# Énumération et numération

Victor Marsault

Telecom-ParisTech

Soutenance de thèse
2016-03-01

# Plan

# La numération, entre mathématiques et informatique



## Nombres

- Quantités abstraites
- Liés aux mathématiques

Ex: $8,\ \sqrt{2},\ \pi$, *etc*.

Représentation

## Mots

- Suite de lettres/chiffres
- Liés à l'informatique

Ex: $abbac,\ 3.141592\cdots,$

Évaluation

# Les trois composants d'un système de numération



## Alphabet

Les chiffres autorisés.

## Une fonction d'évaluation

- mot $\rightarrow$ nombre
- la valeur d'un mot $u$ est notée $\pi(u)$

## Une fonction ou algorithme de représentation

- nombre $\rightarrow$ mot
- la représentation d'un nombre $n$ est noté $\langle n \rangle$

# Exemple: le système usuel (base 10)



## Alphabet

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

# Exemple: le système usuel (base 10)



## Alphabet

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation

 $\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

# Exemple: le système usuel (base 10)



## Alphabet

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation

 $\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

## Évaluation

2 3 5 $\rightarrow$

# Exemple: le système usuel (base 10)



## Alphabet

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation

 $\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

## Évaluation

2 3 5 $\rightarrow$

# Exemple: le système usuel (base 10)



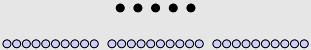
## Alphabet

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation


$\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

## Évaluation



2 3 5 $\rightarrow$

# Exemple: le système usuel (base 10)



## Alphabet

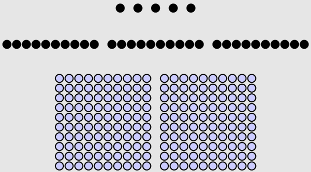
$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation

$\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

## Évaluation

2 3 5 $\rightarrow$



# Exemple: le système usuel (base 10)



## Alphabet

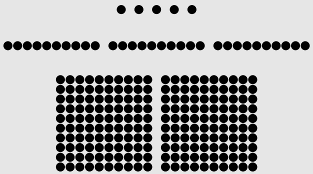
$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

## Représentation

$\rightarrow$ 19 (le chiffre 1 suivi du chiffre 9)

## Évaluation

2 3 5 $\rightarrow$



### Mot

Un *mot* sur un alphabet $A$ est une suite de lettres/chiffres de $A$.

Exemples: 19 ; 01032016 .

### Mot

Un *mot* sur un alphabet $A$ est une suite de lettres/chiffres de $A$.

Exemples: 19 ; 01032016 .

### Langage

Un langage sur un alphabet $A$ est un ensemble de mots.

Exemples:
- les mots dont la valeur est divisible par 7 : $\{7, 14, 21, 28, 35, ...\}$ ;
- les mots dont la valeur est une puissance de 10 : $\{10, 100, 1000, 10000, ...\}$.

### Mot

Un *mot* sur un alphabet $A$ est une suite de lettres/chiffres de $A$.

Exemples: 19 ; 01032016 .

### Langage

Un langage sur un alphabet $A$ est un ensemble de mots.

Exemples:
- les mots dont la valeur est divisible par 7 : $\{7, 14, 21, 28, 35, \ldots\}$ ;
- les mots dont la valeur est une puissance de 10 : $\{10, 100, 1000, 10000, \ldots\}$.

Un système de numération fait correspondre un langage et un ensemble de nombres.

# Plan

# La retenue en base 10



| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |

# La retenue en base 10



| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |

# La retenue en base 10

| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+1$ | $-10$ |
| | | | 10 | 2 |

# La retenue en base 10



| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+1$ | $-10$ |
| | | 9 | 10 | 2 |

# La retenue en base 10

| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+1$ | $-10$ |
| | | 9 | 10 | 2 |
| | | $+1$ | $-10$ | |
| | | 10 | 0 | |

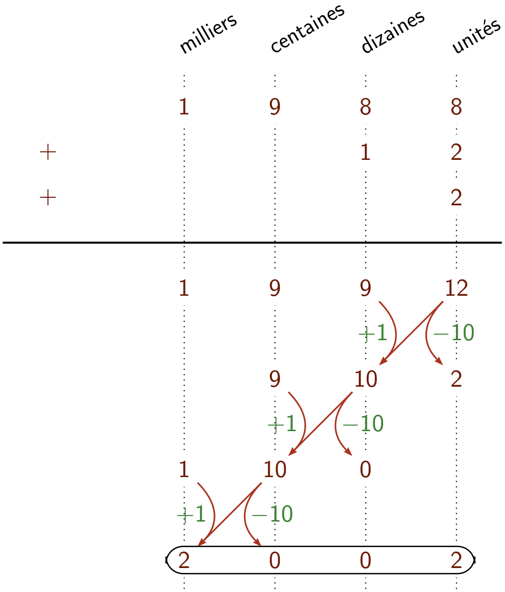
# La retenue en base 10

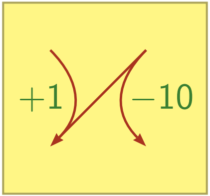
| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+1$ | $-10$ |
| | | 9 | 10 | 2 |
| | | $+1$ | $-10$ | |
| | 1 | 10 | 0 | |
| | $+1$ | $-10$ | | |
| | 2 | 0 | | |

# La retenue en base 10

| | milliers | centaines | dizaines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | +1 | −10 |
| | | 9 | 10 | 2 |
| | | +1 | −10 | |
| | 1 | 10 | 0 | |
| | +1 | −10 | | |
| | 2 | 0 | 0 | 2 |

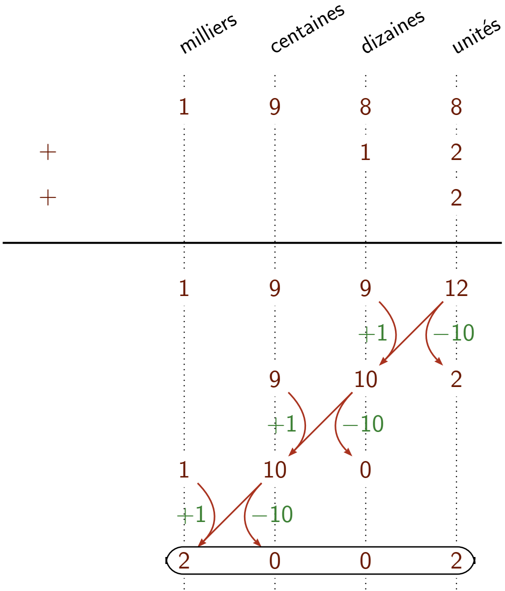
# La retenue en base 10



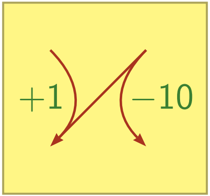



# La retenue en base 10

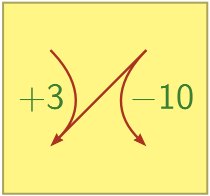





Base $\frac{10}{1}$



Base $\frac{10}{3}$

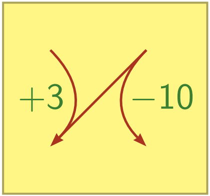


# La retenue en base $\frac{10}{3}$



| $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|

Base $\frac{10}{3}$

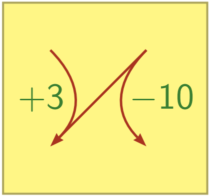


# La retenue en base $\frac{10}{3}$



| | $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |

Base $\frac{10}{3}$

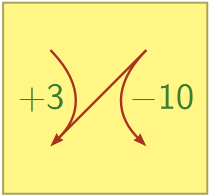


# La retenue en base $\frac{10}{3}$



| | $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| $+$ | | | 1 | 2 |
| $+$ | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+3$ | $-10$ |
| | | | 12 | 2 |

Base $\frac{10}{3}$

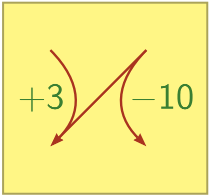


# La retenue en base $\frac{10}{3}$



| | $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| $+$ | | | 1 | 2 |
| $+$ | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+3$ | $-10$ |
| | | 9 | 12 | 2 |
| | | $+3$ | $-10$ | |
| | | 12 | 2 | |

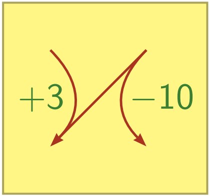
Base $\frac{10}{3}$



# La retenue en base $\frac{10}{3}$

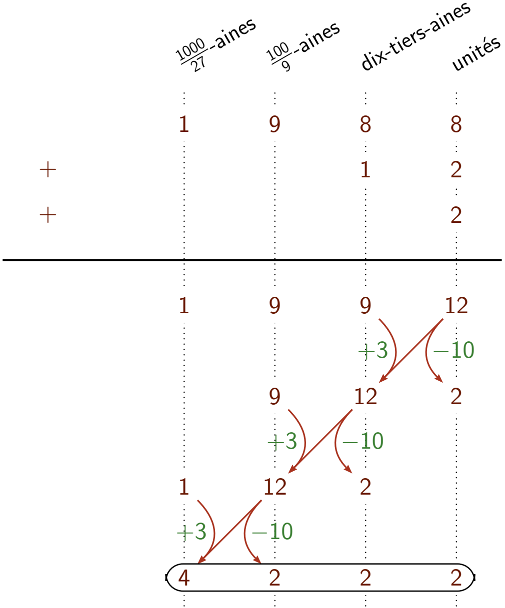
| | $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|---|
| | 1 | 9 | 8 | 8 |
| + | | | 1 | 2 |
| + | | | | 2 |
| | 1 | 9 | 9 | 12 |
| | | | $+3$ | $-10$ |
| | | 9 | 12 | 2 |
| | | $+3$ | $-10$ | |
| | 1 | 12 | 2 | |
| | $+3$ | $-10$ | | |
| | 4 | 2 | | |

Base $\frac{10}{3}$

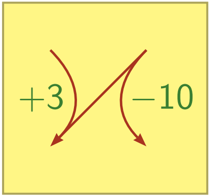


# La retenue en base $\frac{10}{3}$






Base $\frac{10}{3}$



# Calculer la représentation de 42 en base $\frac{10}{3}$

8

| $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|

# Calculer la représentation de 42 en base $\frac{10}{3}$



| $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|
| | | | 42 |

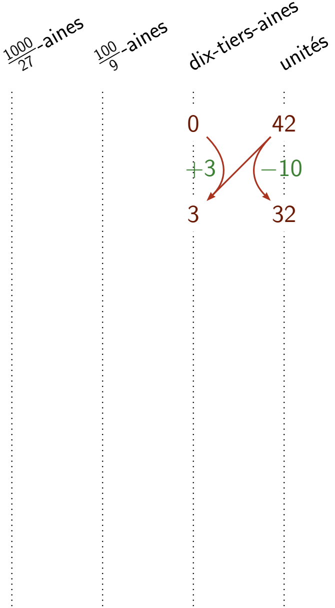
# Calculer la représentation de 42 en base $\frac{10}{3}$



| $\frac{1000}{27}$-aines | $\frac{100}{9}$-aines | dix-tiers-aines | unités |
|---|---|---|---|
| | | 0 | 42 |

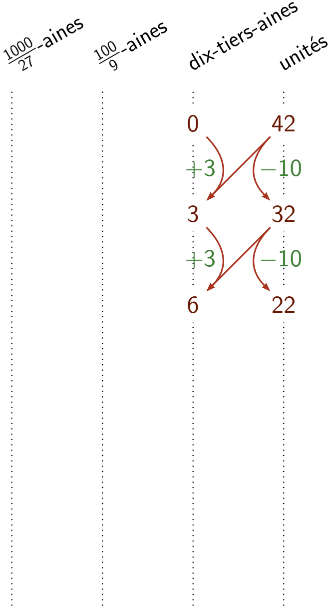
# Calculer la représentation de 42 en base $\frac{10}{3}$





