Linear Recurrence Sequence Automata

and the Addition of Abstract Numeration Systems

WORDS 2025, Nancy, France

O. Carton, JM. Couvreur, M. Delacourt, and N. Ollinger

IRIF, Université Paris Cité, CNRS et LIFO, Université d'Orléans

nicolas.ollinger@univ-orleans.fr

We want to compute on a word

$$abbcbcabcaabbcaababbcbca \cdots$$

some combinatorial properties.

Maybe we want to know the possible lengths of cubes.

We want to compute* on a word

$$abbcbcabcaabbcaababbcbca\cdots$$

* Compute **first-order properties** using the **Büchi-Bruyère** framework. In practice, with tools like **Walnut**.

We want to compute on a **word***

$$abbcbcabcaabbcaababbcbca\cdots$$

* **fixpoint** $\tau^{\omega}(a)$ of $\tau : \begin{array}{l} a \to ab \\ b \to bc \\ c \to a \end{array}$

Pick **your** favorite fixpoint.

We want to compute on a word

$$abbcbcabcaabbcaababbcbca\cdots$$

fixpoint $\tau^\omega(a)$ of some τ.

We need a **numeration system** with **regular addition**!

Positional Numeration Systems

Here, the substitution $\tau : a \to ab,\ b \to bc,\ c \to a$ is **Pisot**.

$$M_\tau = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad P_\tau(X) = X^3 - 2X^2 + X - 1$$

We could use the canonical **Bertrand** PNS associated to its **Pisot root**.

$$\psi : a \to ab,\ b \to c,\ c \to ac \qquad U = (1, 2, 3, \cdots)$$ **[OEIS A005314]**

From **Bruyère-Hansel** and **Frougny-Solomyak**, we know that this PNS has **regular addition** and $\tau^\omega(a)$ is **U-automatic**.

Positional Numeration Systems

Here, the substitution $\tau : a \to ab,\ b \to bc,\ c \to a$ is **Pisot**.

$$M_\tau = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad P_\tau(X) = X^3 - 2X^2 + X - 1$$

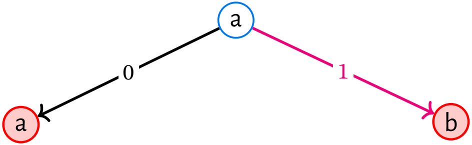
We could use the canonical **Bertrand** PNS associated to its **Pisot root**.

$$\psi : a \to ab,\ b \to c,\ c \to ac \qquad U = (1, 2, 3, \cdots)$$ **[OEIS A005314]**

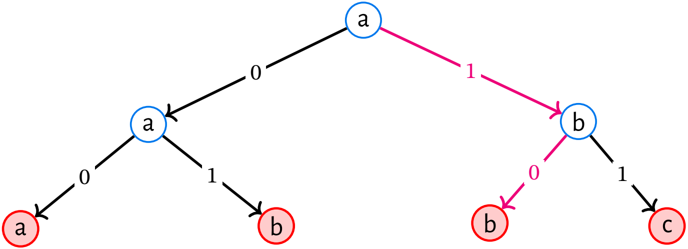
From **Bruyère-Hansel** and **Frougny-Solomyak**, we know that this PNS has **regular addition** and $\tau^\omega(a)$ is **U-automatic**… **In practice?**

a

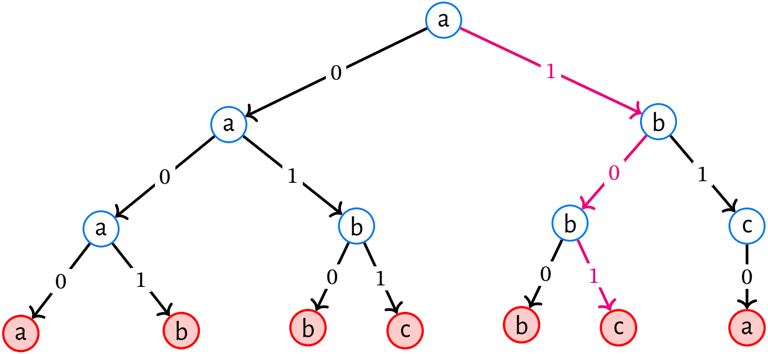
$$\tau : a \mapsto ab, \; b \mapsto bc, \; c \mapsto a$$



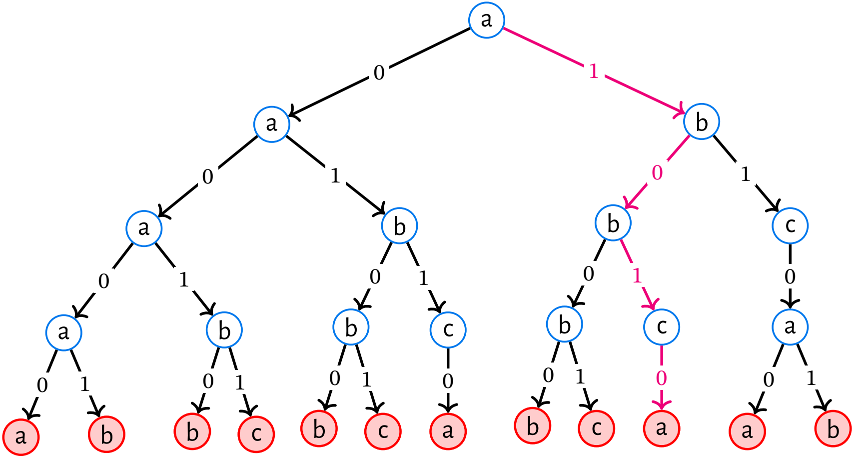
$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



