S-adic characterization of dendric languages: ternary case

France Gheeraert
Joint work with Marie Lejeune and Julien Leroy

November 30, 2021

Notations

- uniformly recurrent (= unif. rec.) languages on these alphabets: $\mathcal{L}$, $\mathcal{L}'$, $\mathcal{L}_N$, ...
- image of a $\mathcal{L}$ under σ : $\sigma^f(\mathcal{L}) = \mathrm{Fac}(\sigma(\mathcal{L}))$

Introduction

S-adic representations

Definition

A primitive *S-adic representation* of a unif. rec. language $\mathcal{L}$ is a primitive sequence of morphisms $(\sigma_n : \mathcal{A}_{n+1}^* \to \mathcal{A}_n^*)_n$ such that

$$\mathcal{L} = \bigcup_N \mathrm{Fac}(\sigma_0 \dots \sigma_N(\mathcal{A}_{N+1})).$$

A sequence $(\sigma_n : \mathcal{A}_{n+1}^* \to \mathcal{A}_n^*)_n$ is *primitive* if, for all N, there exists $m \geq 0$ such that, for all $a \in \mathcal{A}_{N+m+1}$, $\sigma_N \dots \sigma_{N+m}(a)$ contains all the letters of $\mathcal{A}_N$.

S-adic characterization

Question: For a given family $\mathcal{F}$ of languages, can we find a condition C such that

$$\mathcal{L} \in \mathcal{F} \text{ iff } \mathcal{L} \text{ has an } S\text{-adic representation satisfying } C?$$

S-adic characterization

Question: For a given family $\mathcal{F}$ of languages, can we find a condition C such that

$$\mathcal{L} \in \mathcal{F} \text{ iff } \mathcal{L} \text{ has an } S\text{-adic representation satisfying } C?$$

- Sturmian languages [Morse-Hedlund]: (non eventually constant) sequences over two given morphisms

S-adic characterization

Question: For a given family $\mathcal{F}$ of languages, can we find a condition C such that

$$\mathcal{L} \in \mathcal{F} \text{ iff } \mathcal{L} \text{ has an } S\text{-adic representation satisfying } C?$$

- Sturmian languages [Morse-Hedlund]: (non eventually constant) sequences over two given morphisms
- Arnoux-Rauzy languages [Arnoux-Rauzy]
- Episturmian languages [Justin-Pirillo]
- Linearly recurrent languages [Durand]
- Languages such that $p(n+1) - p(n) \leq 2$ [Leroy]
- ...

Extension graphs

$$LE_{\mathcal{L}}(w) = \{a \in \mathcal{A} \mid aw \in \mathcal{L}\}, \quad RE_{\mathcal{L}}(w) = \{b \in \mathcal{A} \mid wb \in \mathcal{L}\},$$
$$E_{\mathcal{L}}(w) = \{(a, b) \in LE_{\mathcal{L}}(w) \times RE_{\mathcal{L}}(w) \mid awb \in \mathcal{L}\}$$

Extension graphs

$$LE_{\mathcal{L}}(w) = \{a \in \mathcal{A} \mid aw \in \mathcal{L}\}, \quad RE_{\mathcal{L}}(w) = \{b \in \mathcal{A} \mid wb \in \mathcal{L}\},$$
$$E_{\mathcal{L}}(w) = \{(a, b) \in LE_{\mathcal{L}}(w) \times RE_{\mathcal{L}}(w) \mid awb \in \mathcal{L}\}$$

Definition

The *extension graph* of $w \in \mathcal{L}$ is the bipartite graph $\mathcal{E}_{\mathcal{L}}(w)$ with vertices $LE_{\mathcal{L}}(w) \sqcup RE_{\mathcal{L}}(w)$ and edges $E_{\mathcal{L}}(w)$.

Extension graphs

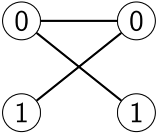
$$LE_{\mathcal{L}}(w) = \{a \in \mathcal{A} \mid aw \in \mathcal{L}\}, \quad RE_{\mathcal{L}}(w) = \{b \in \mathcal{A} \mid wb \in \mathcal{L}\},$$
$$E_{\mathcal{L}}(w) = \{(a, b) \in LE_{\mathcal{L}}(w) \times RE_{\mathcal{L}}(w) \mid awb \in \mathcal{L}\}$$

Definition

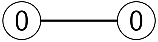
The *extension graph* of $w \in \mathcal{L}$ is the bipartite graph $\mathcal{E}_{\mathcal{L}}(w)$ with vertices $LE_{\mathcal{L}}(w) \sqcup RE_{\mathcal{L}}(w)$ and edges $E_{\mathcal{L}}(w)$.

If $\mathcal{L}$ is the Fibonacci language,

$\mathcal{E}_{\mathcal{L}}(\varepsilon)$



$\mathcal{E}_{\mathcal{L}}(1)$



Dendric languages

Definition (Berthé, De Felice, Dolce, Leroy, Perrin, Reutenauer, Rindone)

A word $w \in \mathcal{L}$ is *dendric* if its extension graph $\mathcal{E}_{\mathcal{L}}(w)$ is a tree.

A language $\mathcal{L}$ is *dendric* if all the words $w \in \mathcal{L}$ are.