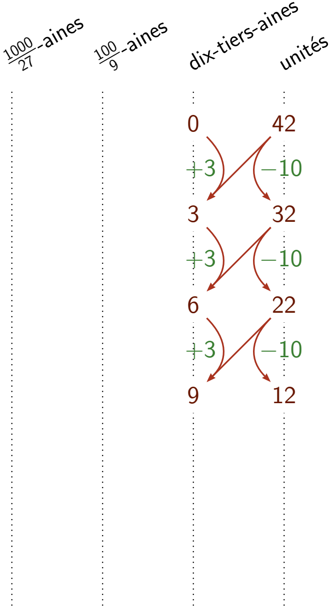
# Calculer la représentation de 42 en base $\frac{10}{3}$





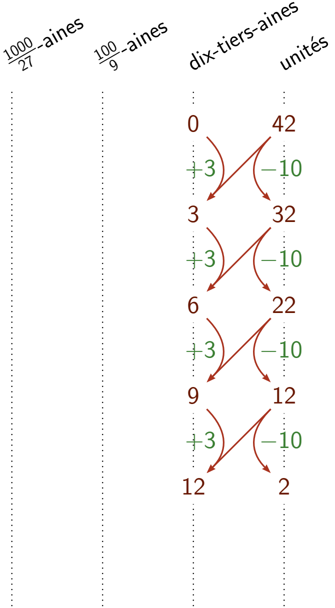
# Calculer la représentation de 42 en base $\frac{10}{3}$





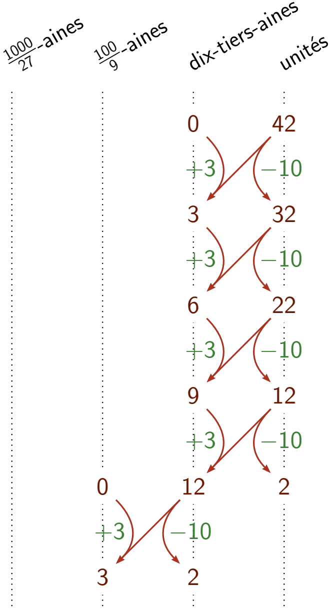
# Calculer la représentation de 42 en base $\frac{10}{3}$



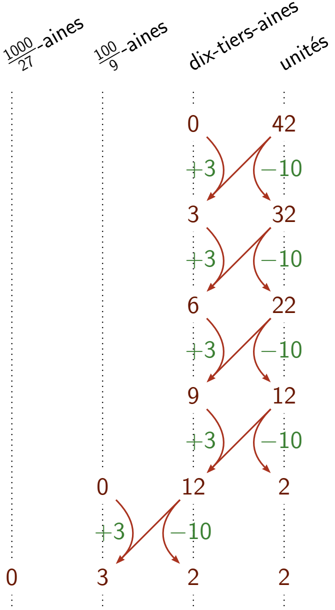


# Calculer la représentation de 42 en base $\frac{10}{3}$

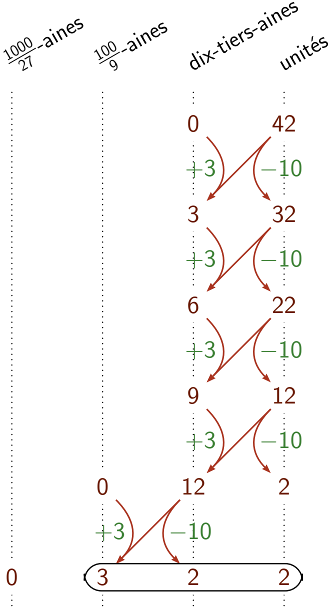




# Calculer la représentation de 42 en base $\frac{10}{3}$

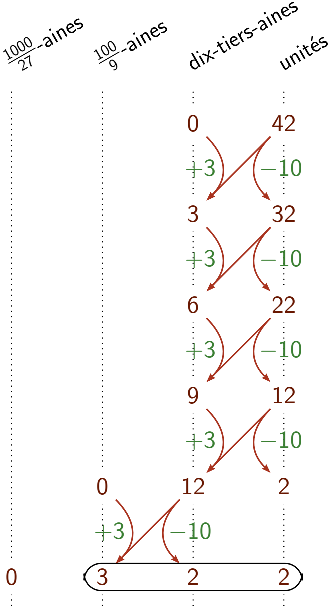


# Calculer la représentation de 42 en base $\frac{10}{3}$



# Calculer la représentation de 42 en base $\frac{10}{3}$






## Vérification: calcul de $\pi(322)$

$$= 3 \times \frac{100}{9} + 2 \times \frac{10}{3} + 2 \times 1$$

$$= \frac{300}{9} + \frac{20}{3} + 2$$

$$= \frac{300 + 60 + 18}{9}$$

$$= \frac{378}{9}$$

$$= 42$$

## Calculer une représentation en base $\frac{10}{3}$



- Si $n < 10$, alors sa représentation est lui-même : $\langle n \rangle = n$.

# Calculer une représentation en base $\frac{10}{3}$

- Si $n < 10$, alors sa représentation est lui-même : $\langle n \rangle = n$.
- Sinon, on calcule le chiffre $a$ le plus à droite de $\langle n \rangle$:

$$\langle n \rangle \;=\; \langle 3 \times k \rangle\, a$$

  - où $k$ est le quotient de la div. euclidienne de $n$ par 10 (le nombre de retenues à appliquer);
  - et $a$ est le reste de la div. euclidienne de $n$ par 10 ($a = n - k \times 10$).

# Résumé de la base $\frac{p}{q}$: la variante FK



## Données

Deux entiers $p$ et $q$, premiers entre eux, tels que $p > q > 1$.

## Alphabet

$$A_p \;=\; \{0,\ 1,\ 2, \ldots, p-1\}$$

## Évaluation

$$\pi(a_n \cdots a_1\, a_0) \;=\; a_n \left(\frac{p}{q}\right)^n + \cdots + a_2 \left(\frac{p}{q}\right)^2 + a_1 \left(\frac{p}{q}\right) + a_0$$

## Représentation

Calculée de droite à gauche, grâce à la formule:

$$\langle n \rangle \;=\; \langle k \times q \rangle\, a \qquad \text{où} \quad n = (p \times k) + a \quad \text{et} \quad 0 < a < p \quad .$$

# La base $\frac{p}{q}$ standard (AFS)



## Données

Deux entiers $p$ et $q$, premiers entre eux, tels que $p > q > 1$.

## Alphabet

$$A_p \; = \; \{0,\; 1,\; 2, \ldots, p-1\}$$

## Évaluation

$$\pi(a_n \cdots a_1\, a_0) \; = \; \boxed{\frac{1}{q}} \left[ a_n \left(\frac{p}{q}\right)^n + \cdots + a_2 \left(\frac{p}{q}\right)^2 + a_1 \left(\frac{p}{q}\right) + a_0 \right]$$

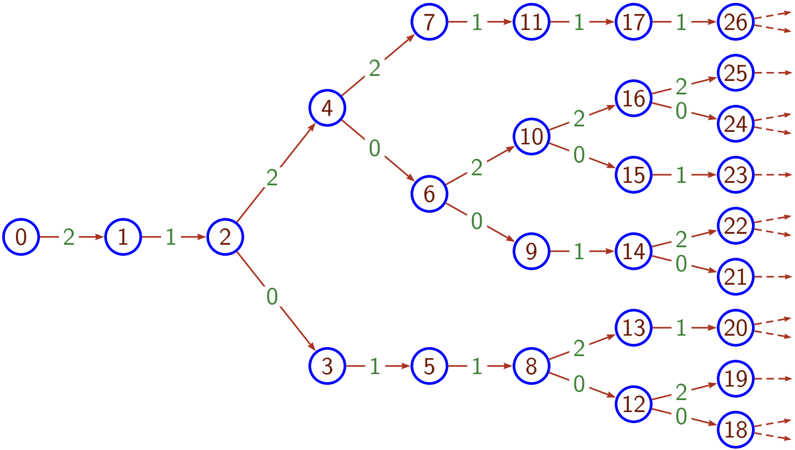
## Représentation

Calculée de droite à gauche, grâce à la formule:

$$\langle n \rangle \; = \; \langle k \cancel{\times q} \rangle\, a \qquad \text{où} \;\; (n \times q) = (p \times k) + a \;\; \text{et} \;\; 0 < a < p \;\; .$$

# Le langage $L_{\frac{p}{q}}$

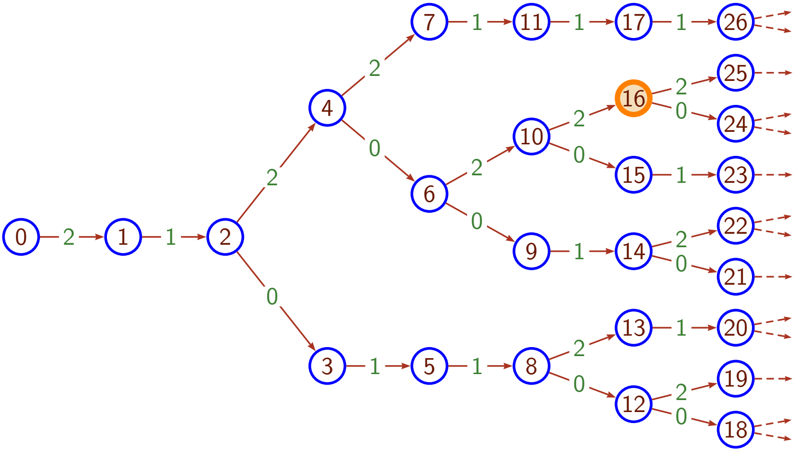
$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




Arbre étiqueté qui représente le langage $L_{\frac{3}{2}}$

# Le langage $L_{\frac{p}{q}}$

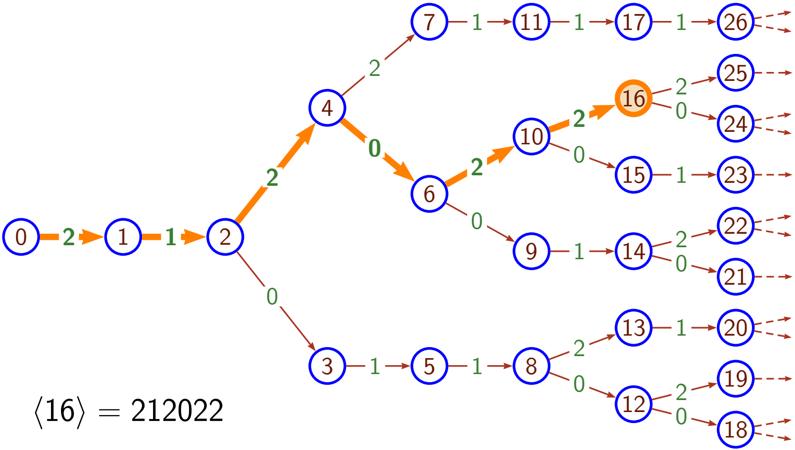
$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




Arbre étiqueté qui représente le langage $L_{\frac{3}{2}}$

# Le langage $L_{\frac{p}{q}}$

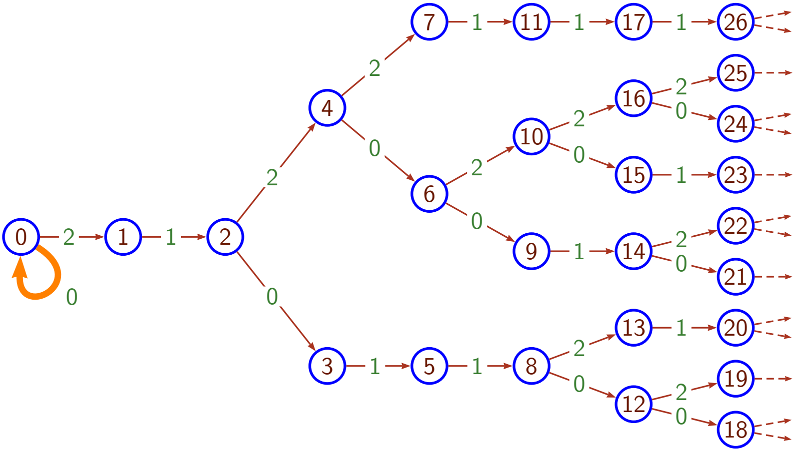
$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