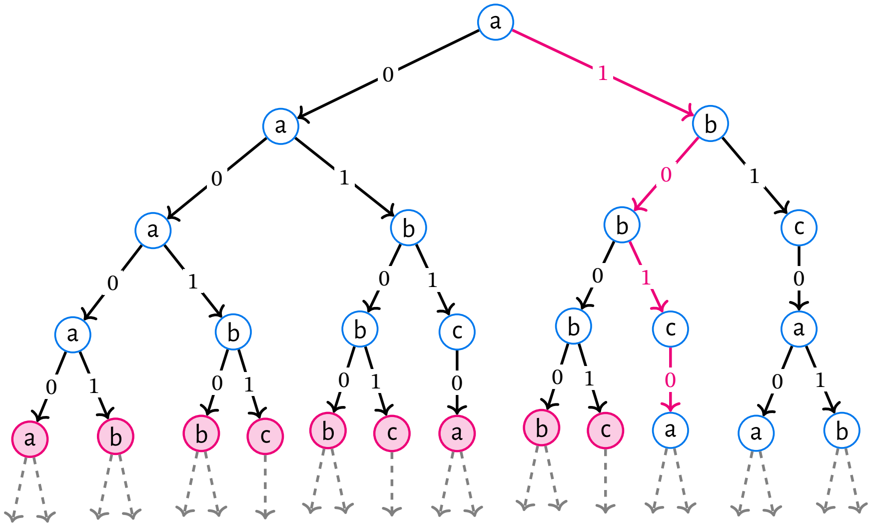
$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



$$\tau : a \mapsto ab, \ b \mapsto bc, \ c \mapsto a$$



$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



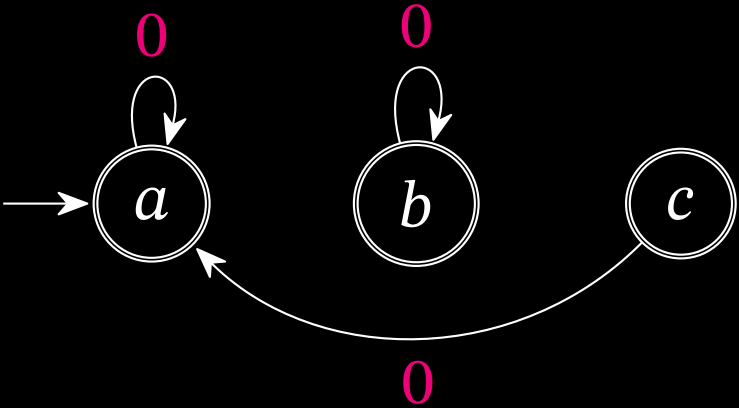
$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$

Dumont-Thomas numeration system

$$\tau : \begin{aligned} a &\to ab \\ b &\to bc \\ c &\to a \end{aligned}$$

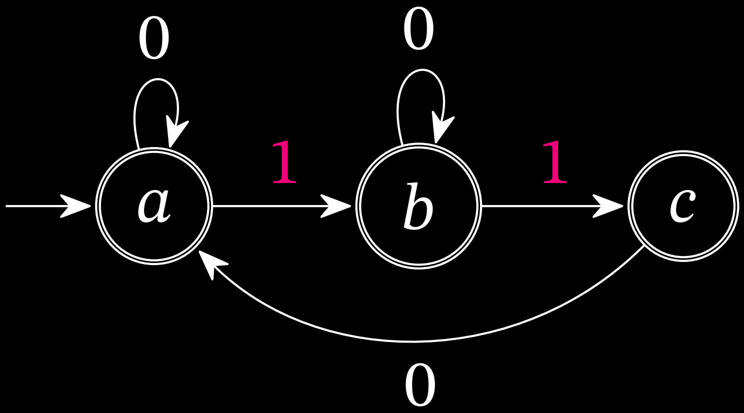
Dumont-Thomas numeration system

$$\tau : \begin{array}{l} a \to \overset{\color{magenta}{0}}{a}\overset{\color{magenta}{1}}{b} \\ b \to bc \\ c \to a \end{array}$$



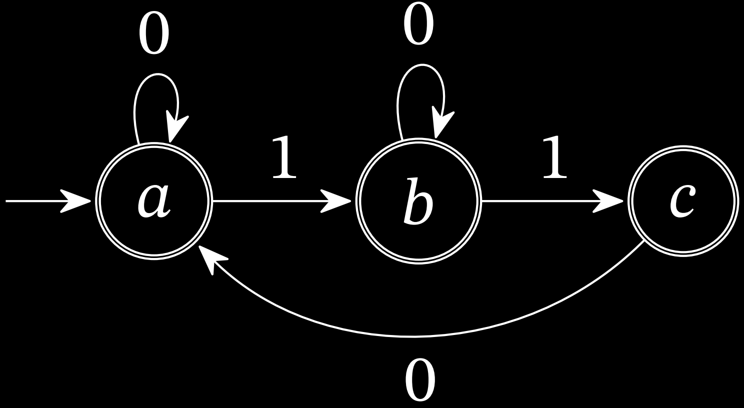
Dumont-Thomas numeration system

$$\tau : \begin{array}{l} \\ a \to \\ b \to \\ c \to \end{array} \begin{array}{l} 0\ 1 \\ ab \\ bc \\ a \end{array}$$



Dumont-Thomas numeration system

$$\begin{array}{rl} & \;\;0\;1 \\ \tau: & a \to ab \\ & b \to bc \\ & c \to a \end{array}$$



Dumont-Thomas numeration system

$$\begin{array}{rl} & \ \ {}^{0\,1} \\ \tau : a & \to ab \\ b & \to bc \\ c & \to a \end{array}$$

As a DFA this automaton gives an **Abstract Numeration System** for radix order.

