is the structure of  $\mathbf{L}(S)$ ?
- How many factors of length  $n$  are in  $\mathbf{L}(S)$ ?
- Where and how do words occur in  $\mathbf{L}(S)$ ?
- **Can words be extended in  $\mathbf{L}(S)$  in a predictable way?**



Part 2



# Special factors

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- $u \in \mathbf{L}(S)$  is **left special** if  $\#\{a \in A \mid au \in \mathbf{L}(S)\} \geq 2$ .

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- $u \in \mathbf{L}(S)$  is **left special** if  $\#\{a \in A \mid au \in \mathbf{L}(S)\} \geq 2$ .
- $u \in \mathbf{L}(S)$  is **right special** if  $\#\{a \in A \mid ua \in \mathbf{L}(S)\} \geq 2$ .

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

- $u \in \mathbf{L}(S)$  is **left special** if  $\#\{a \in A \mid au \in \mathbf{L}(S)\} \geq 2$ .
- $u \in \mathbf{L}(S)$  is **right special** if  $\#\{a \in A \mid ua \in \mathbf{L}(S)\} \geq 2$ .
- $u \in \mathbf{L}(S)$  is **bispecial** if it is both left and right special.

## FIBONACCI

$$S = (\{a, b\}, \varphi: a \mapsto ab, b \mapsto a, a)$$

ordinary

bab.

a — a.

ababa

ordinary

baa.

a — b

abaab

ordinary

aab.

b — a.

baaba

bispecial.

aba.

a — a  
~~b — a~~  
~~a — b~~  
b — b.

aabaa

aabab

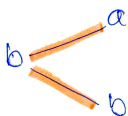
babaa.

## THUE-MORSE

$$S = (\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$$

right special

aab.

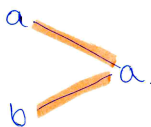


baba

baab

left special

abb.

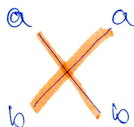


aabba

babba

bispecial

aba.

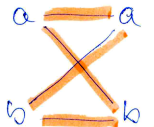


aabab

babaa

bispecial.

abba.



aabbaa

aabbab

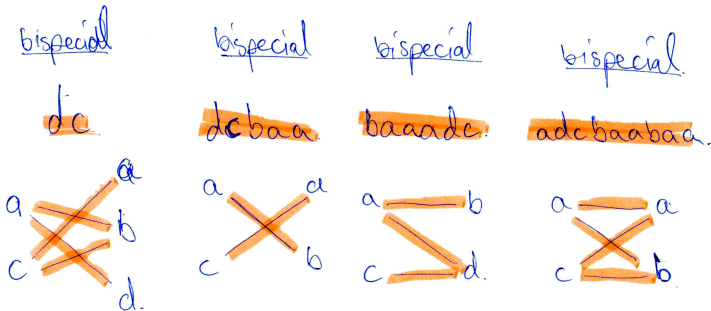
babbaa

babbab.



## QUATERNARY SYSTEM

$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, \quad c)$



## RECURSIVE EXPRESSIONS FOR FACTOR COMPLEXITY

$$p_S(n) = \#\{u \in \mathbf{L}(S) : |u| = n\}$$

$$m_S(u) = \#(\mathbf{L}(S) \cap AuA) - \#(\mathbf{L}(S) \cap uA) - \#(\mathbf{L}(S) \cap uA) + 1$$

(bilateral order)

## RECURSIVE EXPRESSIONS FOR FACTOR COMPLEXITY

$$p_S(n) = \#\{u \in \mathbf{L}(S) : |u| = n\}$$
$$m_S(u) = \#(\mathbf{L}(S) \cap AuA) - \#(\mathbf{L}(S) \cap uA) - \#(\mathbf{L}(S) \cap uA) + 1$$

(bilateral order)

### Theorem (Cassaigne, 1997)

$$p_S(n+2) - 2p_S(n+1) + p_S(n) = \sum_{|u|=n} m_S(u).$$

## RECURSIVE EXPRESSIONS FOR FACTOR COMPLEXITY

$$p_S(n) = \#\{u \in \mathbf{L}(S) : |u| = n\}$$

$$m_S(u) = \#(\mathbf{L}(S) \cap AuA) - \#(\mathbf{L}(S) \cap uA) - \#(\mathbf{L}(S) \cap uA) + 1$$