$\langle 16 \rangle = 212022$

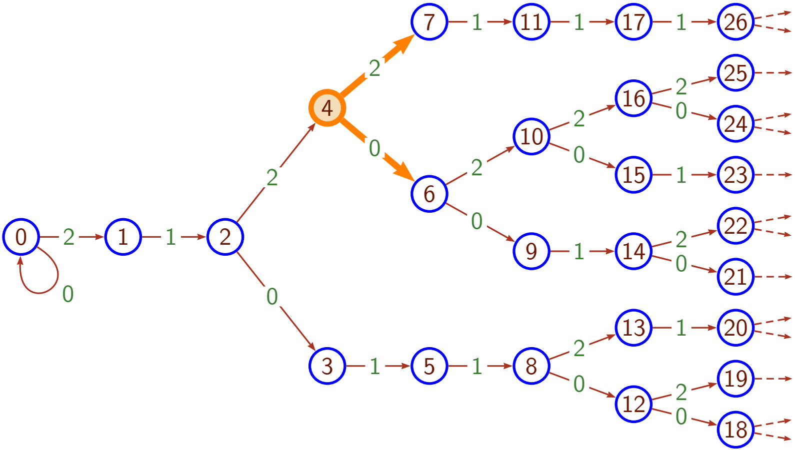
Arbre étiqueté qui représente le langage $L_{\frac{3}{2}}$

$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




I-Arbre étiqueté qui représente le langage calable $0^* L_{\frac{3}{2}}$

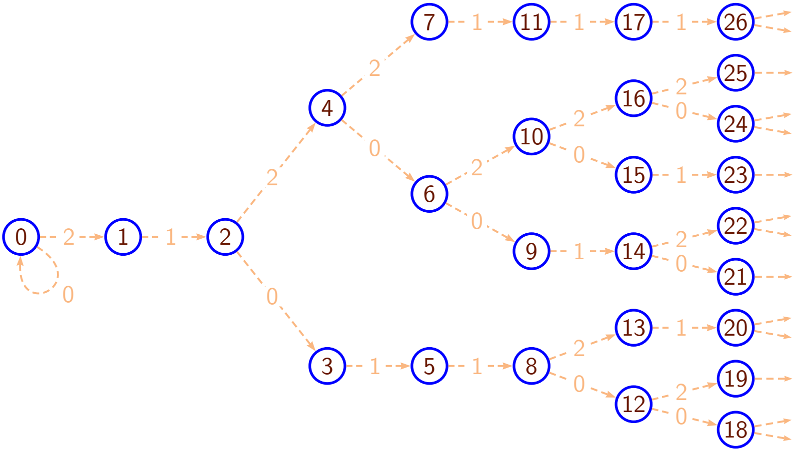
$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




I-Arbre étiqueté qui représente le langage calable $0^* L_{\frac{3}{2}}$

# Le langage $L_{\frac{p}{q}}$

$L_{\frac{p}{q}}$: l'ensemble des représentations des nombres entiers en base $\frac{p}{q}$.




I-Arbre étiqueté qui représente le langage calable $0^* L_{\frac{3}{2}}$

## Quelques propriétés de $L_{\frac{p}{q}}$



### Lemme (Akiyama, Frougny, Sakarovitch, 2008)

- Le langage $L_{\frac{p}{q}}$ est clos par préfixe.
- Le langage $L_{\frac{p}{q}}$ est prolongeable à droite.

### Théorème (Akiyama, Frougny, Sakarovitch, 2008)

$L_{\frac{p}{q}}$ n'est pas un langage régulier, ni même algébrique.

# Quelques propriétés de $L_{\frac{p}{q}}$



**Lemme (Akiyama, Frougny, Sakarovitch, 2008)**

- Le langage $L_{\frac{p}{q}}$ est clos par préfixe.
- Le langage $L_{\frac{p}{q}}$ est prolongeable à droite.

**Théorème (Akiyama, Frougny, Sakarovitch, 2008)**

$L_{\frac{p}{q}}$ n'est pas un langage régulier, ni même algébrique.

$L_{\frac{p}{q}}$ possède la propriété d'itération préfixe finie (FLIP).

# La propriété d'itération préfixe finie (FLIP)



## Definition

$A$: un alphabet.
$L$: un langage sur $A$.

Le langage $L$ est dit *FLIP* si
$\forall u, v \in A^*$, $\exists$ qu'un nombre fini d'entiers $i$
tels que $u\,v^i$ est le préfixe d'un mot de $L$.

Exemple : les préfixes d'un mot infini apériodique.

# La propriété d'itération préfixe finie (FLIP)



## Definition

$A$: un alphabet.
$L$: un langage sur $A$.

Le langage $L$ est dit *FLIP* si
$\forall u, v \in A^*$, $\exists$ qu'un nombre fini d'entiers $i$
tels que $u\, v^i$ est le préfixe d'un mot de $L$.

Exemple : les préfixes d'un mot infini apériodique.

## Remarques

Hormis $L_{\frac{p}{q}}$, on ne connaît pas d'exemples naturels de langage FLIP.

# La propriété d'itération préfixe finie (FLIP)



## Definition

$A$: un alphabet.
$L$: un langage sur $A$.

Le langage $L$ est dit *FLIP* si
$\forall u, v \in A^*$, $\exists$ qu'un nombre fini d'entiers $i$
tels que $u\, v^i$ est le préfixe d'un mot de $L$.

Exemple : les préfixes d'un mot infini apériodique.

## Remarques

Hormis $L_{\frac{p}{q}}$, on ne connaît pas d'exemples naturels de langage FLIP.

Chaque branche infinie de $L$ (représenté comme un arbre) est étiquetée par un mot apériodique.

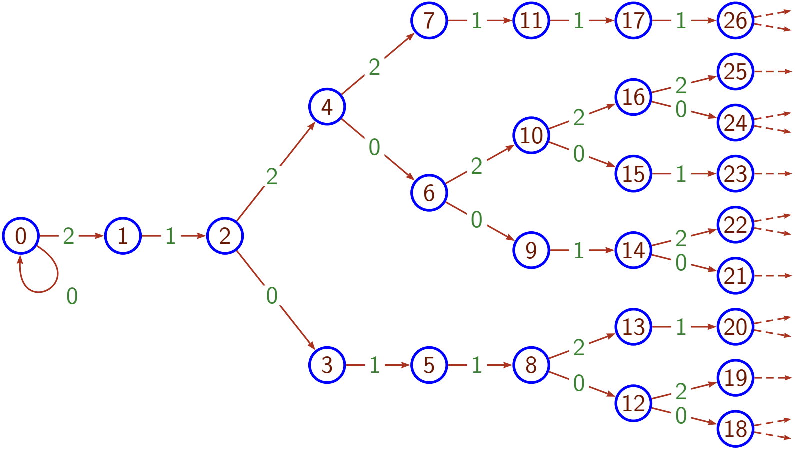
### Théorème

$\frac{p}{q}$: une base rationnelle.
$M$: sous-monoïde additif de $\mathbb{Q}$.

$$\left.\begin{array}{l}M \text{ est finiment engendré} \\ \text{tous les nombres de } M \text{ sont représentables}\end{array}\right\} \Longrightarrow \langle M \rangle \text{ est FLIP.}$$

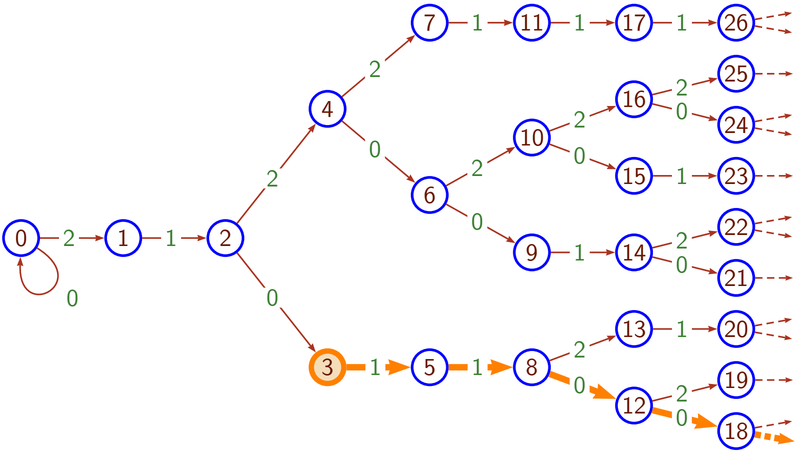
## Mot minimal de $n$: $w_n^-$

Le plus petit mot qui étiquette une branche partant de $n$.



## Mot minimal de $n$: $w_n^-$

Le plus petit mot qui étiquette une branche partant de $n$.

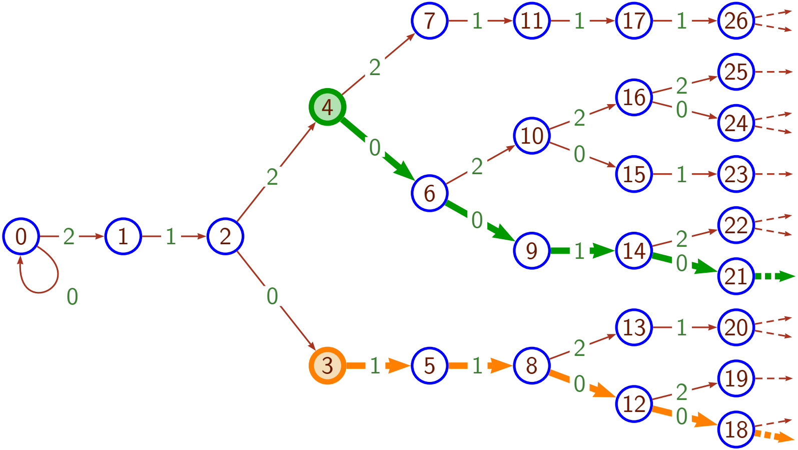


# Regarder $L_{\frac{p}{q}}$ autrement



## Mot minimal de $n$: $w_n^-$

Le plus petit mot qui étiquette une branche partant de $n$.



### Definition

On note $\xi$ la fonction définie par :

$$\begin{array}{rccc} \xi : & (A_q)^\omega & \longrightarrow & (A_q)^\omega \\ & w_n^- & \longmapsto & w_{n+1}^- \end{array}$$

### Théoreme

La fonction $\xi$ est réalisée par un transducteur infini $\mathcal{D}_{\frac{p}{q}}$ dont la structure est très proche de celle de $0^* L_{\frac{p}{q}}$.

## Definition

On note $\xi$ la fonction définie par :

$$\begin{array}{rccc} \xi: & (A_q)^\omega & \longrightarrow & (A_q)^\omega \\ & w_n^- & \longmapsto & w_{n+1}^- \end{array}$$

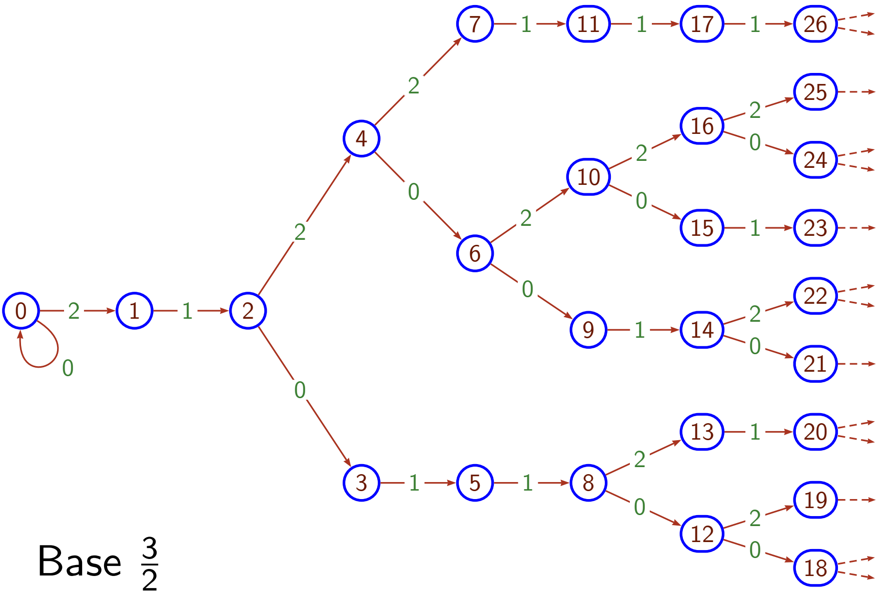
## Théoreme

La fonction $\xi$ est réalisée par un transducteur infini $\mathcal{D}_{\frac{p}{q}}$ dont la structure est très proche de celle de $0^*L_{\frac{p}{q}}$.