As a DFAO the same automaton gives an **automatic representation** for the fixpoint.

We still need a **regular addition** to compute.

ANS and rational series

Theorem (Lecomte-Rigo) The **valuation series** of an **abstract numeration system** (ANS) is **$\mathbb{N}$-rational**.

$$\nu_{\mathcal{S}} : u \mapsto \begin{cases} \mathrm{val}_{\mathcal{S}}(u) & \text{if defined} \\ 0 & \text{if not} \end{cases}$$

Can we use this to compute the addition?

First we need to define **synchronized addition** on series.

Notation The canonical **isomorphism** between $\bigcup_{n\geqslant 0} (\Sigma^n \times \Gamma^n)$ and $(\Sigma \times \Gamma)^*$ is denoted as $\langle .,. \rangle$ for every alphabets Σ, Γ:

$$\langle u, v\rangle = (u_1, v_1)\cdots(u_m, v_m) \quad \forall u, v \in \Sigma^m \times \Gamma^m$$

Synchronized addition

Definition The **synchronized addition** $f \oplus g$ between two series $f : \Sigma^* \to \mathbb{K}$ and $g : \Gamma^* \to \mathbb{K}$ is defined as

$$\left(f \oplus g\right)(\langle u, v\rangle) = f(u) + g(v) \quad \forall (u, v) \in \Sigma^m \times \Gamma^m$$

The **support** $\mathrm{supp}(f)$ of a series f is the language $\Sigma^* \setminus f^{-1}(0)$.

Proposition An **ANS** $\mathcal{S}$ has **regular addition** if and only if $\mathrm{supp}(\nu_{\mathcal{S}} \oplus \nu_{\mathcal{S}} \oplus -\nu_{\mathcal{S}})$ is **regular**.

Unfortunately, $\nu_{\mathcal{S}} \oplus \nu_{\mathcal{S}} \oplus -\nu_{\mathcal{S}}$ is a **rational $\mathbb{Z}$-series** and testing if the support of a $\mathbb{Z}$-series is regular is generally **undecidable**…

$\mathcal{S}$-automatic

DFAO

$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$

$\mathcal{S}$-synchronized

$\mathcal{S}$-automatic

DFAO

$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$

DFA, closed by **FO formulaes**

$\langle \mathrm{repr}_{\mathcal{S}}(n), \mathrm{repr}_{\mathcal{S}}(f(n)) \rangle \in L(\mathcal{A})$

$\mathcal{S}$-regular

$\mathcal{S}$-synchronized

$\mathcal{S}$-automatic

DFAO

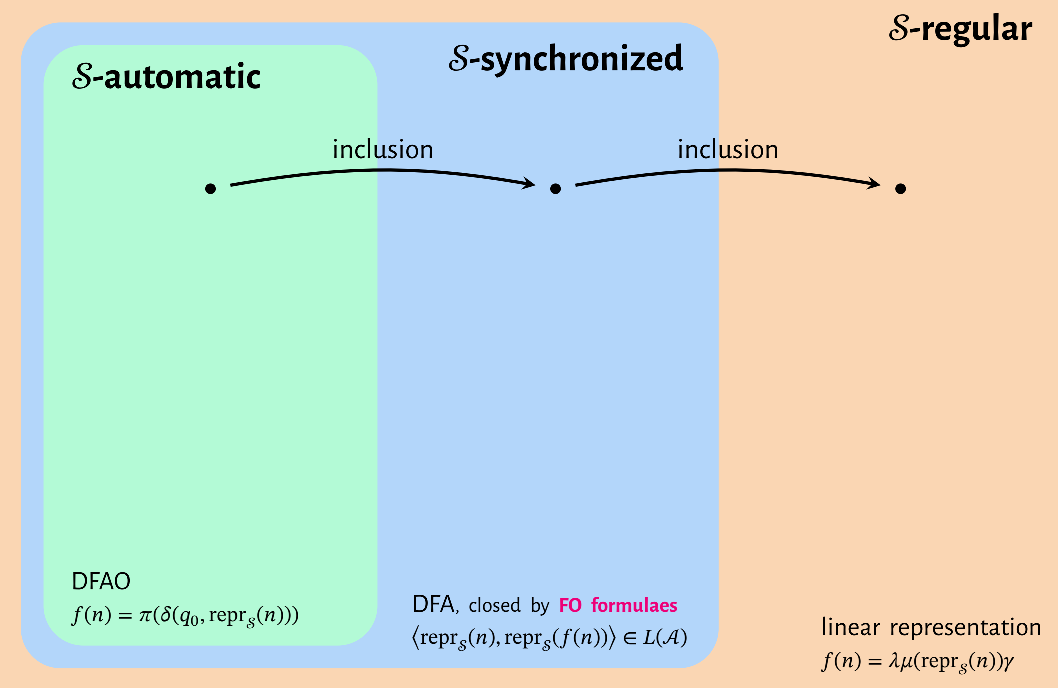
$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$