**(bilateral order)**

### Theorem (Cassaigne, 1997)

$$p_S(n+2) - 2p_S(n+1) + p_S(n) = \sum_{|u|=n} m_S(u).$$

- $m_S(u) = 0$  when  $u$  is not bispecial.

$$S = (\{a, b\}, \quad \varphi: a \mapsto ab, b \mapsto ba, \quad a)$$

## Theorem (Luca & Mione, 1994)

Every bispecial factor  $u \in \mathbf{L}(S)$  of length  $\geq 2$  is of the form

$$\varphi^n(ab), \quad \varphi^n(ba), \quad \varphi^n(aba), \quad \varphi^n(bab),$$

$$S = (\{a, b\}, \quad \varphi: a \mapsto ab, b \mapsto ba, \quad a)$$

## Theorem (Luca & Mione, 1994)

Every bispecial factor  $u \in \mathbf{L}(S)$  of length  $\geq 2$  is of the form

$$\varphi^n(ab), \quad \varphi^n(ba), \quad \varphi^n(aba), \quad \varphi^n(bab),$$

## Corollary (Brlek, 1989; Cassaigne, 1997)

In the Thue–Morse system the values of  $p_S(n)$  for  $n \geq 3$  are

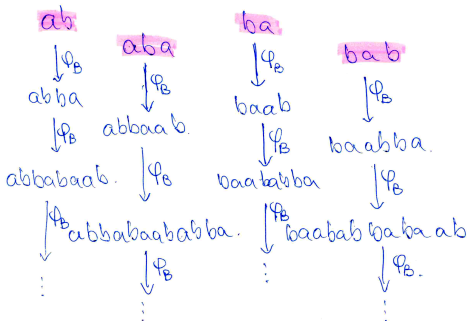
$$p_S(n) = \begin{cases} 4n - 2 \cdot 2^m - 4 & \text{if } 2 \cdot 2^n < n \leq 3 \cdot 2^m \\ 2n + 4 \cdot 2^m - 2 & \text{if } 3 \cdot 2^n < n \leq 4 \cdot 2^m \end{cases}$$

$$S = (\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$$

- Every bispecial factor is of the form  $\varphi_B^n(a)$  where  $\varphi_B(u) = \varphi(u)a$ .
- The complexity is  $p_S(n) = n + 1$  (Sturmian system).

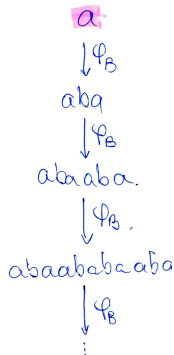
THUE-MORSE

$$\varphi_B(u) = \varphi(u)$$



FIBONACCI

$$\varphi_B(u) = \varphi(u)a$$









## Kluda, 2012

Under certain **key conditions**, there exists a finite set of initial bispecial words and a finitely determined rule from which all other bispecial factors are obtained.

## Kluda, 2012

Under certain **key conditions**, there exists a finite set of initial bispecial words and a finitely determined rule from which all other bispecial factors are obtained.

- These conditions also imply that the set of initial bispecial words and the rule can both be **calculated algorithmically**.

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

**NON-PUSHY** There are only finitely many  $u \in \mathbf{L}(S)$  such that  $\lim_{n \rightarrow \infty} |\varphi^n(u)| < \infty$ .

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

**NON-PUSHY** There are only finitely many  $u \in \mathbf{L}(S)$  such that  $\lim_{n \rightarrow \infty} |\varphi^n(u)| < \infty$ .

**EXTENDABLE** For all  $u \in \mathbf{L}(S)$ , there are letters  $a, b \in A$  such that  $aub \in \mathbf{L}(S)$ .

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

**NON-PUSHY** There are only finitely many  $u \in \mathbf{L}(S)$  such that  $\lim_{n \rightarrow \infty} |\varphi^n(u)| < \infty$ .