Dendric languages

Definition (Berthé, De Felice, Dolce, Leroy, Perrin, Reutenauer, Rindone)

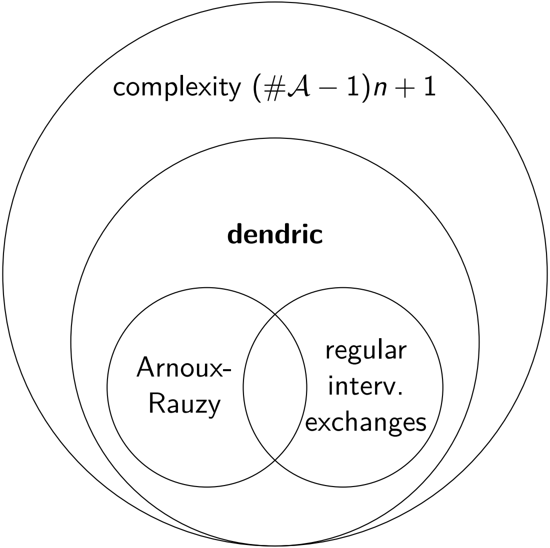
A word $w \in \mathcal{L}$ is *dendric* if its extension graph $\mathcal{E}_{\mathcal{L}}(w)$ is a tree.

A language $\mathcal{L}$ is *dendric* if all the words $w \in \mathcal{L}$ are.

Definition (Dolce, Perrin)

A language $\mathcal{L}$ is *eventually dendric* if there exists n such that all the words $w \in \mathcal{L}_{\geq n}$ are dendric.

Relation with other families



Return words

Definition

A *return word* for $w \neq \varepsilon$ in $\mathcal{L}$ is a word u such that

$$uw \in \mathcal{L}, \quad |uw|_w = 2, \quad uw \in w\mathcal{A}^*.$$

The set of return words for w is denoted $\mathcal{R}_{\mathcal{L}}(w)$.

Return words

Definition

A *return word* for $w \neq \varepsilon$ in $\mathcal{L}$ is a word u such that

$$uw \in \mathcal{L}, \quad |uw|_w = 2, \quad uw \in w\mathcal{A}^*.$$

The set of return words for w is denoted $\mathcal{R}_{\mathcal{L}}(w)$.

In the Fibonacci language,

$$\mathcal{R}_{\mathcal{L}}(0) = \{0, 01\}, \quad \mathcal{R}_{\mathcal{L}}(1) = \{10, 100\}.$$

Return words

Definition

A *return word* for $w \neq \varepsilon$ in $\mathcal{L}$ is a word u such that

$$uw \in \mathcal{L}, \quad |uw|_w = 2, \quad uw \in w\mathcal{A}^*.$$

The set of return words for w is denoted $\mathcal{R}_\mathcal{L}(w)$.

In the Fibonacci language,

$$\mathcal{R}_\mathcal{L}(0) = \{0, 01\}, \quad \mathcal{R}_\mathcal{L}(1) = \{10, 100\}.$$

Theorem (Balková, Pelantová, Steiner)

Let $\mathcal{L}$ be a unif. rec. dendric language. For all non empty $w \in \mathcal{L}$,

$$\#\mathcal{R}_\mathcal{L}(w) = \#\mathcal{A}.$$

Derived language of a dendric language

Definition

The *derived language* of $\mathcal{L}$ with respect to $w \neq \varepsilon$ is the language

$$\mathcal{L}' = \{u \in \mathcal{B}^* \mid \sigma(u)w \in \mathcal{L}\}$$

where $\sigma : \mathcal{B}^* \to \mathcal{A}^*$ is such that $\sigma(\mathcal{B}) = \mathcal{R}_{\mathcal{L}}(w)$. Then

$$\mathcal{L} = \sigma^f(\mathcal{L}').$$

Derived language of a dendric language

Definition

The *derived language* of $\mathcal{L}$ with respect to $w \neq \varepsilon$ is the language

$$\mathcal{L}' = \{u \in \mathcal{B}^* \mid \sigma(u)w \in \mathcal{L}\}$$

where $\sigma : \mathcal{B}^* \to \mathcal{A}^*$ is such that $\sigma(\mathcal{B}) = \mathcal{R}_{\mathcal{L}}(w)$. Then

$$\mathcal{L} = \sigma^f(\mathcal{L}').$$

Theorem (Berthé *et al.*)

The derived language of a unif. rec. dendric language with respect to any word is a unif. rec. dendric language.

Construction of S-adic representations

We can build primitive S-adic representations of a unif. rec. dendric language $\mathcal{L} = \mathcal{L}_0 \subseteq \mathcal{A}^*$ in the following way:

1. pick a non empty word $w_0 \in \mathcal{L}_0$;
2. define $\mathcal{L}_1 \subseteq \mathcal{A}^*$ as the derived language of $\mathcal{L}_0$ with respect to w;
3. denote $\sigma_0 : \mathcal{A}^* \to \mathcal{A}^*$ the associated morphism, i.e. such that $\mathcal{L}_0 = \sigma_0^f(\mathcal{L}_1)$;
4. go back to step 1 with $\mathcal{L}_1$ to define $\mathcal{L}_2$ and σ_1, and so on.