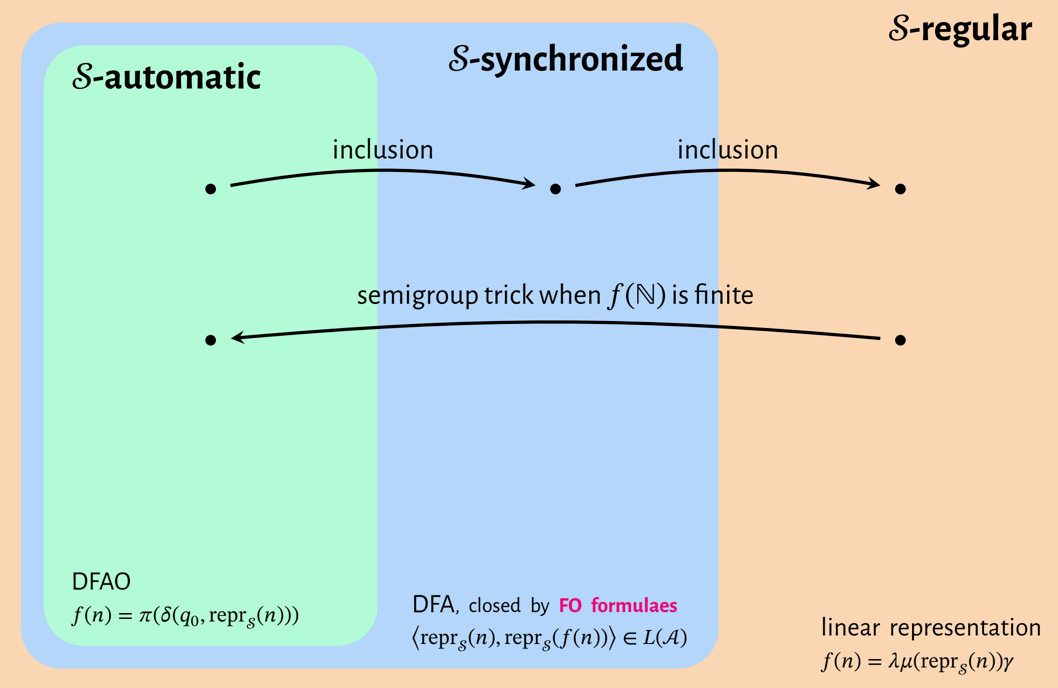
DFA, closed by **FO formulaes**

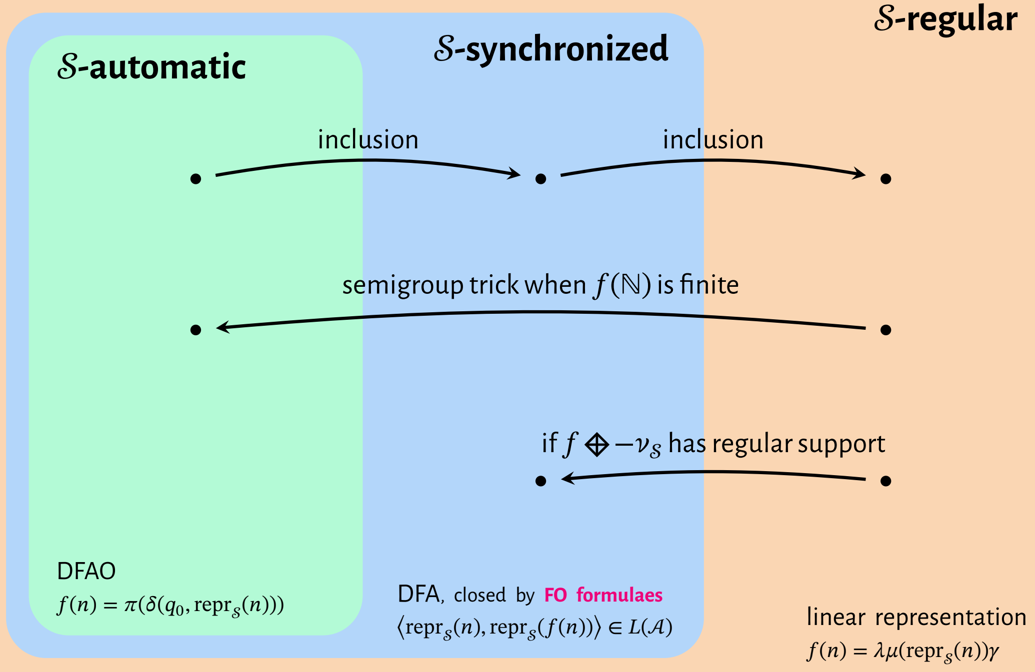
$\langle \mathrm{repr}_{\mathcal{S}}(n), \mathrm{repr}_{\mathcal{S}}(f(n)) \rangle \in L(\mathcal{A})$

linear representation

$f(n) = \lambda\mu(\mathrm{repr}_{\mathcal{S}}(n))\gamma$


$\mathcal{S}$-regular
$\mathcal{S}$-synchronized
$\mathcal{S}$-automatic
inclusion
inclusion
DFAO
$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$
DFA, closed by FO formulaes
$\langle \mathrm{repr}_{\mathcal{S}}(n), \mathrm{repr}_{\mathcal{S}}(f(n)) \rangle \in L(\mathcal{A})$
linear representation
$f(n) = \lambda\mu(\mathrm{repr}_{\mathcal{S}}(n))\gamma$