On passe de $0^*L_{\frac{p}{q}}$ à $\mathcal{D}_{\frac{p}{q}}$ en deux étapes:

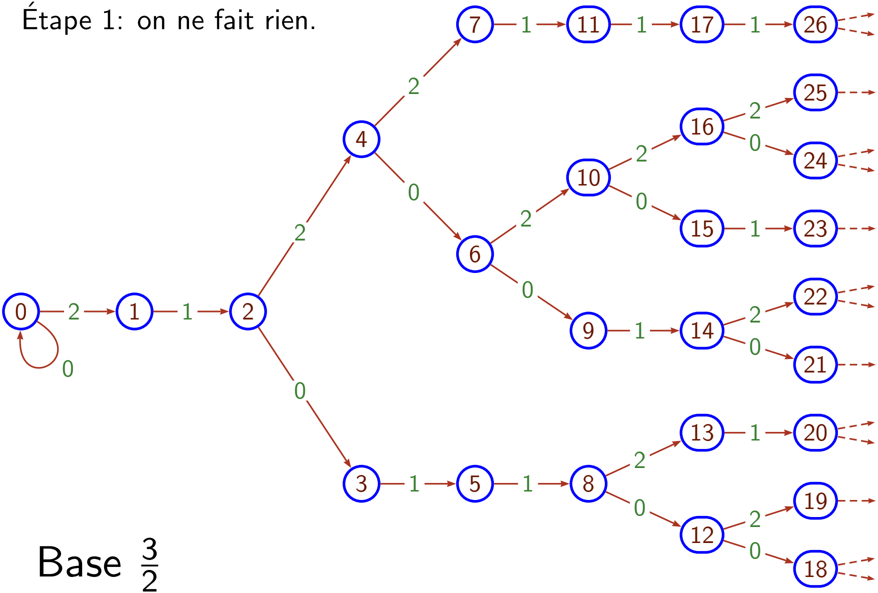
- Modifier l'alphabet (afin qu'il contienne $(2q-1)$ chiffres)
    - suppressions des petites lettres
    - ou ajout de chiffres *négatifs* ;
- puis appliquer une substitution sur les étiquettes.

# Ex. 1: quand la base est moyenne ($p = 2q - 1$)




Base $\frac{3}{2}$

Étape 1: on ne fait rien.




Base $\frac{3}{2}$

# Ex. 1: quand la base est moyenne ($p = 2q - 1$)

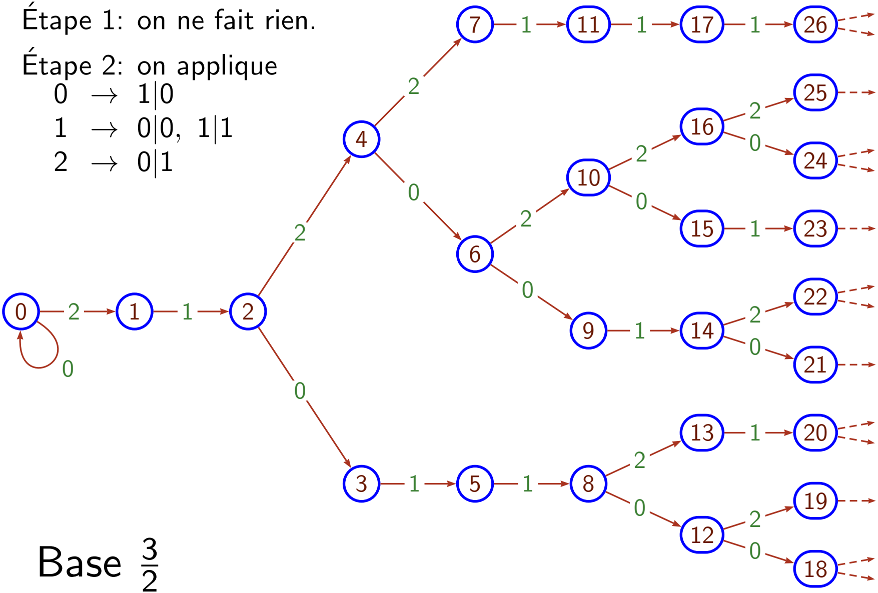
Étape 1: on ne fait rien.

Étape 2: on applique

$0 \rightarrow 1|0$

$1 \rightarrow 0|0,\ 1|1$

$2 \rightarrow 0|1$



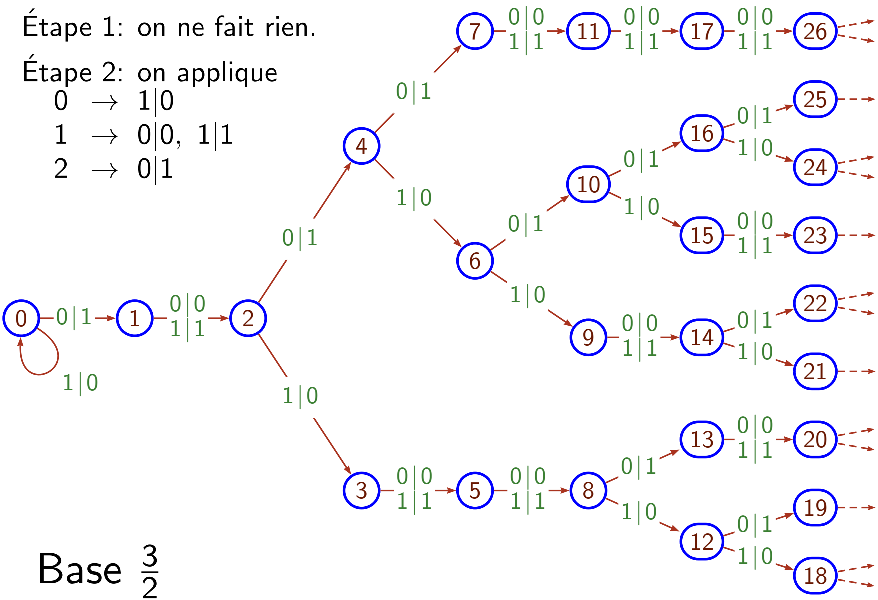

Base $\frac{3}{2}$

# Ex. 1: quand la base est moyenne ($p = 2q - 1$)

Étape 1: on ne fait rien.

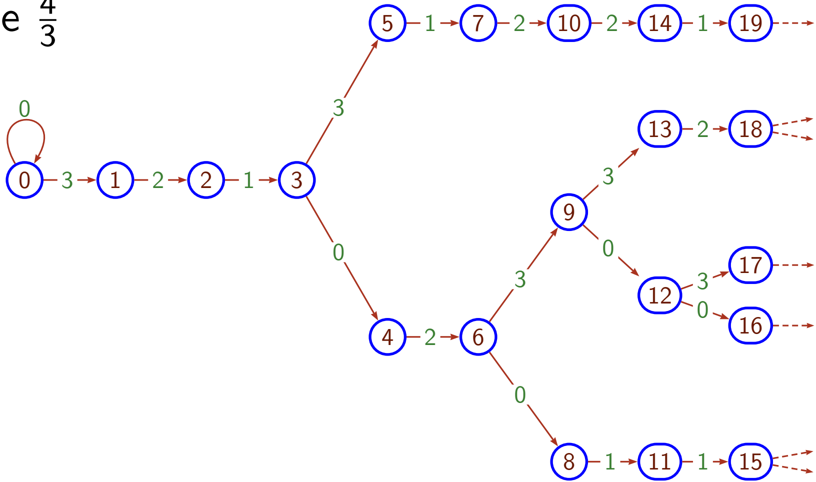
Étape 2: on applique

$$\begin{array}{lcl} 0 & \to & 1|0 \\ 1 & \to & 0|0,\ 1|1 \\ 2 & \to & 0|1 \end{array}$$



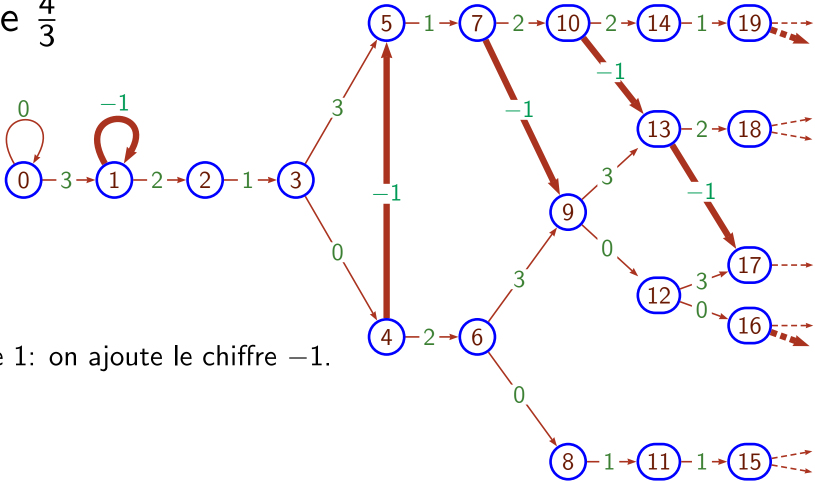

Base $\frac{3}{2}$

Base $\frac{4}{3}$



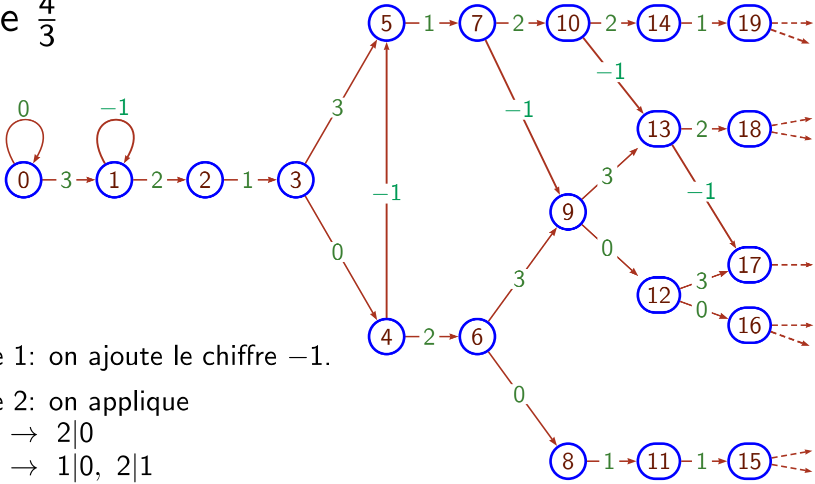
# Ex. 2: quand la base est petite ($p < 2q - 1$)

Base $\frac{4}{3}$




Étape 1: on ajoute le chiffre $-1$.

Base $\frac{4}{3}$




Étape 1: on ajoute le chiffre $-1$.

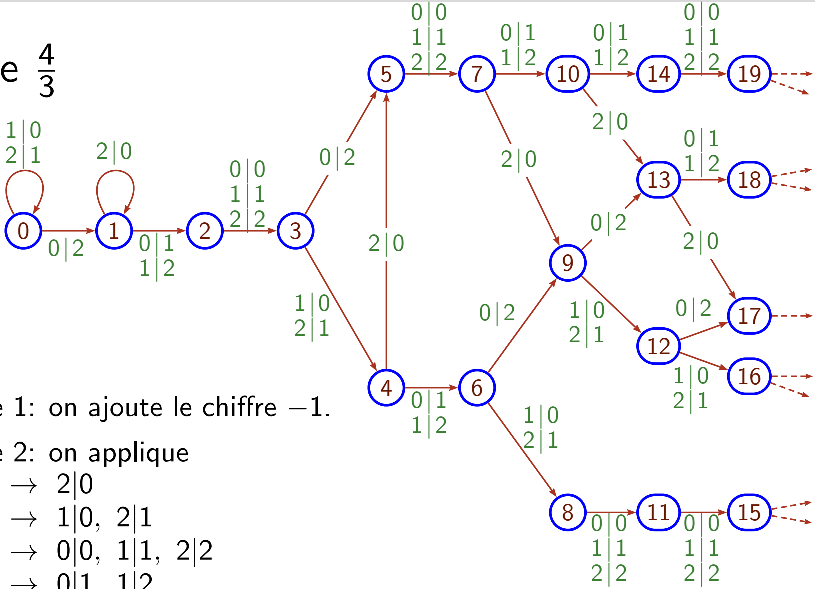
Étape 2: on applique

$$\begin{array}{ccl} -1 & \to & 2|0 \\ 0 & \to & 1|0,\ 2|1 \\ 1 & \to & 0|0,\ 1|1,\ 2|2 \\ 2 & \to & 0|1,\ 1|2 \\ 3 & \to & 0|2 \end{array}$$

# Ex. 2: quand la base est petite ($p < 2q - 1$)



## Base $\frac{4}{3}$




Étape 1: on ajoute le chiffre $-1$.