Construction of S-adic representations

We can build primitive S-adic representations of a unif. rec. dendric language $\mathcal{L} = \mathcal{L}_0 \subseteq \mathcal{A}^*$ in the following way:

1. pick a non empty word $w_0 \in \mathcal{L}_0$;
2. define $\mathcal{L}_1 \subseteq \mathcal{A}^*$ as the derived language of $\mathcal{L}_0$ with respect to w;
3. denote $\sigma_0 : \mathcal{A}^* \to \mathcal{A}^*$ the associated morphism, i.e. such that $\mathcal{L}_0 = \sigma_0^f(\mathcal{L}_1)$;
4. go back to step 1 with $\mathcal{L}_1$ to define $\mathcal{L}_2$ and σ_1, and so on.

$$\mathcal{L} = \sigma_0^f(\mathcal{L}_1) = \sigma_0^f(\sigma_1^f(\mathcal{L}_2)) = \dots$$

Return morphisms

Definition

A *return morphism* for $w \neq \varepsilon$ is an injective morphism $\sigma : \mathcal{A}^* \to \mathcal{B}^*$ such that, for all $a \in \mathcal{A}$,

$$|\sigma(a)w|_w = 2, \quad \sigma(a)w \in w\mathcal{B}^*.$$

For any language $\mathcal{L}$, $\sigma(\mathcal{A})$ is the set of return words for w in $\sigma^f(\mathcal{L})$.

Return morphisms

Definition

A *return morphism* for $w \neq \varepsilon$ is an injective morphism $\sigma : \mathcal{A}^* \to \mathcal{B}^*$ such that, for all $a \in \mathcal{A}$,

$$|\sigma(a)w|_w = 2, \quad \sigma(a)w \in w\mathcal{B}^*.$$

For any language $\mathcal{L}$, $\sigma(\mathcal{A})$ is the set of return words for w in $\sigma^f(\mathcal{L})$.

$$\sigma : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 021 \\ 2 \mapsto 022221 \end{cases} \qquad \tau : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 010 \\ 2 \mapsto 010210 \end{cases}$$

Return morphisms

Definition

A *return morphism* for $w \neq \varepsilon$ is an injective morphism $\sigma : \mathcal{A}^* \to \mathcal{B}^*$ such that, for all $a \in \mathcal{A}$,

$$|\sigma(a)w|_w = 2, \quad \sigma(a)w \in w\mathcal{B}^*.$$

For any language $\mathcal{L}$, $\sigma(\mathcal{A})$ is the set of return words for w in $\sigma^f(\mathcal{L})$.

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases} \qquad \tau : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 010 \\ 2 \mapsto 010210 \end{cases}$$

Return morphisms

Definition

A *return morphism* for $w \neq \varepsilon$ is an injective morphism $\sigma : \mathcal{A}^* \to \mathcal{B}^*$ such that, for all $a \in \mathcal{A}$,

$$|\sigma(a)w|_w = 2, \quad \sigma(a)w \in w\mathcal{B}^*.$$

For any language $\mathcal{L}$, $\sigma(\mathcal{A})$ is the set of return words for w in $\sigma^f(\mathcal{L})$.

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases} \qquad \tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

Return morphisms

Definition

A *return morphism* for $w \neq \varepsilon$ is an injective morphism $\sigma : \mathcal{A}^* \to \mathcal{B}^*$ such that, for all $a \in \mathcal{A}$,

$$|\sigma(a)w|_w = 2, \quad \sigma(a)w \in w\mathcal{B}^*.$$

For any language $\mathcal{L}$, $\sigma(\mathcal{A})$ is the set of return words for w in $\sigma^f(\mathcal{L})$.

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases} \qquad \tau : \begin{cases} 0 \mapsto 01010 \\ 1 \mapsto 010010 \\ 2 \mapsto 010210010 \end{cases}$$

Dendric images under return morphism

Dendric images: goal

Given an unif. rec. dendric language $\mathcal{L}$ and a return morphism σ for w, when is $\sigma^f(\mathcal{L})$ (unif. rec.) dendric?

Dendric images: goal

Given an unif. rec. dendric language $\mathcal{L}$ and a return morphism σ for w, when is $\sigma^f(\mathcal{L})$ (unif. rec.) dendric?

$\rightarrow$ What can we say about $\mathcal{E}_{\sigma^f(\mathcal{L})}(u)$?

Dendric images: goal

Given an unif. rec. dendric language $\mathcal{L}$ and a return morphism σ for w, when is $\sigma^f(\mathcal{L})$ (unif. rec.) dendric?

$\rightarrow$ What can we say about $\mathcal{E}_{\sigma^f(\mathcal{L})}(u)$?

Two cases:

- $|u|_w = 0$: u is an *initial factor*;
- $|u|_w > 0$: u is an *extended image*.

Initial factors and dendric morphisms

If u is an initial factor, then each occurrence of u is as an internal factor of some $\sigma(\alpha)w$, $\alpha \in \mathcal{A}$.

Initial factors and dendric morphisms

If u is an initial factor, then each occurrence of u is as an internal factor of some $\sigma(\alpha)w$, $\alpha \in \mathcal{A}$.
Thus

$$aub \in \sigma^f(\mathcal{L}) \Leftrightarrow \exists \alpha \in \mathcal{A} \text{ st. } aub \in \mathrm{Fac}(\sigma(\alpha)w).$$

Initial factors and dendric morphisms

If u is an initial factor, then each occurrence of u is as an internal factor of some $\sigma(\alpha)w$, $\alpha \in \mathcal{A}$.
Thus

$$aub \in \sigma^f(\mathcal{L}) \Leftrightarrow \exists \alpha \in \mathcal{A} \text{ st. } aub \in \mathrm{Fac}(\sigma(\alpha)w).$$

In other words,

$$\mathcal{E}_{\sigma^f(\mathcal{L})}(u) = \mathcal{E}_{F_\sigma}(u).$$

where

$$F_\sigma = \bigcup_{\alpha \in \mathcal{A}} \mathrm{Fac}(\sigma(\alpha)w).$$

Initial factors and dendric morphisms

If u is an initial factor, then each occurrence of u is as an internal factor of some $\sigma(\alpha)w$, $\alpha \in \mathcal{A}$.
Thus

$$aub \in \sigma^f(\mathcal{L}) \Leftrightarrow \exists \alpha \in \mathcal{A} \text{ st. } aub \in \mathrm{Fac}(\sigma(\alpha)w).$$

In other words,

$$\mathcal{E}_{\sigma^f(\mathcal{L})}(u) = \mathcal{E}_{F_\sigma}(u).$$

where

$$F_\sigma = \bigcup_{\alpha \in \mathcal{A}} \mathrm{Fac}(\sigma(\alpha)w).$$

Definition

A return morphism σ for w is *dendric* if, for all $u \in F_\sigma$ such that $|u|_w = 0$, u is dendric in F_σ.