𝒮-regular
𝒮-synchronized
𝒮-automatic
inclusion
inclusion
semigroup trick when $f(\mathbb{N})$ is finite
DFAO
$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$
DFA, closed by FO formulaes
$\langle \mathrm{repr}_{\mathcal{S}}(n), \mathrm{repr}_{\mathcal{S}}(f(n)) \rangle \in L(\mathcal{A})$
linear representation
$f(n) = \lambda \mu(\mathrm{repr}_{\mathcal{S}}(n)) \gamma$


$\mathcal{S}$-regular
$\mathcal{S}$-synchronized
$\mathcal{S}$-automatic
inclusion
inclusion
semigroup trick when $f(\mathbb{N})$ is finite
if $f \oplus -\nu_{\mathcal{S}}$ has regular support
DFAO
$f(n) = \pi(\delta(q_0, \mathrm{repr}_{\mathcal{S}}(n)))$
DFA, closed by FO formulaes
$\langle \mathrm{repr}_{\mathcal{S}}(n), \mathrm{repr}_{\mathcal{S}}(f(n)) \rangle \in L(\mathcal{A})$
linear representation
$f(n) = \lambda \mu(\mathrm{repr}_{\mathcal{S}}(n)) \gamma$

In this talk

We introduce **Sequence Automata** as a generic way to manipulate similar constructions of $\mathbb{Z}$-series related to **linear recurrence sequences**.

We lift the **Bruyère-Hansel** and **Frougny-Solomyak** proofs to obtain **regular support** for the associated series under **Pisot** conditions.

It comes with an implementation and **practical tools** for Walnut users.

```
[1]: %DT dmua "a->ab, b->bc, c->a"

Studying substitution a->ab, b->bc, c->a
abbcbcabcaabbcaababbcbcaababbcabbcbcabcaababbcabbcbcaabbcbcabcaabbcaab...
Substitution polynomial: X^3-2X^2+X-1
Minimized substitution a->ab, b->bc, c->a
Substitution polynomial: X^3-2X^2+X-1
Addition polynomial: X^3-2X^2+X-1 (n=3)
θ=1.754877666246692
Substitution automaton: 3 states, 3 final states, 5 transitions.
>>> Writing /Users/nopid/Downloads/explosubst/Word Automata Library/Dmua.txt in format Walnut

>>> Writing /Users/nopid/Downloads/explosubst/Word Automata Library/DmuaParent.txt in format Walnut
```

Dumont-Thomas prefix factoring

Dumont and Thomas proved that **every prefix** p of $\tau^{\omega}(a)$ can be represented using a **unique sequence** $(p_i, a_i)_{i=0}^{k} \in (\Sigma^* \times \Sigma)^*$ as

$$p = \prod_{i=0}^{k} \tau^{k-i}(p_i) \quad \text{where } p_i a_i \text{ is a prefix of } \tau(a_{i-1}) \qquad \forall i \leqslant k$$

$$\text{(with } a_{-1} = a\text{)}$$

Thus every natural number can be decomposed as a sum of elements of **prefix length sequences**:

$$|p| = \sum_{i=0}^{k} \left|\tau^{k-i}(p_i)\right|$$

Linear Recurrence Sequences

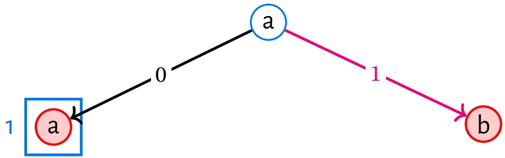
$$\tau : \begin{aligned} a &\to ab \\ b &\to bc \\ c &\to a \end{aligned} \qquad\qquad M_\tau = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

By **Cayley-Hamilton theorem**, all the sequences $(|\tau^n(u)|)$ are **linear recurrence sequences** (LRS) satisfying the recurrence relation given by the **monic characteristic polynomial** of M_τ, here $X^3 - 2X^2 + X - 1$.

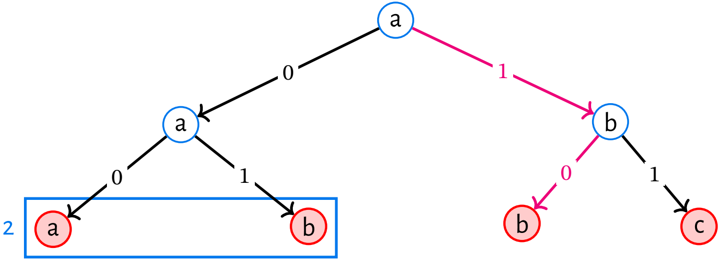
$$\begin{array}{rcccccc} |\tau^n(a)| & = & 1, & 2, & 4, & 7, & \dots \\ |\tau^n(b)| & = & 1, & 2, & 3, & 5, & \dots \\ |\tau^n(c)| & = & 1, & 1, & 2, & 4, & \dots \end{array}$$

Prefix length sequences $(|\tau^n(p_i)|)$ are **linear combinations** of these sequences: they satisfy the same recurrence relation.