Étape 2: on applique

$-1 \rightarrow 2|0$

$0 \rightarrow 1|0,\ 2|1$

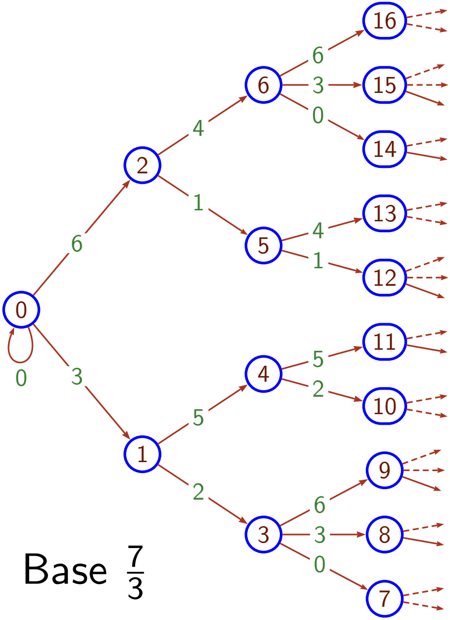
$1 \rightarrow 0|0,\ 1|1,\ 2|2$

$2 \rightarrow 0|1,\ 1|2$

$3 \rightarrow 0|2$

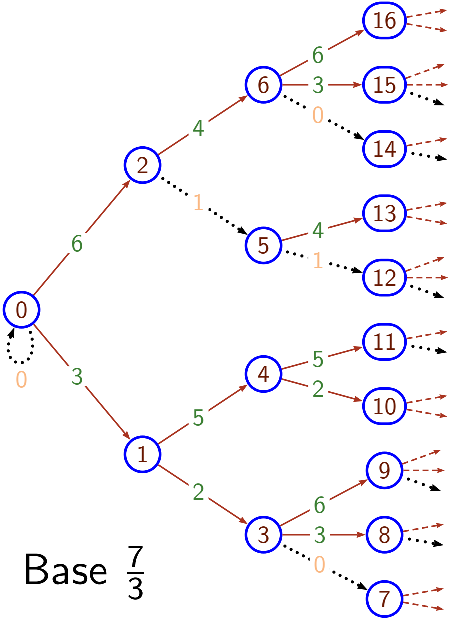
## Ex. 3: quand la base est grande ($p > 2q - 1$)






Base $\frac{7}{3}$

# Ex. 3: quand la base est grande ($p > 2q - 1$)

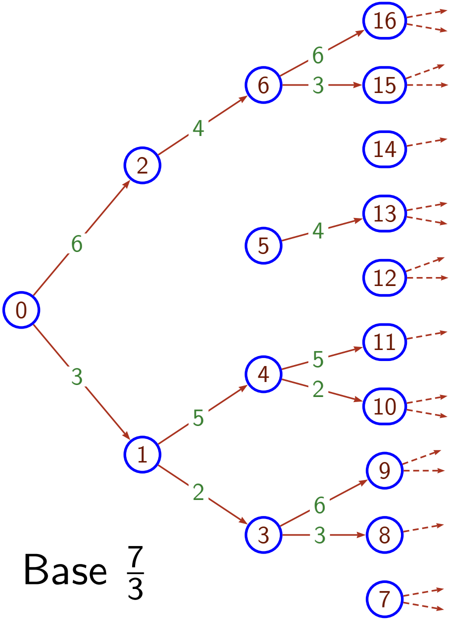



Base $\frac{7}{3}$

Étape 1: on supprime les 0 et 1.

# Ex. 3: quand la base est grande ($p > 2q - 1$)



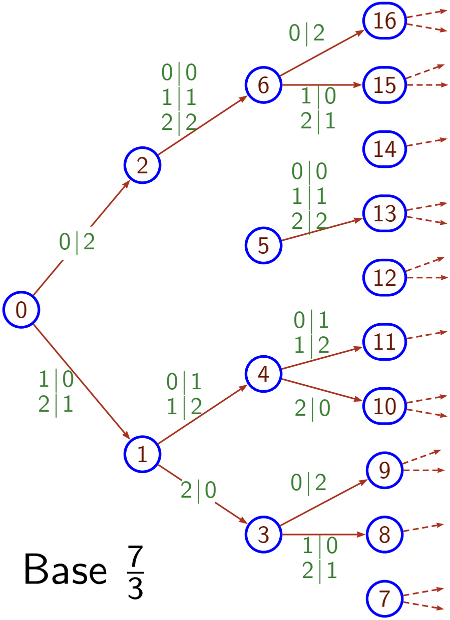



Base $\frac{7}{3}$

Étape 1: on supprime les 0 et 1.

Étape 2: on applique

$$\begin{array}{lcl} 2 & \to & 2|0 \\ 3 & \to & 1|0,\ 2|1 \\ 4 & \to & 0|0,\ 1|1,\ 2|2 \\ 5 & \to & 0|1,\ 1|2 \\ 6 & \to & 0|2 \end{array}$$




Base $\frac{7}{3}$

Étape 1: on supprime les 0 et 1.

Étape 2: on applique

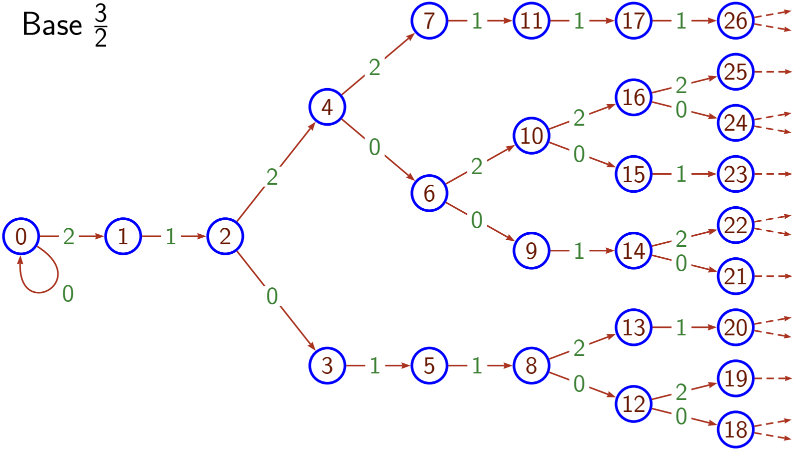
$$\begin{array}{lcl} 2 & \to & 2|0 \\ 3 & \to & 1|0,\ 2|1 \\ 4 & \to & 0|0,\ 1|1,\ 2|2 \\ 5 & \to & 0|1,\ 1|2 \\ 6 & \to & 0|2 \end{array}$$

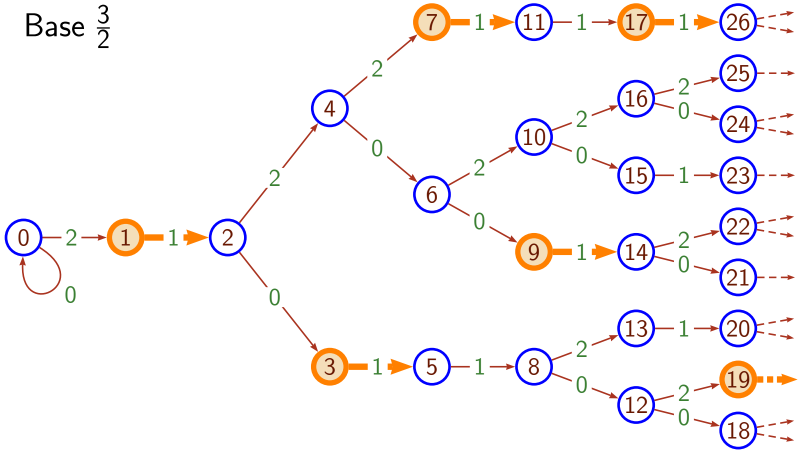
# Plan

# $L_{\frac{3}{2}}$ possède une régularité "transversale"



Base $\frac{3}{2}$

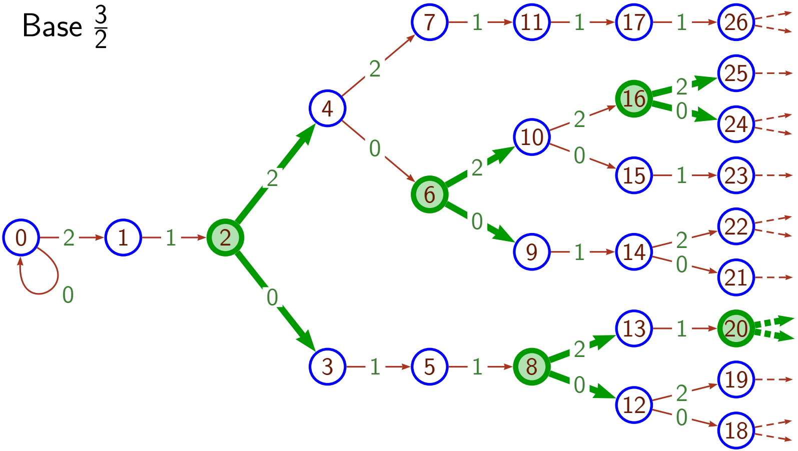





- Les nœuds impairs ont un arc sortant étiqueté par 1.

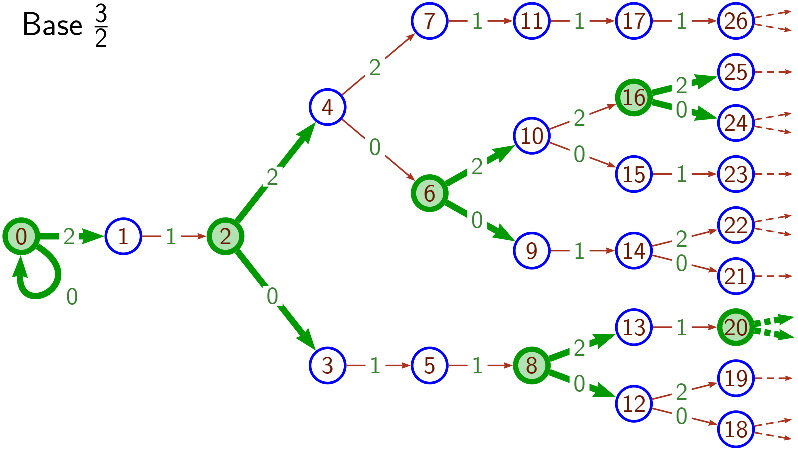
# $L_{\frac{3}{2}}$ possède une régularité “transversale”






- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

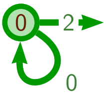
# Construire $L_{\frac{3}{2}}$ périodiquement avec $2, 1, 2, 1, \ldots$



Base $\frac{3}{2}$


- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

Base $\frac{3}{2}$

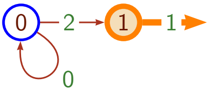



- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

## Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .

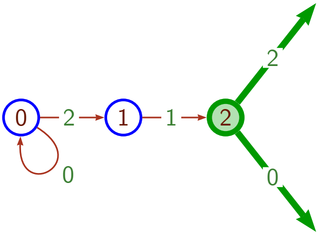


Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

Base $\frac{3}{2}$

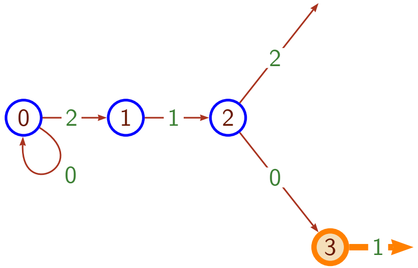



- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .