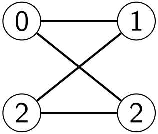
Examples

$$\sigma : \begin{cases} 0 \mapsto 01\textcolor{gray}{0} \\ 1 \mapsto 021\textcolor{gray}{0} \\ 2 \mapsto 022221\textcolor{gray}{0} \end{cases}$$

Examples

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases}$$

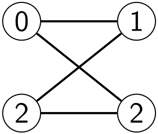
$\mathcal{E}_{F_\sigma}(2)$



Examples

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases}$$

$\mathcal{E}_{F_\sigma}(2)$

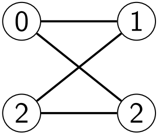


$\Rightarrow$ σ is not dendric

Examples

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases}$$

$\mathcal{E}_{F_\sigma}(2)$



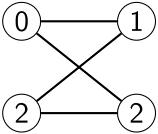
$\Rightarrow \sigma$ is not dendric

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 010210001 \end{cases}$$

Examples

$$\sigma : \begin{cases} 0 \mapsto 010 \\ 1 \mapsto 0210 \\ 2 \mapsto 0222210 \end{cases}$$

$\mathcal{E}_{F_\sigma}(2)$



$\Rightarrow$ σ is not dendric

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$\mathcal{E}_{F_\sigma}(\varepsilon)$, $\mathcal{E}_{F_\sigma}(0)$ and $\mathcal{E}_{F_\sigma}(10)$ are trees

$\Rightarrow$ τ is dendric

Extended images

Proposition (G., Lejeune, Leroy)

If $u \in \sigma^f(\mathcal{L})$ is an extended image, there exist unique $s, p \in \mathcal{A}^*$, $v \in \mathcal{L}$ such that

- $u = s\sigma(v)p$,
- *s is a proper suffix of an element of $\sigma(\mathcal{A})$,*
- *$p \in w\mathcal{B}^*$ is a proper prefix of an element of $\sigma(\mathcal{A})w$.*

We will then specify that u is an extended image <u>of v</u> (under σ).

Extended images

Proposition (G., Lejeune, Leroy)

If $u \in \sigma^f(\mathcal{L})$ is an extended image, there exist unique $s, p \in \mathcal{A}^*$, $v \in \mathcal{L}$ such that

- $u = s\sigma(v)p$,
- *s is a proper suffix of an element of $\sigma(\mathcal{A})$,*
- *$p \in w\mathcal{B}^*$ is a proper prefix of an element of $\sigma(\mathcal{A})w$.*

$\Rightarrow$ *Every occurrence of u is as an internal factor of some $\sigma(\alpha v \beta)w$*

We will then specify that u is an extended image <u>of v</u> (under σ).

Extended images

Proposition (G., Lejeune, Leroy)

If $u \in \sigma^f(\mathcal{L})$ is an extended image, there exist unique $s, p \in \mathcal{A}^*$, $v \in \mathcal{L}$ such that

- $u = s\sigma(v)p$,
- *s is a proper suffix of an element of $\sigma(\mathcal{A})$,*
- *$p \in w\mathcal{B}^*$ is a proper prefix of an element of $\sigma(\mathcal{A})w$.*

$\Rightarrow$ *Every occurrence of u is as an internal factor of some $\sigma(\alpha v \beta)w$*

Moreover, $(a, b) \in E_{\sigma^f(\mathcal{L})}(u)$ if and only if

$$\exists (\alpha, \beta) \in E_{\mathcal{L}}(v) \text{ st. } \sigma(\alpha) \in \mathcal{B}^* as \text{ and } \sigma(\beta)w \in pb\mathcal{B}^*.$$

We will then specify that u is an extended image <u>of v</u> (under σ).

Example

$$\boxed{(a, b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha, \beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta) w \in pb\mathcal{A}^*}$$

Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

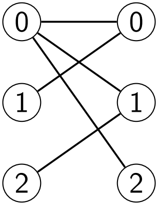
$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$\mathcal{E}_{\mathcal{L}}(v)$$



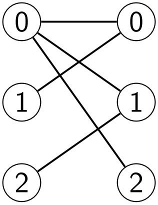
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta) w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$



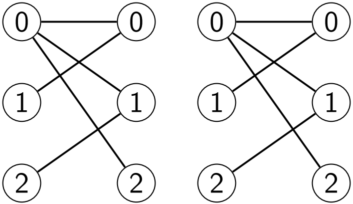
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$



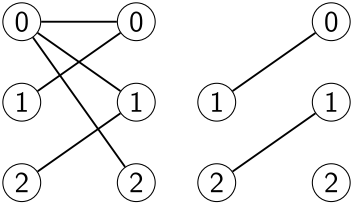
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta) w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\; p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$