**EXTENDABLE** For all  $u \in \mathbf{L}(S)$ , there are letters  $a, b \in A$  such that  $aub \in \mathbf{L}(S)$ .

**INJECTIVE**  $\varphi$  is injective on  $\mathbf{L}(S)$ .

$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

**NON-PUSHY** There are only finitely many  $u \in \mathbf{L}(S)$  such that  $\lim_{n \rightarrow \infty} |\varphi^n(u)| < \infty$ .

**EXTENDABLE** For all  $u \in \mathbf{L}(S)$ , there are letters  $a, b \in A$  such that  $aub \in \mathbf{L}(S)$ .

**INJECTIVE**  $\varphi$  is injective on  $\mathbf{L}(S)$ . (**can be weakened!**)



$$S = (A, \varphi: A^* \rightarrow A^*, w \in A^*)$$

**PROPAGATING** For all  $a \in A$ ,  $\varphi(a) \neq \varepsilon$  (non-erasing).

**NON-PUSHY** There are only finitely many  $u \in \mathbf{L}(S)$  such that  $\lim_{n \rightarrow \infty} |\varphi^n(u)| < \infty$ .

**EXTENDABLE** For all  $u \in \mathbf{L}(S)$ , there are letters  $a, b \in A$  such that  $aub \in \mathbf{L}(S)$ .

**INJECTIVE**  $\varphi$  is injective on  $\mathbf{L}(S)$ . (**can be weakened!**)

**CIRCULAR** More on this later..

Part 3

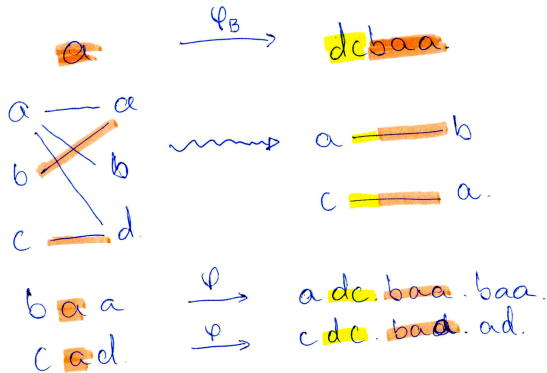


**Core ideas**

# Bispecial production rule



$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$

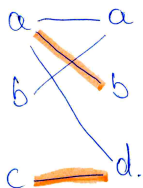


# Bispecial production rule



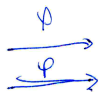
$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$

a



?

a a b  
c a d.



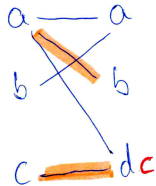
baa. baa. adc.  
cdc. baad. ad.

# Bispecial production rule



$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, \quad c)$

~~a~~



?

$a \quad \underline{a} \quad b$   
 $c \quad \underline{a} \quad d \quad c$

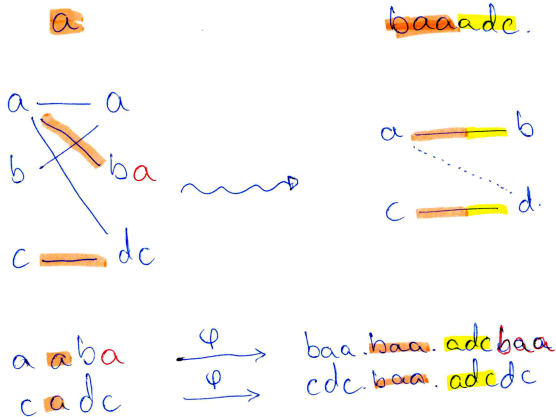
$\xrightarrow{\varphi}$

$baa. \underline{baa}. \underline{adc}.$   
 $cdc. \underline{baa}. \underline{adcdc}.$

# Bispecial production rule



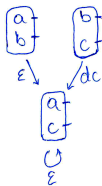
$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$



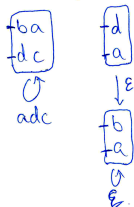
$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, c)$

$\varphi_B(u)$  depends on admissible left and right extensions of  $u$  in  $S$ .