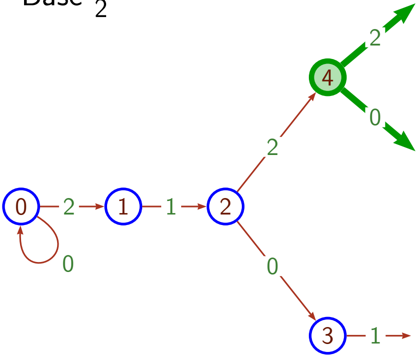
Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .

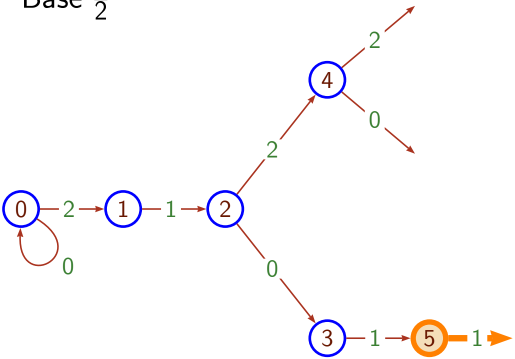
Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, ...

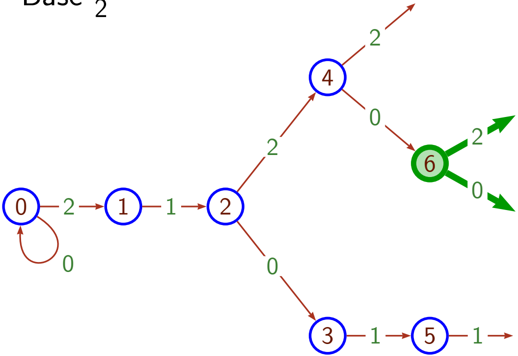
Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .

Base $\frac{3}{2}$

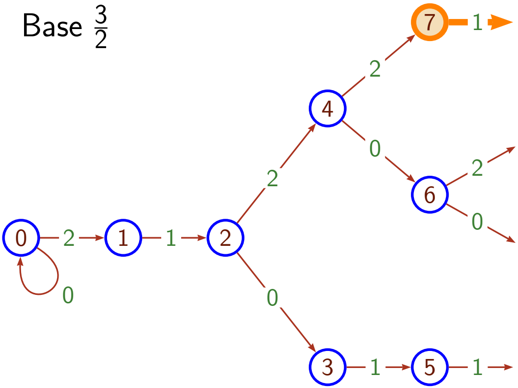



- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .



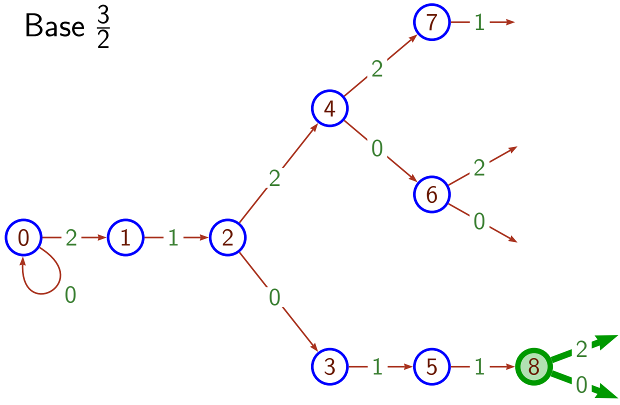
Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec $2, 1, 2, 1, \ldots$



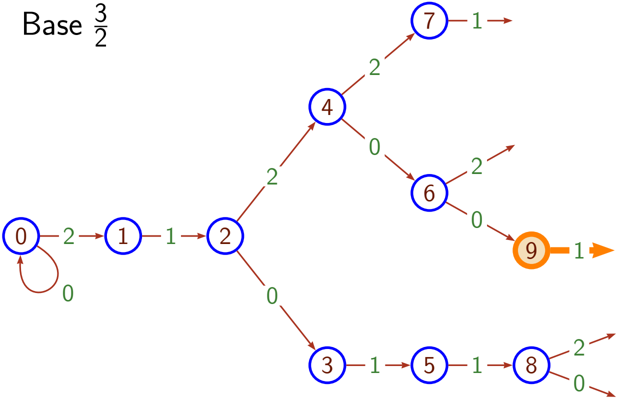



- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec $2, 1, 2, 1, \ldots$



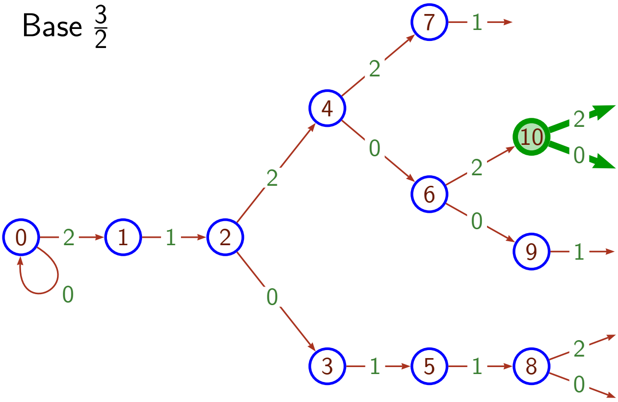
Base $\frac{3}{2}$




- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, ...

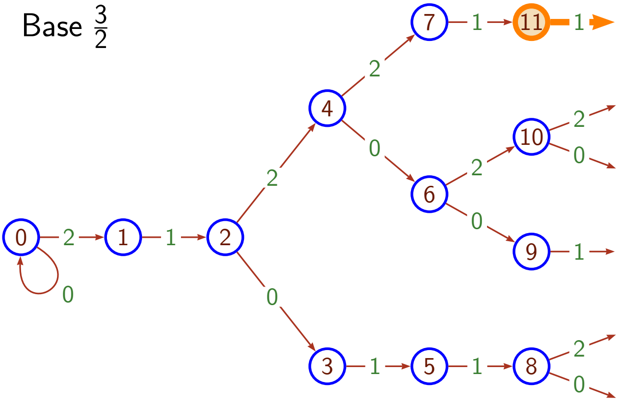





- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

## Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .

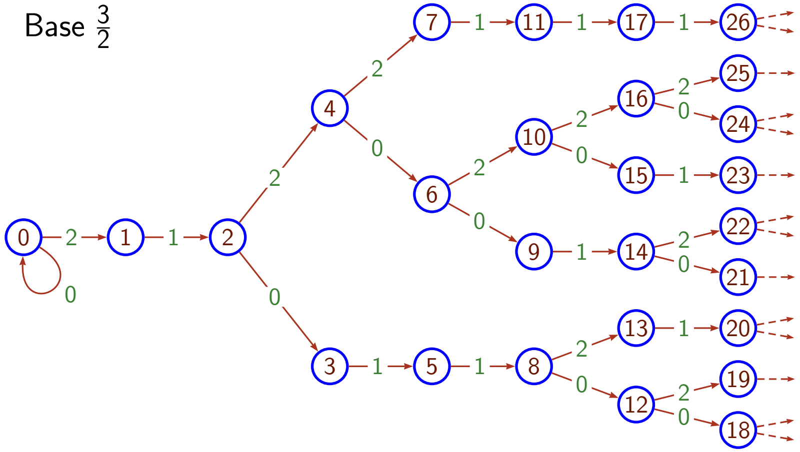





- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

# Construire $L_{\frac{3}{2}}$ périodiquement avec 2, 1, 2, 1, . . .

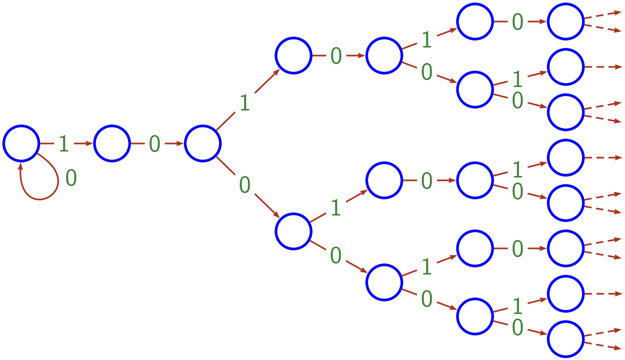





- Les nœuds impairs ont un arc sortant étiqueté par 1.
- Les nœuds pairs ont deux arcs sortants étiquetés par 0 et 2.

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.

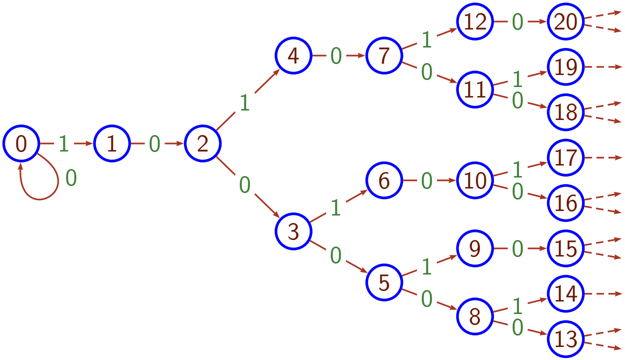


# Signature et étiquetage d'un arbre étiqueté



$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.

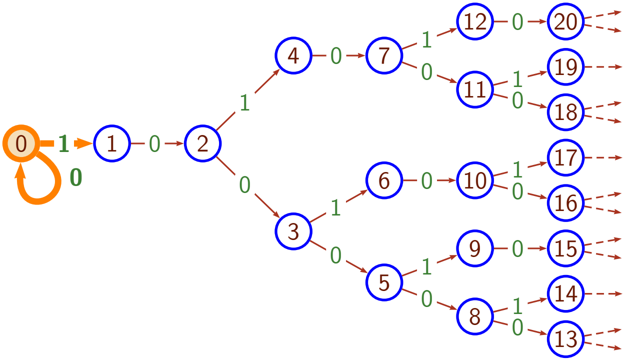



- Signature : $\boldsymbol{s}$ $=$
- Étiquetage : $\boldsymbol{\lambda}$ $=$

# Signature et étiquetage d'un arbre étiqueté

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.



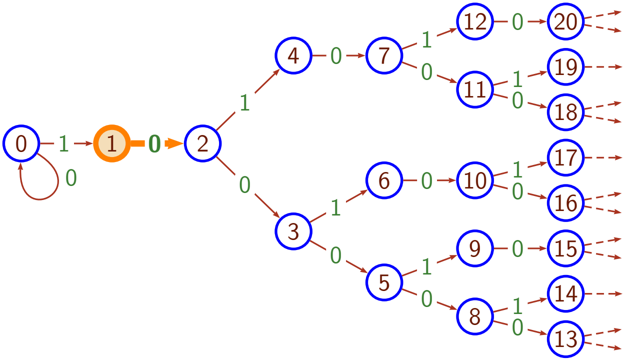

- Signature : $\boldsymbol{s} = 2$
- Étiquetage : $\boldsymbol{\lambda} = 01$

# Signature et étiquetage d'un arbre étiqueté



$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.



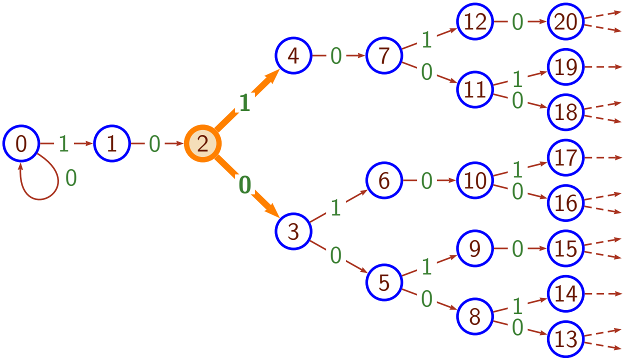

- Signature : $\boldsymbol{s} = 2\ 1$
- Étiquetage : $\boldsymbol{\lambda} = 01\ 0$

# Signature et étiquetage d'un arbre étiqueté



$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

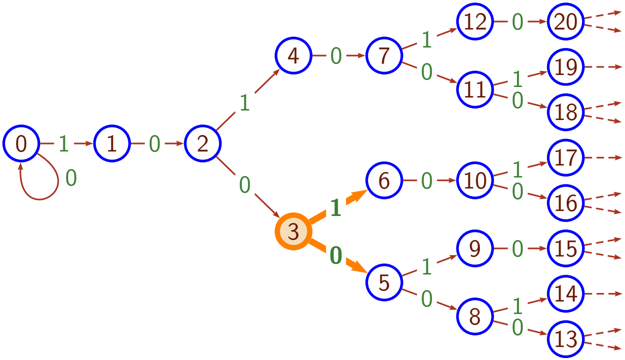
L'ordre de $A$ induit un parcours en largeur canonique de $T$.