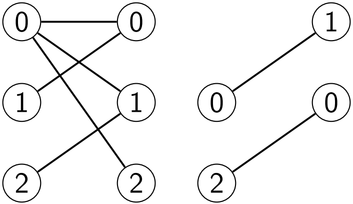
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10, \; p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$ $\mathcal{E}_{\tau^f(\mathcal{L})}(u)$



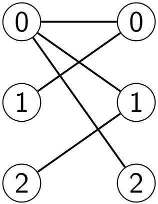
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

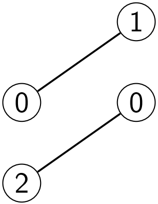
$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 010210 01 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$
$$u' = \sigma(v)010 \longrightarrow s = \varepsilon,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$



$\mathcal{E}_{\tau^f(\mathcal{L})}(u)$



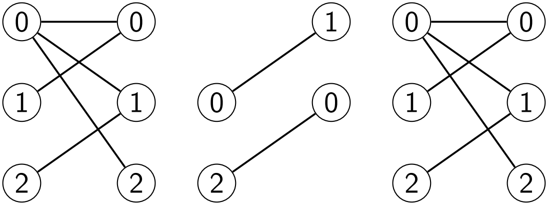
Example

$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$
$$u' = \sigma(v)010 \longrightarrow s = \varepsilon,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$ $\mathcal{E}_{\tau^f(\mathcal{L})}(u)$



Example

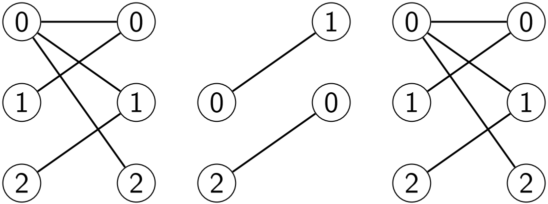
$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta) w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$

$$u' = \sigma(v)010 \longrightarrow s = \varepsilon,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$ $\mathcal{E}_{\tau^f(\mathcal{L})}(u)$



Example

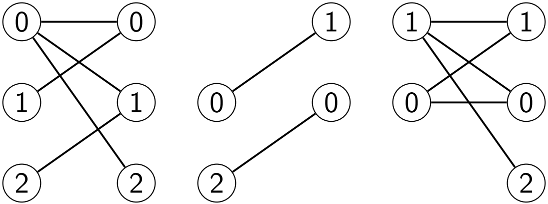
$$\boxed{(a,b) \in E_{\sigma^f(\mathcal{L})}(u) \Leftrightarrow \exists(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{A}^* as \wedge \sigma(\beta)w \in pb\mathcal{A}^*}$$

$$\tau : \begin{cases} 0 \mapsto 0101 \\ 1 \mapsto 01001 \\ 2 \mapsto 01021001 \end{cases}$$

$$u = 10\sigma(v)010 \longrightarrow s = 10,\ p = 010$$

$$u' = \sigma(v)010 \longrightarrow s = \varepsilon,\ p = 010$$

$\mathcal{E}_{\mathcal{L}}(v)$ $\mathcal{E}_{\tau^f(\mathcal{L})}(u)$ $\mathcal{E}_{\tau^f(\mathcal{L})}(u')$



Dendric extended images

$\mathcal{E}_{\mathcal{L},s,p}(v)$ is the subgraph of $\mathcal{E}_{\mathcal{L}}(v)$ generated by the edges

$$\{(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{B}^+ s \text{ and } \sigma(\beta) w \in p\mathcal{B}^+\}$$

i.e. the graph obtained after the first step.

Dendric extended images

$\mathcal{E}_{\mathcal{L},s,p}(v)$ is the subgraph of $\mathcal{E}_{\mathcal{L}}(v)$ generated by the edges

$$\{(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{B}^+ s \text{ and } \sigma(\beta)w \in p\mathcal{B}^+\}$$

i.e. the graph obtained after the first step.

Theorem (G., Lejeune, Leroy)

If $v \in \mathcal{L}$ is dendric, then the following are equivalent:

1. *all the extended images of v are dendric (in $\sigma^f(\mathcal{L})$);*
2. *for all $s, p \in \mathcal{B}^*$, the graph $\mathcal{E}_{\mathcal{L},s,p}(v)$ is connected;*

Dendric extended images

$\mathcal{E}_{\mathcal{L},s,p}(v)$ is the subgraph of $\mathcal{E}_{\mathcal{L}}(v)$ generated by the edges

$$\{(\alpha,\beta) \in E_{\mathcal{L}}(v) : \sigma(\alpha) \in \mathcal{B}^{+}s \text{ and } \sigma(\beta)w \in p\mathcal{B}^{+}\}$$

i.e. the graph obtained after the first step.

Theorem (G., Lejeune, Leroy)

If $v \in \mathcal{L}$ is dendric, then the following are equivalent:

1. *all the extended images of v are dendric (in $\sigma^{f}(\mathcal{L})$);*
2. *for all $s, p \in \mathcal{B}^{*}$, the graph $\mathcal{E}_{\mathcal{L},s,p}(v)$ is connected;*
3. *for all $s, p \in \mathcal{B}^{*}$, the graphs $\mathcal{E}_{\mathcal{L},s,\varepsilon}(v)$ and $\mathcal{E}_{\mathcal{L},\varepsilon,p}(v)$ are connected.*

Dendric images: result

Corollary

The image of a unif. rec. dendric language $\mathcal{L}$ under a return morphism σ is dendric if and only if σ is dendric and the conditions $\mathcal{C}^L(\sigma, \mathcal{L})$ and $\mathcal{C}^R(\sigma, \mathcal{L})$ are satisfied.