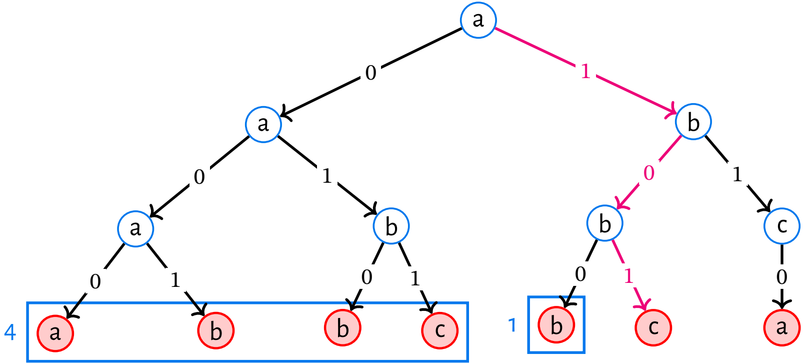
$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



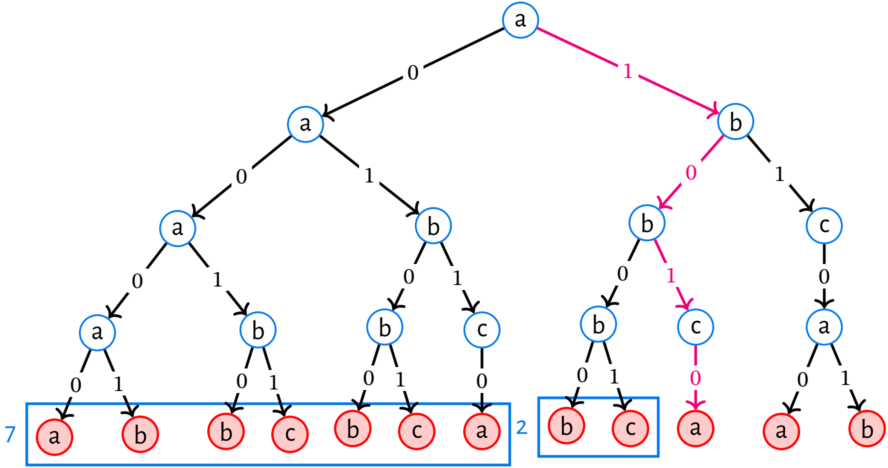
$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



$$\tau \,:\, a \mapsto ab,\; b \mapsto bc,\; c \mapsto a$$

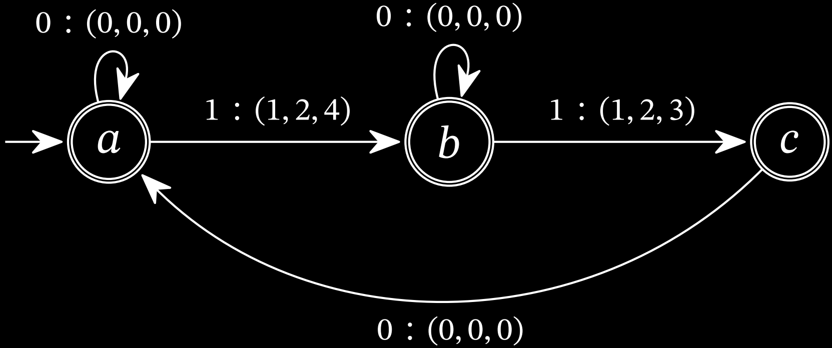


$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$



$$\tau : a \mapsto ab,\ b \mapsto bc,\ c \mapsto a$$

$X^3 - 2X^2 + X - 1$



Sequence Automaton

Sequence Automata

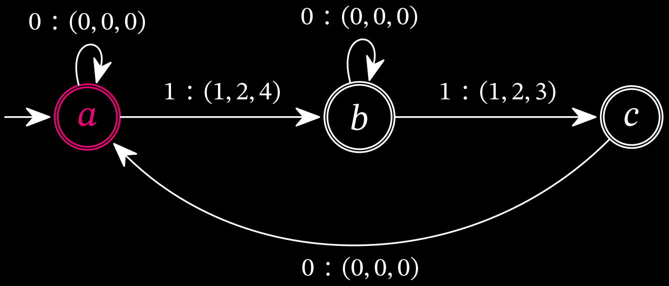
Definition A **sequence automaton** is a **partial DFA** $(Q, \Sigma, \delta, q_0, F)$ equipped with a **partial vector map** $\pi : Q \times \Sigma \to \mathbb{Z}^{\mathbb{N}}$ sharing a common domain with the **partial transition map** $\delta : Q \times \Sigma \to Q$.

Let (0) denote the zero sequence and σ denote the **shift map** on sequences: $\sigma(u)[n] = u[n+1]$ for all $u \in \mathbb{Z}^{\mathbb{N}}$ and $n \geqslant 0$.