- Signature : $\boldsymbol{s} \;=\; 2\ 1\ 2$
- Étiquetage : $\boldsymbol{\lambda} \;=\; 01\ 0\ 01$

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

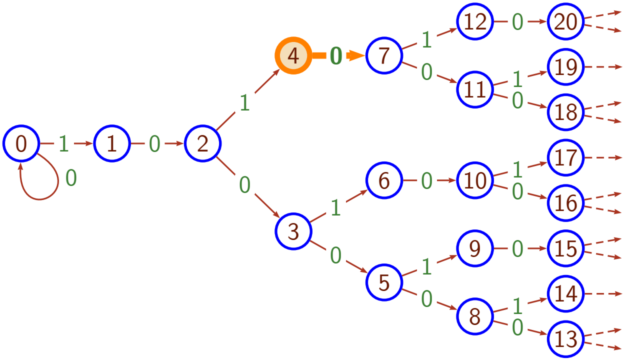
L'ordre de $A$ induit un parcours en largeur canonique de $T$.




- Signature : $\boldsymbol{s} \;=\; 2\ 1\ 2\ \boxed{2}$
- Étiquetage : $\boldsymbol{\lambda} \;=\; 01\ 0\ 01\ \boxed{01}$

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.

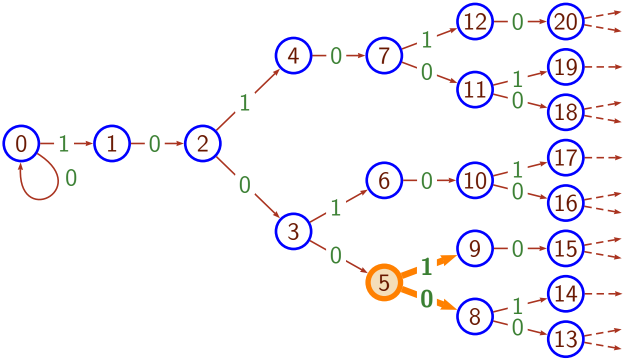



- Signature : $\boldsymbol{s} \;=\; 2\ 1\ 2\ 2\ 1$
- Étiquetage : $\boldsymbol{\lambda} \;=\; 01\ 0\ 01\ 01\ 0$

# Signature et étiquetage d'un arbre étiqueté

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

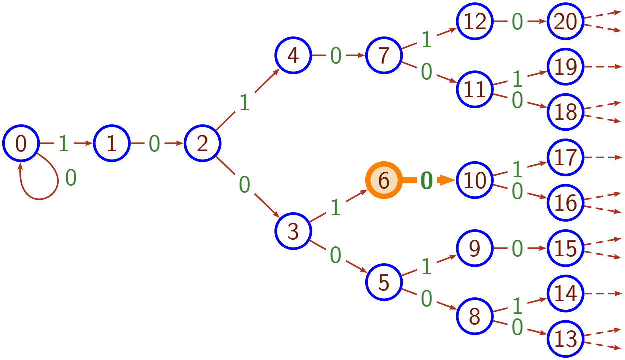
L'ordre de $A$ induit un parcours en largeur canonique de $T$.




- Signature : $\boldsymbol{s} \ = \ 2\ 1\ 2\ 2\ 1\ 2$
- Étiquetage : $\boldsymbol{\lambda} \ = \ 01\ 0\ 01\ 01\ 0\ 01$

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

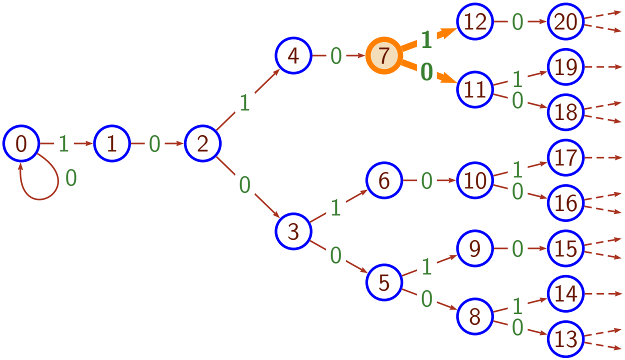
L'ordre de $A$ induit un parcours en largeur canonique de $T$.




- Signature : $\boldsymbol{s} \; = \; 2\;1\;2\;2\;1\;2\;\boxed{1}$
- Étiquetage : $\boldsymbol{\lambda} \; = \; 01\;0\;01\;01\;0\;01\;\boxed{0}$

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

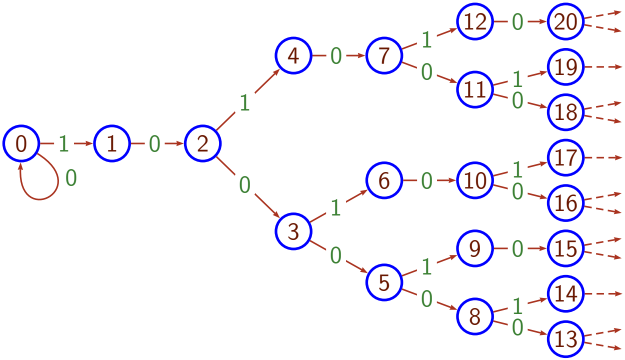
L'ordre de $A$ induit un parcours en largeur canonique de $T$.




- Signature : $\boldsymbol{s} \; = \; 2\ 1\ 2\ 2\ 1\ 2\ 1\ 2$
- Étiquetage : $\boldsymbol{\lambda} \; = \; 01\ 0\ 01\ 01\ 0\ 01\ 0\ 01$

$A$: un alphabet totalement ordonné.
$T$: un i-arbre étiqueté sur $A$ (ou langage calable, clos par préfixe).

L'ordre de $A$ induit un parcours en largeur canonique de $T$.



- Signature : $\boldsymbol{s} \;=\; 2\ 1\ 2\ 2\ 1\ 2\ 1\ 2\ 2\ 1\ 2\ 2\ 1\ \cdots$
- Étiquetage : $\boldsymbol{\lambda} \;=\; 01\ 0\ 01\ 01\ 0\ 01\ 0\ 01\ 01\ 0\ 01\ 01\ 0\ \cdots$

# Signatures périodiques



**Théorème**

$\frac{p}{q}$: une base entière ou rationnelle.

Le Langage $0^* L_{\frac{p}{q}}$ a

- pour signature le mot périodique $(\boldsymbol{r}_{\frac{p}{q}})^\omega$,

  où $\boldsymbol{r}_{\frac{p}{q}}$ est le *rythme de Christoffel* de pente $\frac{p}{q}$ ;

- et pour étiquetage le mot périodique $(\boldsymbol{\gamma}_{\frac{p}{q}})^\omega$, où

$$\gamma_{\frac{p}{q}} = (0, (q \mod p), (2\,q \mod p), \ldots, ((p-1)\,q) \mod p) .$$

**Théorème**

$\forall \boldsymbol{r}$: un mot fini sur un alphabets de chiffres.
$\exists \gamma_{\boldsymbol{r}}$: l'*étiquetage spécial* associé à $\boldsymbol{r}$.

$0^*L$: le langage calable de signature $\boldsymbol{r}^\omega$ et d'étiquetage $(\gamma_{\boldsymbol{r}})^\omega$.
$\frac{p}{q}$: la moyenne des lettres de $\boldsymbol{r}$.

Alors, $L$ est une représentation des entiers en base $\frac{p}{q}$ sur un alphabet (fini) non-canonique.

### Théorème

$z^*L$: un langage calable et clos-par-préfixe.

$L$ est régulier $\iff$ sa signature étiquetée est *s-morphique*$^\dagger$.

La démonstration utilise une transformation

morphismes de mots $\longleftrightarrow$ automates

que l'on retrouve dans les travaux de

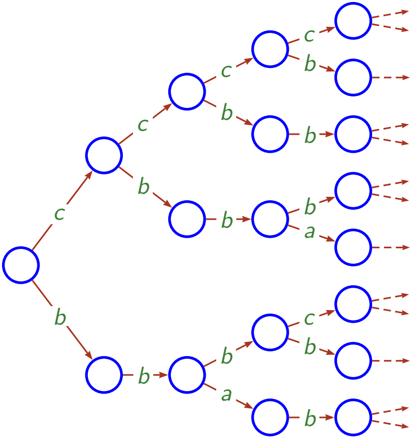
- Dumont et Thomas (1989, 1991, 1993);
- Rigo et Maes (2002).

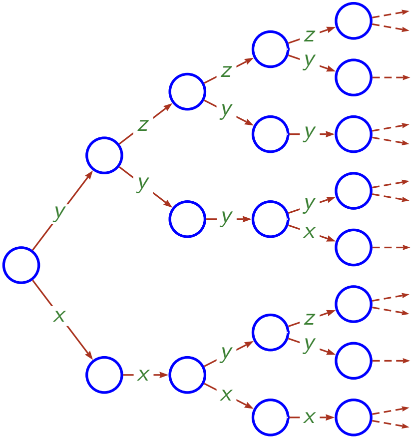
$\dagger$ Une signature étiquetée $(\boldsymbol{s}, \boldsymbol{\lambda})$ est s-morphique si

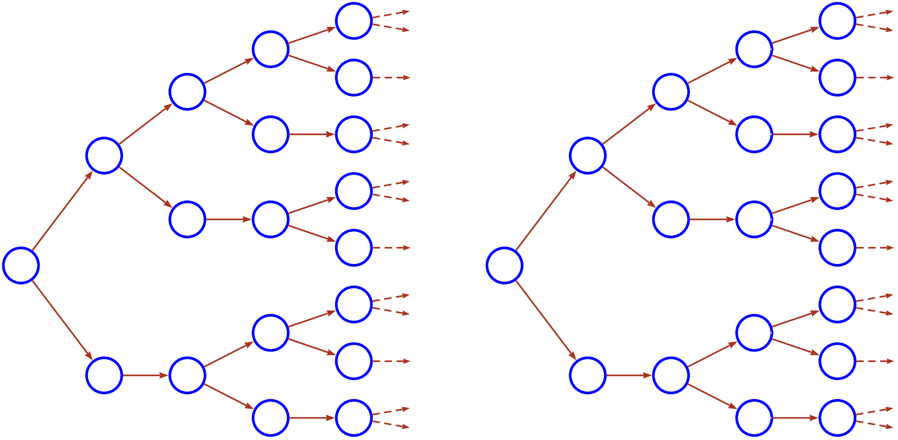
$$\boldsymbol{s} = f_\sigma(\sigma^\omega(a)) \qquad \text{et} \qquad \boldsymbol{\lambda} = g(\sigma^\omega(a))$$

- $\sigma$ est un endormorphisme de mots;
- $f_\sigma$ est le morphisme défini par: pour tout $b$, $f_\sigma(b) = |\sigma(b)|$ ;
- $g$ est un morphisme vérifiant: pour tout $b$, $|g(b)| = |\sigma(b)|$ .