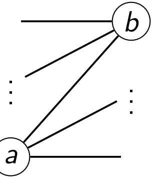
$$\mathcal{C}^L(\sigma, \mathcal{L}) \equiv \forall\, v \in \mathcal{L}, \forall\, s \in \mathcal{B}^*, \mathcal{E}_{\mathcal{L},s,\varepsilon}(v) \text{ is connected}$$
$$\mathcal{C}^R(\sigma, \mathcal{L}) \equiv \forall\, v \in \mathcal{L}, \forall\, p \in \mathcal{B}^*, \mathcal{E}_{\mathcal{L},\varepsilon,p}(v) \text{ is connected}$$

Arnoux-Rauzy languages

If there exist a and b such that

$$E_{\mathcal{L}}(v) = (a \times RE_{\mathcal{L}}(v)) \cup (LE_{\mathcal{L}}(v) \times b),$$

$\mathcal{E}_{\mathcal{L}}(v)$



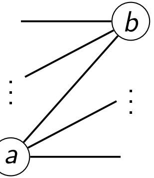
then the extended images of v are always dendric.

Arnoux-Rauzy languages

If there exist a and b such that

$$E_{\mathcal{L}}(v) = (a \times RE_{\mathcal{L}}(v)) \cup (LE_{\mathcal{L}}(v) \times b),$$

$\mathcal{E}_{\mathcal{L}}(v)$



then the extended images of v are always dendric.

Corollary

The image of an Arnoux-Rauzy under a return morphism is dendric if and only if the morphism is dendric.

Eventually dendric languages

Proposition (Dolce, Perrin)

A language $\mathcal{L}$ is eventually dendric if and only if there exists N such that all words of $\mathcal{L}_{\geq N}$ satisfy the condition of the previous slide.

Eventually dendric languages

Proposition (Dolce, Perrin)

A language $\mathcal{L}$ is eventually dendric if and only if there exists N such that all words of $\mathcal{L}_{\geq N}$ satisfy the condition of the previous slide.

Corollary

The image of an unif. rec. eventually dendric language under a return morphism is eventually dendric.

Eventually dendric languages

Proposition (Dolce, Perrin)

A language $\mathcal{L}$ is eventually dendric if and only if there exists N such that all words of $\mathcal{L}_{\geq N}$ satisfy the condition of the previous slide.

Corollary

The image of an unif. rec. eventually dendric language under a return morphism is eventually dendric.

Corollary

A language $\mathcal{L}$ is an unif. rec. eventually dendric language if and only if there exist a return morphism σ and an unif. rec. dendric language $\mathcal{L}'$ such that $\mathcal{L} = \sigma^f(\mathcal{L}')$.

Summary of what we obtained

Each unif. rec. dendric language $\mathcal{L}$ has a primitive S-adic representation $(\sigma_n)_n$ such that

1. for all N, σ_N is a dendric return morphism,
2. if $\mathcal{L}_{N+1}$ is the language with S-adic representation $(\sigma_n)_{n>N}$, then the conditions $\mathcal{C}^L(\sigma_N, \mathcal{L}_{N+1})$ and $\mathcal{C}^R(\sigma_N, \mathcal{L}_{N+1})$ are satisfied.

Summary of what we obtained

Each unif. rec. dendric language $\mathcal{L}$ has a primitive S-adic representation $(\sigma_n)_n$ such that

1. for all N, σ_N is a dendric return morphism,
2. if $\mathcal{L}_{N+1}$ is the language with S-adic representation $(\sigma_n)_{n>N}$, then the conditions $\mathcal{C}^L(\sigma_N, \mathcal{L}_{N+1})$ and $\mathcal{C}^R(\sigma_N, \mathcal{L}_{N+1})$ are satisfied.

Proposition

If $\mathcal{L}$ has a primitive S-adic representation $(\sigma_n)_n$ satisfying conditions 1 and 2 above, then $\mathcal{L}$ is unif. rec. dendric.

First (very) naive graph

Proposition

A language $\mathcal{L} \subseteq \mathcal{A}^*$ is unif. rec. dendric if and only if it has a primitive S-adic representation labeling a path in the graph defined as follows

- *each vertex corresponds to a (unif. rec.) language on $\mathcal{A}$;*
- *for each dendric return morphism $\sigma : \mathcal{A}^* \to \mathcal{A}^*$ and each language $\mathcal{L}$, there is an edge from $\sigma^f(\mathcal{L})$ to $\mathcal{L}$ if and only if conditions $\mathcal{C}^L(\sigma, \mathcal{L})$ and $\mathcal{C}^R(\sigma, \mathcal{L})$ are satisfied.*

Ternary case

Ternary case: goal

We work on the alphabet $\mathcal{A}_3 = \{1, 2, 3\}$.

To obtain a simpler description of the characterization, we look at

1. the vertices
2. the edges

Vertices: Goal

We want to associate an object $o(\mathcal{L})$ to each (unif. rec. dendric) language $\mathcal{L}$ such that

- conditions $\mathcal{C}^L(\sigma, \mathcal{L})$ and $\mathcal{C}^R(\sigma, \mathcal{L})$ only depend on σ and $o(\mathcal{L})$;
- if $o(\mathcal{L}) = o(\mathcal{L}')$, then $o(\sigma^f(\mathcal{L})) = o(\sigma^f(\mathcal{L}'))$.