The **transition map** and **vector map** are inductively extended from symbols to words as follows, for all $q \in Q$, $u \in \Sigma^*$ and $a \in \Sigma$:

$$\begin{aligned} \delta(q, \varepsilon) &= q & \pi(q, \varepsilon) &= (0) \\ \delta(q, ua) &= \delta(\delta(q, u), a) & \pi(q, ua) &= \sigma\pi(q, u) + \pi(\delta(q, u), a) \end{aligned}$$

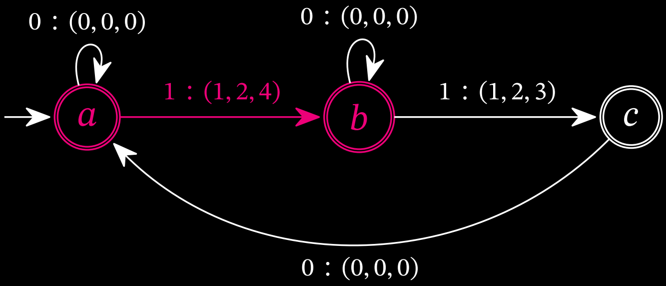
$X^3 - 2X^2 + X - 1$



$(0, 0, 0)$

$\pi(a, {}_{\blacktriangle}1010) = \ldots$

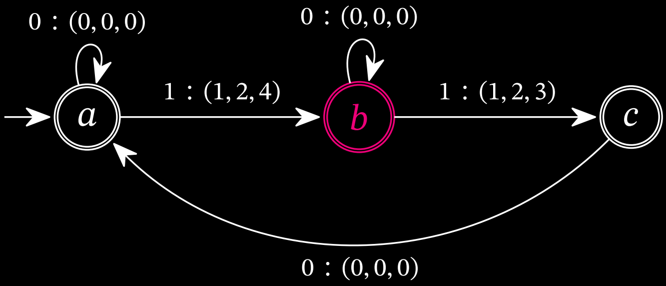
$$X^3 - 2X^2 + X - 1$$



$$\sigma(0,0,0) + (1,2,4) = (0+1, 0+2, 0+4)$$

$$\pi(a, 1010) = \ldots$$

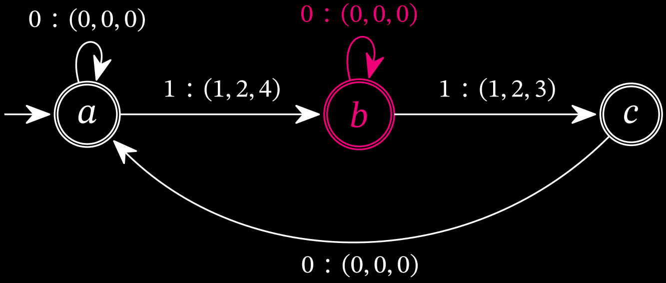
$$X^3 - 2X^2 + X - 1$$



$$(1, 2, 4)$$

$$\pi(a, 1_{\blacktriangle}010) = \ldots$$

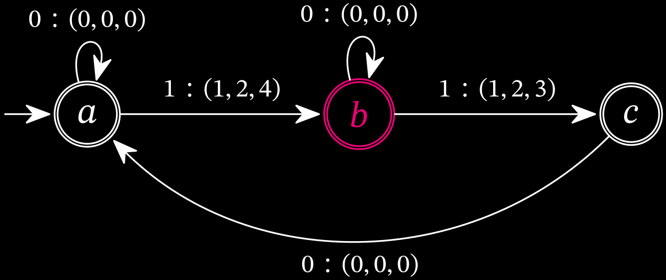
$$X^3 - 2X^2 + X - 1$$



$$\sigma(1,2,4) + (0,0,0) = (2+0, 4+0, 7+0)$$

$$\pi(a, 1010) = \ldots$$

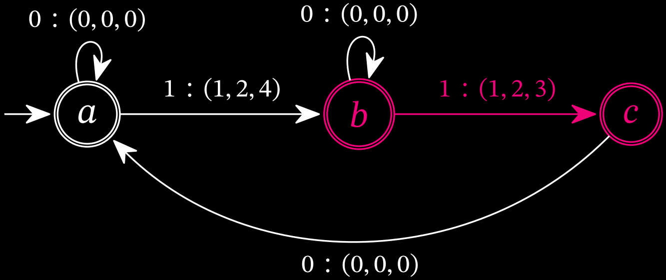
$X^3 - 2X^2 + X - 1$



$(2, 4, 7)$

$\pi(a, 10\,10) = \ldots$

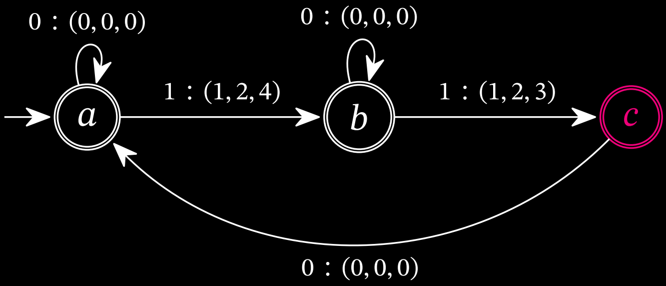
$$X^3 - 2X^2 + X - 1$$



$$\sigma(2,4,7) + (1,2,3) = (4+1, 7+2, 12+3)$$

$$\pi(a, 1010) = \dots$$

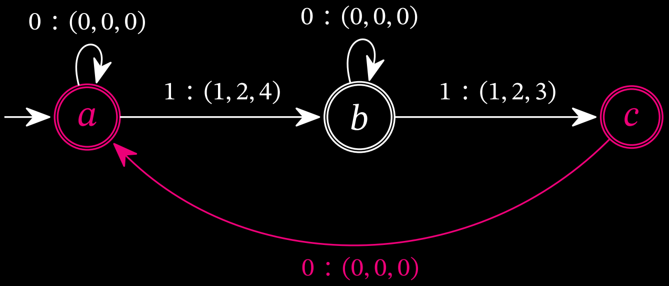
$$X^3 - 2X^2 + X - 1$$



$$(5, 9, 15)$$

$$\pi(a, 101\,0) = \dots$$

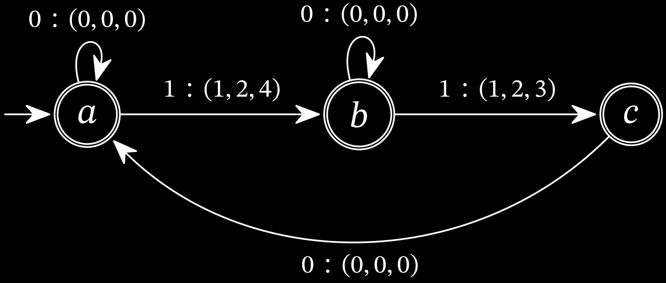
$$X^3 - 2X^2 + X - 1$$



$$\sigma(5, 9, 15) + (0, 0, 0) = (9 + 0, 15 + 0, 26 + 0)$$

$$\pi(a, 1010) = \ldots$$

$$X^3 - 2X^2 + X - 1$$



$$(9, 15, 26)$$

$$\pi(a, 1010) = (9, 15, 26)$$

Series and Linear Combinations

Definition The **series** $s_{\mathcal{A}}$ of a **sequence automaton** $\mathcal{A}$ maps every word $u \in \Sigma^*$ to the first element of its vector $\pi(q_0, u)[0]$.

Using **product DFA** and linear combination of vector maps, one can combine **sequence automata** to produce **sequence automata**:

- $\mathcal{A} + \mathcal{B}$ of series $s_{\mathcal{A}} \oplus s_{\mathcal{B}}$;
- $\alpha\mathcal{A}$ of series $\alpha s_{\mathcal{A}}$ for every scalar α.

Proposition The **valuation series** $\nu_{\mathcal{S}}$ of an ANS is the **series of a sequence automaton**. As a consequence $\nu_{\mathcal{S}} \oplus \nu_{\mathcal{S}} \oplus -\nu_{\mathcal{S}}$ is also the **series of a sequence automaton**.

Linear Recurrence Sequence Automata

Definition When all sequences in the **vector map** of a sequence automaton are **LRS**, the **recurrence polynomial** of the automaton is the **minimal polynomial** of all these sequences.

Remark When all LRS come from **incidence matrices**, the **recurrence polynomial** is **monic**.

Theorem The **support** of the series of a sequence automaton with **(ultimately) Pisot recurrence polynomial** is **regular**.

The proof lift bounds and techniques from **Bruyère-Hansel** and **Frougny-Solomyak** into sequence automata series.