Left:



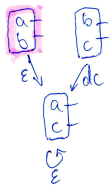
Right:



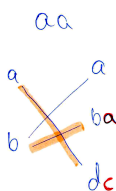
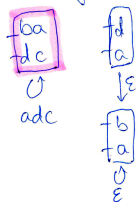
$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, \epsilon)$

$\varphi_B(u)$  depends on admissible left and right extensions of  $u$  in  $S$ .

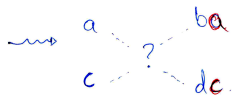
Left



Right



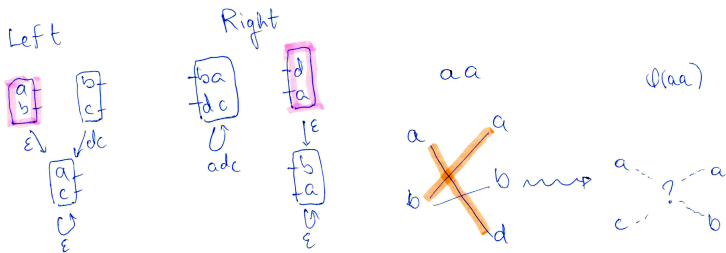
$\varphi(aa)adc.$





$S = (\{a, b, c, d\}, \varphi: a \mapsto baa, b \mapsto adc, c \mapsto cdc, d \mapsto ad, \epsilon)$

$\varphi_B(u)$  depends on admissible left and right extensions of  $u$  in  $S$ .



## Theorem (Klouta, 2012)

Under the **key conditions**<sup>2</sup>:

1.  $S$  has forky graphs which can be effectively computed.
2. There is a computable constant  $N$  such that any bispecial factor longer than  $N$  is the image by  $\varphi_B$  of a shorter one.

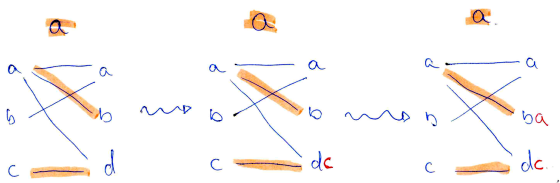
---

<sup>2</sup>Propagating, non-pushy, extendable, injective, circular

## Theorem (Klouta, 2012)

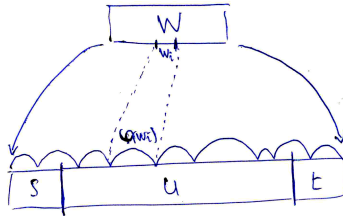
Under the **key conditions**<sup>2</sup>:

1.  $S$  has forky graphs which can be effectively computed.
  2. There is a computable constant  $N$  such that any bispecial factor longer than  $N$  is the image by  $\varphi_B$  of a shorter one.
- Main difficulty: make sure the extension process terminates.

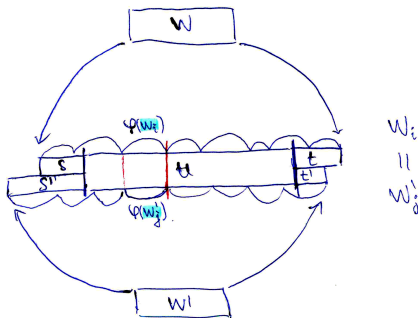


<sup>2</sup>Propagating, non-pushy, extendable, injective, circular

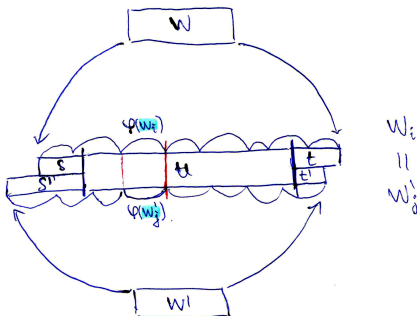
## INTERPRETATION



## SYNCHRONIZED INTERPRETATIONS



## SYNCHRONIZED INTERPRETATIONS

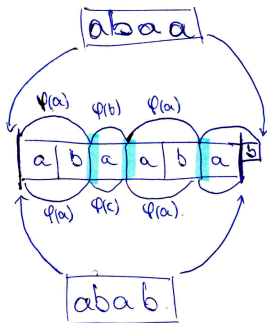


**CIRCULAR** Interpretations **synchronize** after a **bounded delay**<sup>3</sup>.

<sup>3</sup>using definition of Klouda & Starosta, 2019

## FIBONACCI

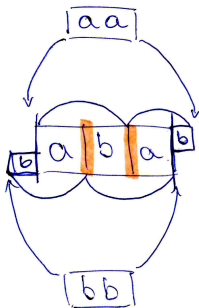
$$S = (\{a, b\}, \quad \varphi: a \mapsto ab, b \mapsto a, \quad a)$$