Vertices: Goal

We want to associate an object $o(\mathcal{L}) = (o^L(\mathcal{L}), o^R(\mathcal{L}))$ to each (unif. rec. dendric) language $\mathcal{L}$ such that

- condition $\mathcal{C}^L(\sigma, \mathcal{L})$ (resp. $\mathcal{C}^R(\sigma, \mathcal{L})$) only depends on σ and $o^L(\mathcal{L})$ (resp. $o^R(\mathcal{L})$);
- if $o(\mathcal{L}) = o(\mathcal{L}')$, then $o(\sigma^f(\mathcal{L})) = o(\sigma^f(\mathcal{L}'))$.

We will only look at the left side for now.

Example

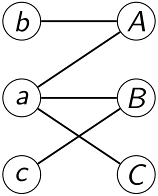
$$\mathcal{C}^L(\sigma, \mathcal{L}) \equiv \forall\, v \in \mathcal{L}, \forall\, s \in \mathcal{B}^*, \mathcal{E}_{\mathcal{L},s,\varepsilon}(v) \text{ is connected}$$

Example

$$\mathcal{C}^L(\sigma, \mathcal{L}) \equiv \forall\, v \in \mathcal{L}, \forall\, s \in \mathcal{B}^*, \mathcal{E}_{\mathcal{L},s,\varepsilon}(v) \text{ is connected}$$

Which sets of left vertices can we remove while keeping connected graphs?

$\mathcal{E}_{\mathcal{L}}(v)$

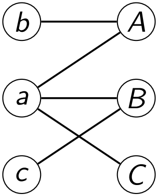


Example

$$\mathcal{C}^L(\sigma, \mathcal{L}) \equiv \forall\, v \in \mathcal{L}, \forall\, s \in \mathcal{B}^*, \mathcal{E}_{\mathcal{L},s,\varepsilon}(v) \text{ is connected}$$

Which sets of left vertices can we remove while keeping connected graphs?

$\mathcal{E}_{\mathcal{L}}(v)$



A letter is *(left-)problematic* if removing it on the left will disconnect some extension graph

Object $o^L(\mathcal{L})$

We define

$$o^L(\mathcal{L}) = \text{ set of left-problematic letters}$$

It is such that

- condition $\mathcal{C}^L(\sigma, \mathcal{L})$ only depends on σ and $o^L(\mathcal{L})$,
- if $o^L(\mathcal{L}) = o^L(\mathcal{L}')$, then $o^L(\sigma^f(\mathcal{L})) = o^L(\sigma^f(\mathcal{L}'))$.

Object $o^L(\mathcal{L})$

We define

$$o^L(\mathcal{L}) = \text{ set of left-problematic letters}$$

It is such that

- condition $\mathcal{C}^L(\sigma, \mathcal{L})$ only depends on σ and $o^L(\mathcal{L})$,
- if $o^L(\mathcal{L}) = o^L(\mathcal{L}')$, then $o^L(\sigma^f(\mathcal{L})) = o^L(\sigma^f(\mathcal{L}'))$.

Proposition

If $\mathcal{L}$ is a unif. rec. ternary dendric language, then there is at most one left-problematic letter.

New set of vertices

Definition

For $o = (o^L, o^R) \in \{\emptyset, \{1\}, \{2\}, \{3\}\}^2$, if $\mathcal{L}$ is such that $o = o(\mathcal{L})$, we can define

- $\mathcal{C}^L(\sigma, o) \equiv \mathcal{C}^L(\sigma, \mathcal{L})$
- $\mathcal{C}^R(\sigma, o) \equiv \mathcal{C}^R(\sigma, \mathcal{L})$
- $\sigma(o) = o(\sigma^f(\mathcal{L}))$

We obtain a new graph:

- the vertices are the elements of $\{\emptyset, \{1\}, \{2\}, \{3\}\}^2$;
- for each dendric return morphism $\sigma : \mathcal{A}^* \to \mathcal{A}^*$ and each vertex o, there is an edge from $\sigma(o)$ to o if and only if conditions $\mathcal{C}^L(\sigma, o)$ and $\mathcal{C}^R(\sigma, o)$ are satisfied.

Edges: Goal

We can build primitive S-adic representations of a unif. rec. dendric language $\mathcal{L} = \mathcal{L}_0 \subseteq \mathcal{A}^*$ in the following way:

1. pick a non empty word $w_0 \in \mathcal{L}_0$;
2. define $\mathcal{L}_1 \subseteq \mathcal{A}^*$ as the derived language of $\mathcal{L}_0$ with respect to w;
3. denote $\sigma_0 : \mathcal{A}^* \to \mathcal{A}^*$ the associated morphism, i.e. such that $\mathcal{L}_0 = \sigma_0^f(\mathcal{L}_1)$;
4. go back to step 1 with $\mathcal{L}_1$ to define $\mathcal{L}_2$ and σ_1, and so on.

Edges: Goal

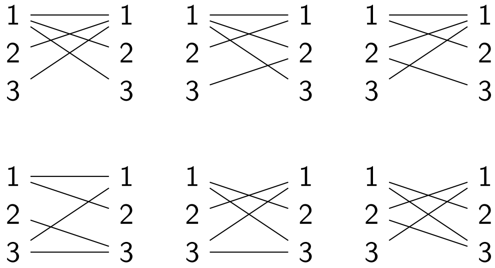
We can build primitive S-adic representations of a unif. rec. dendric language $\mathcal{L} = \mathcal{L}_0 \subseteq \mathcal{A}^*$ in the following way:

1. pick a non empty word $w_0 \in \mathcal{L}_0$;
2. define $\mathcal{L}_1 \subseteq \mathcal{A}^*$ as the derived language of $\mathcal{L}_0$ with respect to w;
3. denote $\sigma_0 : \mathcal{A}^* \to \mathcal{A}^*$ the associated morphism, i.e. such that $\mathcal{L}_0 = \sigma_0^f(\mathcal{L}_1)$;
4. go back to step 1 with $\mathcal{L}_1$ to define $\mathcal{L}_2$ and σ_1, and so on.