Case 1. starting from a Pisot substitution

When starting from a (ultimately) **Pisot substitution**, **sequence automata** let us compute numerous DFA and DFAO:

- **addition DFA** for **Dumont-Thomas numeration systems** of fixpoints;

Case 1. starting from a Pisot substitution

When starting from a (ultimately) **Pisot substitution**, **sequence automata** let us compute numerous DFA and DFAO:

- **addition DFA** for **Dumont-Thomas numeration systems** of fixpoints;
- **conversion between numeration systems** sharing the same **recurrence polynomial**, for example the associated **Bertrand** numeration system;

Case 1. starting from a Pisot substitution

When starting from a (ultimately) **Pisot substitution**, **sequence automata** let us compute numerous DFA and DFAO:

- **addition DFA** for **Dumont-Thomas numeration systems** of fixpoints;
- **conversion between numeration systems** sharing the same **recurrence polynomial**, for example the associated **Bertrand** numeration system;
- **abelian complexity DFAO** for Pisot substitutions [arXiV 2504.13584];

Case 1. starting from a Pisot substitution

When starting from a (ultimately) **Pisot substitution**, **sequence automata** let us compute numerous DFA and DFAO:

- **addition DFA** for **Dumont-Thomas numeration systems** of fixpoints;
- **conversion between numeration systems** sharing the same **recurrence polynomial**, for example the associated **Bertrand** numeration system;
- **abelian complexity DFAO** for Pisot substitutions **[arXiV 2504.13584]**;
- **addition DFA** for **Abstract Numeration Systems** built as morphic images of fixpoints;
- …

Case 2. the Non Pisot substitutions

In general, we can still **construct the sequence automata**.

On a **case-by-case** basis for each application:

- sometimes the **recurrence polynomial** of the automaton is **Pisot**;

$$a \to aba,\ b \to bab$$

- sometimes it is not but a **manual bound** let us derive a valid DFA;

$$a \to aba,\ b \to b$$

- sometimes a **pumping argument** let us prove there is no DFA.

$$a \to ab,\ b \to bc,\ c \to c$$

In Practice

We maintain a collection of **Python tools** to interact with **Walnut**:

- **licofage** — sequence automata library generating Walnut files:
  - basic usage: give a substitution, get the addition DFA and word DFAO;
  - advanced usage: generic sequence automata manipulation in Python;
- **ratser** — missing library for rational power series and semigroup trick;
- **walnut-kernel** — a Jupyter kernel to combine Walnut and all this (hello Walnut Notebooks!);
- **jupywalnut** — a Docker container for a simplified use of everything;
- **walnut-cli** — Walnut + Java runtime packaged together.

Everything is available on **GitHub**, install Python libs from **Pypi**.

```
$ pip install walnut-cli
$ walnut
```