Synchronization delay is 0.

## THUE-MORSE

$$S = (\{a, b\}, \varphi: a \mapsto ab, b \mapsto ba, a)$$

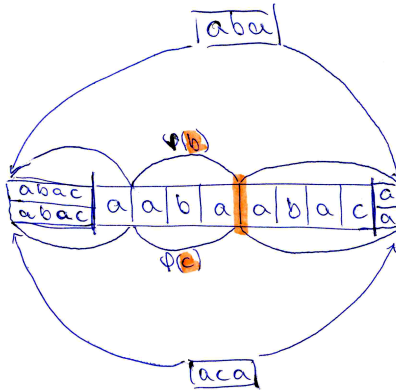


Synchronization delay is 1.



## TERNARY SYSTEM

$$S = (\{a, b, c\}, \varphi: a \mapsto abaca, b \mapsto aba, c \mapsto aba, a)$$

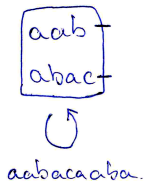


Synchronization delay is 9.

$$S = (\{a, b, c\}, \varphi: a \mapsto abaca, b \mapsto aba, c \mapsto aba, a)$$

This is a non-injective systems which **has computable forky graphs** if we ignore the bispecial factor  $a$  (of length 1).

Left.



Right.



$$S = (\{a, b, c\}, \varphi: a \mapsto \text{acabaca}, b \mapsto \text{aba}, c \mapsto \text{aba}, a)$$

This is a non-injective systems which **does not have computable forky graphs**. It satisfies all key conditions except for injectivity.



1. Decide the **key conditions** algorithmically.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.
  - 1.4 **Injective**: unclear, use Sardinas–Paterson as proxy.



1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.
  - 1.4 **Injective**: unclear, use Sardinas–Paterson as proxy.
  - 1.5 **Circular**: Klouda & Starosta, 2019.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.
  - 1.4 **Injective**: unclear, use Sardinas–Paterson as proxy.
  - 1.5 **Circular**: Klouda & Starosta, 2019.
2. Calculate the **minimal delay** for circularity.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.
  - 1.4 **Injective**: unclear, use Sardinas–Paterson as proxy.
  - 1.5 **Circular**: Klouda & Starosta, 2019.
2. Calculate the **minimal delay** for circularity.
3. Calculate reasonable **forky graphs**.

1. Decide the **key conditions** algorithmically.
  - 1.1 **Propagating**: trivial.
  - 1.2 **Non-pushy**: Ehrenfeucht & Rozenberg, 1983.
  - 1.3 **Extendable**: simple.
  - 1.4 **Injective**: unclear, use Sardinas–Paterson as proxy.
  - 1.5 **Circular**: Klouda & Starosta, 2019.
2. Calculate the **minimal delay** for circularity.
3. Calculate reasonable **forky graphs**.
4. Calculate an **initial set** of bispecial factors.

1. Package in a new SageMath class **DFOL**.

1. Package in a new SageMath class **DFOL**.
2. Web server for easy access.

1. Package in a new SageMath class **DFOL**.
2. Web server for easy access.
3. Decide (eventual) injectivity on the language.

1. Package in a new SageMath class **DFOL**.
2. Web server for easy access.
3. Decide (eventual) injectivity on the language.
4. Beyond injective, extendable, non-erasing.



1. Package in a new SageMath class **DFOL**.
2. Web server for easy access.
3. Decide (eventual) injectivity on the language.
4. Beyond injective, extendable, non-erasing.
5. Finer structure of bispecial factors.