We pick w in a "clever" way to reduce the set of return morphisms that appear.

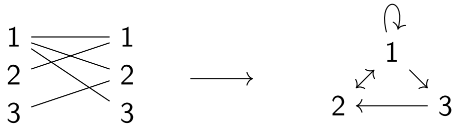
Possible extension graphs

The extension graph of ε in a unif. rec. dendric language is, up to a permutation, one of



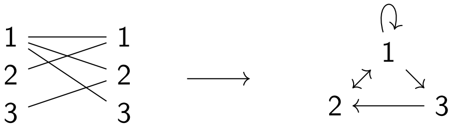
Finding the return words

From $\mathcal{E}_{\mathcal{L}}(\varepsilon)$, we build the Rauzy graph of order 1.



Finding the return words

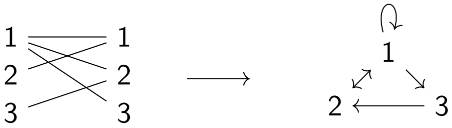
From $\mathcal{E}_{\mathcal{L}}(\varepsilon)$, we build the Rauzy graph of order 1.



The return words for 1 are among the paths from 1 to 1 in the Rauzy graph of order 1.

Finding the return words

From $\mathcal{E}_{\mathcal{L}}(\varepsilon)$, we build the Rauzy graph of order 1.



The return words for 1 are among the paths from 1 to 1 in the Rauzy graph of order 1.

$$\beta : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 132 \end{cases}$$

Set of morphisms

$$\alpha : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 13 \end{cases} \qquad \beta : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 132 \end{cases} \qquad \gamma : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 123 \end{cases}$$

$$\delta^{(k)} : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 123^k \\ 3 \mapsto 123^{k+1} \end{cases} \qquad \zeta^{(k)} : \begin{cases} 1 \mapsto 13^k \\ 2 \mapsto 12 \\ 3 \mapsto 13^{k+1} \end{cases} \qquad \eta : \begin{cases} 1 \mapsto 13 \\ 2 \mapsto 12 \\ 3 \mapsto 123 \end{cases}$$

Set of morphisms

$$\alpha : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 13 \end{cases} \qquad \beta : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 132 \end{cases} \qquad \gamma : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 12 \\ 3 \mapsto 123 \end{cases}$$

$$\delta^{(k)} : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 123^k \\ 3 \mapsto 123^{k+1} \end{cases} \qquad \zeta^{(k)} : \begin{cases} 1 \mapsto 13^k \\ 2 \mapsto 12 \\ 3 \mapsto 13^{k+1} \end{cases} \qquad \eta : \begin{cases} 1 \mapsto 13 \\ 2 \mapsto 12 \\ 3 \mapsto 123 \end{cases}$$

$$\mathcal{S}_3 = \{\alpha, \beta, \gamma, \eta\} \cup \{\delta^{(k)}, \zeta^{(k)} \mid k \geq 1\}$$

Simpler graph

Theorem (G., Lejeune, Leroy)

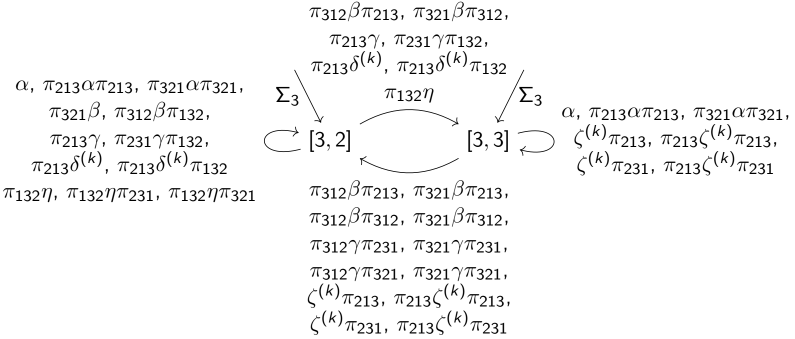
A language is unif. rec. ternary dendric if and only if it has a primitive S-adic representation labeling an infinite path in the graph defined as follows

- *the vertices are the elements of* $\{\emptyset, \{1\}, \{2\}, \{3\}\}^2$;
- *for each* $\sigma \in \Sigma_3 \mathcal{S}_3 \Sigma_3$ *and each vertex* o*, there is an edge from* $\sigma(o)$ *to* o *if and only if conditions* $\mathcal{C}^L(\sigma, o)$ *and* $\mathcal{C}^R(\sigma, o)$ *are satisfied.*

Even simpler graph

Theorem (G., Lejeune, Leroy)

A language is unif. rec. ternary dendric if and only if it has a primitive S-adic representation labeling an infinite path in the following graph.



Conclusion

Further questions

- Finding a simple graph in the case of a larger alphabet

Further questions

- Finding a simple graph in the case of a larger alphabet
- S-adic conjecture : there exists an S-adic characterization of the languages of at most linear complexity

Further questions

- Finding a simple graph in the case of a larger alphabet
- S-adic conjecture : there exists an S-adic characterization of the languages of at most linear complexity
- Can we use this characterization to study other properties of (eventually) dendric languages/shift spaces ?

Thank you for your attention!