# Oublier les étiquettes

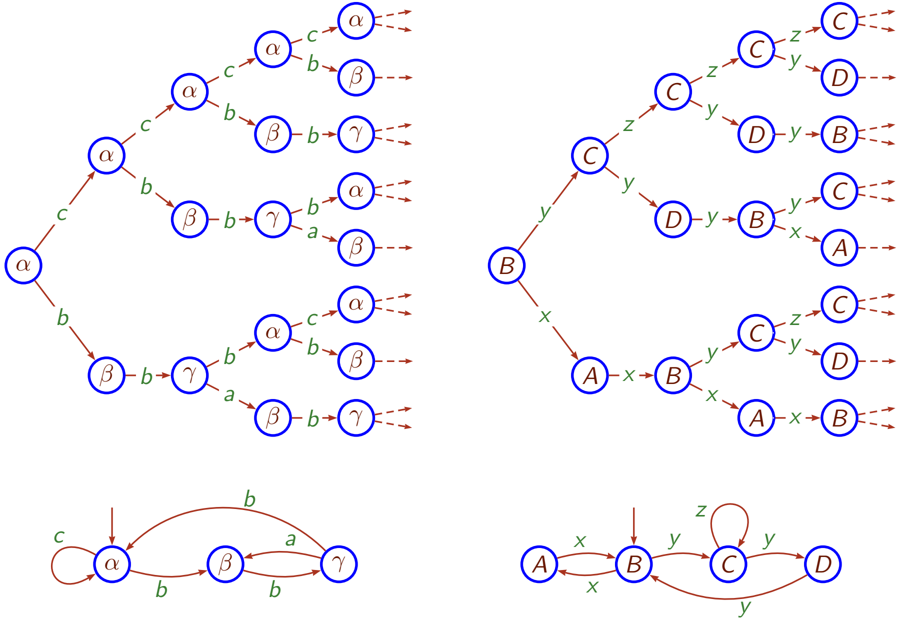




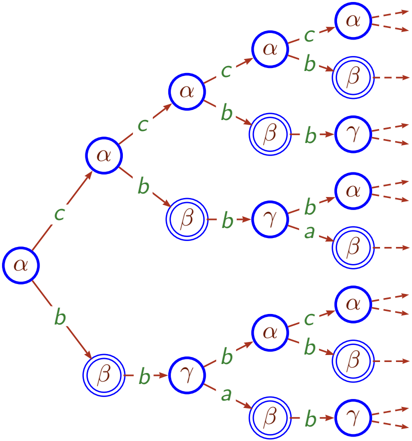
# Oublier les étiquettes

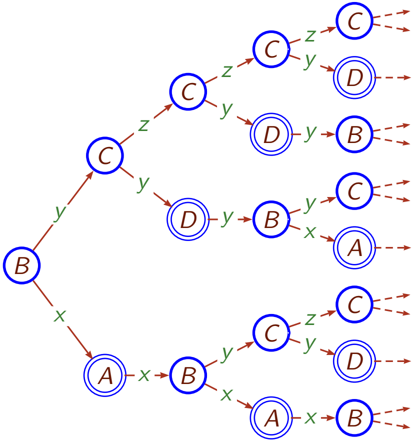


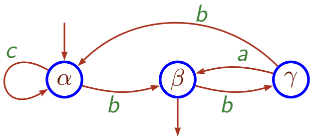


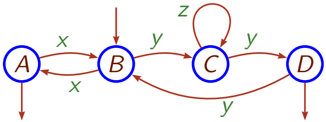
# Oublier les étiquettes





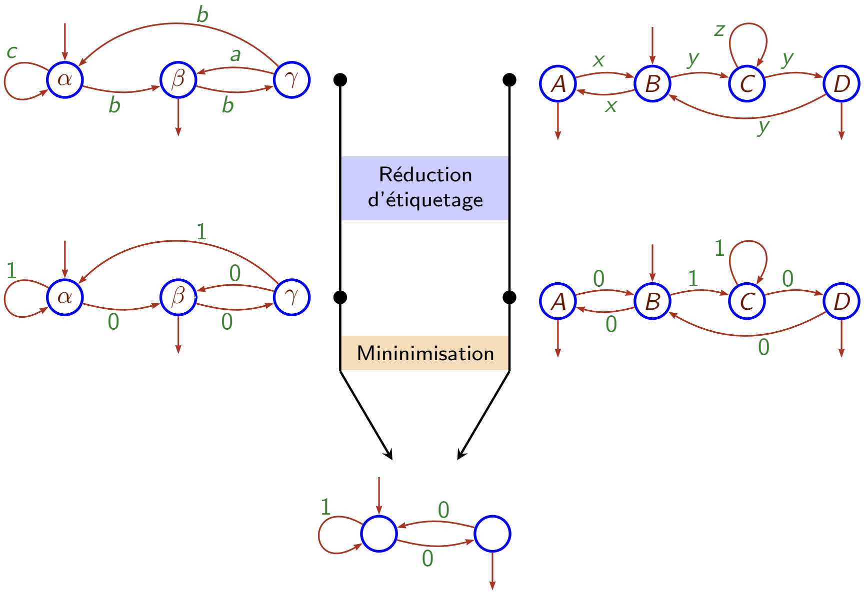



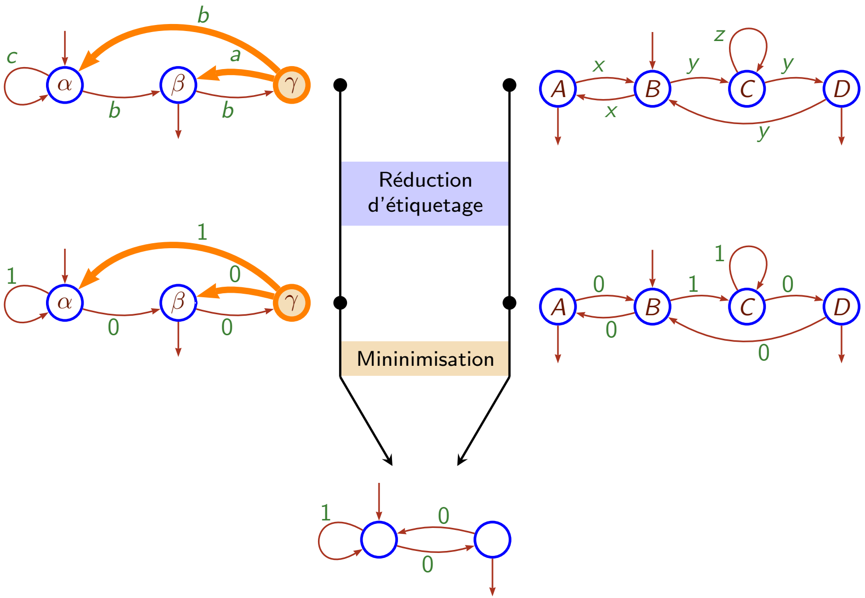
# Surminimisation



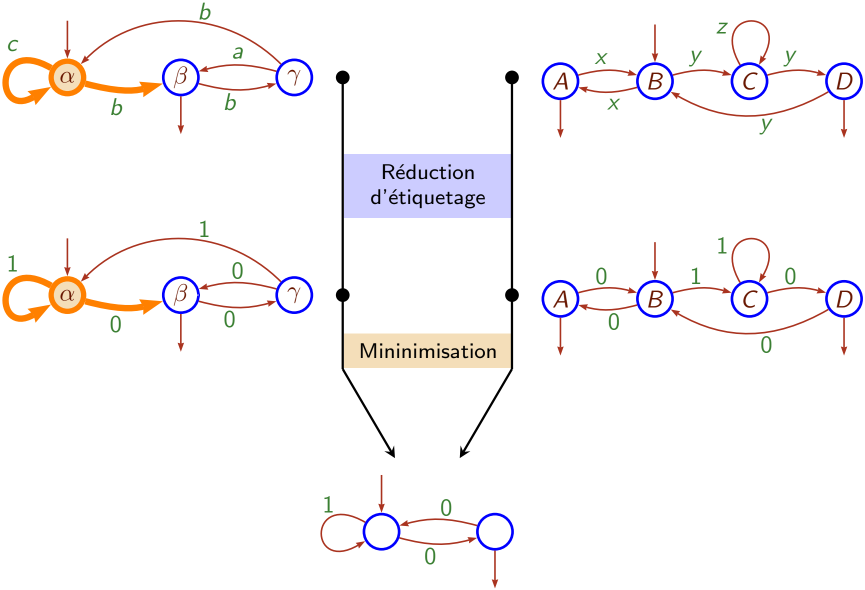


# Surminimisation





# Surminimisation



# Plan

# La base entière



Un entier $p > 1$

Alphabet: $A_p \;=\; \{0, 1, \dots (p-1)\}$

## Représentation (par divisions euclidiennes successives)

$$\langle n \rangle \;=\; \langle n' \rangle \, a \quad ,$$

où, $(n', a)$ est le résultat de la div. euclidienne de $n$ par $p$.

## Évaluation

$$\pi(a_k \cdots a_1 \, a_0) \;=\; (a_k \times p^k) + \cdots + (a_1 \times p^1) + a_0.$$

# La base entière



Un entier $p > 1$
Alphabet: $A_p \;=\; \{0, 1, \ldots (p-1)\}$

## Représentation (par divisions euclidiennes successives)

$$\langle n \rangle \;=\; \langle n' \rangle \, a \quad ,$$

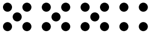
où, $(n', a)$ est le résultat de la div. euclidienne de $n$ par $p$.

## Évaluation

$$\pi(a_k \cdots a_1 \, a_0) \;=\; (a_k \times p^k) + \cdots + (a_1 \times p^1) + a_0 .$$

## Convention (LSDF)

On écrit le chiffre de poids faible en premier.

Exemple: La quantité  s'écrit donc 91 (au lieu de 19).

# La base entière



Un entier $p > 1$
Alphabet: $A_p \;=\; \{0, 1, \ldots (p-1)\}$

## Représentation (par divisions euclidiennes successives)

$$\langle n \rangle \;=\; a \, \langle n' \rangle \quad ,$$

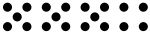
où, $(n', a)$ est le résultat de la div. euclidienne de $n$ par $p$.

## Évaluation

$$\pi(a_0 \, a_1 \;\cdots\; a_k) \;=\; (a_k \times p^k) + \cdots + (a_1 \times p^1) + a_0.$$

## Convention (LSDF)

On écrit le chiffre de poids faible en premier.

Exemple: La quantité  s'écrit donc 91 (au lieu de 19).

# Langage simple, ensemble simple



## Langage régulier

Un langage est dit *régulier* s'il est accepté par un automate.

Un tel langage est considéré *simple*.

# Langage simple, ensemble simple

## Langage régulier

Un langage est dit *régulier* s'il est accepté par un automate.

Un tel langage est considéré *simple*.

## Ensemble $p$-reconnaissable

$p$: une base entière.
$E$: un ensemble d'entiers.

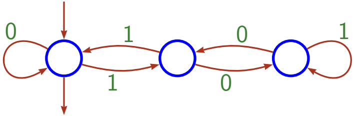
$$E \text{ est } p\text{-reconnaissable} \iff \langle E \rangle \text{ est régulier}$$

Un tel ensemble est considéré *simple* relativement à $p$.

## Théorème (Pascal?)

Les ensembles ultimement périodiques d'entiers sont *p*-reconnaissables pour toute base *p*.

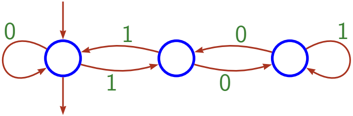
Exemple: l'ensemble des entiers divisibles par 3, écrits en base 2:



### Théorème (Pascal?)

Les ensembles ultimement périodiques d'entiers sont $p$-reconnaissables pour toute base $p$.

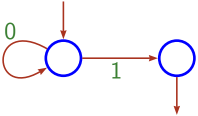
Exemple: l'ensemble des entiers divisibles par 3, écrits en base 2:




### Théorème (?)

Il existe des ensembles qui sont $p$-reconnaissables pour certaines bases $p$ mais pas pour d'autres.

Exemple: l'ensemble des puissances de 10, écrits en base 10:



# Ensembles $p$-reconnaissables (2)



### Théorème (Cobham, 1969)

$p, p'$: deux bases entières multiplicativement indépendantes†.
$E$: un ensemble d'entiers.

Si $E$ est à la fois $p$- et $p'$-reconnaissable, alors $E$ est ultimement périodique.

### Corollaire

Si un ensemble d'entiers $E$ est $p$-reconnaissable dans toutes les bases $p$, alors $E$ est ultimement périodique.

† Deux entiers $b$ et $b'$ sont multiplicativement indépendants si aucune puissance de l'un n'est égale à une puissance de l'autre.

# Le problème ULTIME PÉRIODICITÉ



## Donnée

- une base $p$
- un automate $\mathcal{A}$

## Question

Le langage $L(\mathcal{A})$ est-il la représentation, en base $p$, d'un ensemble ultimement périodique d'entiers ?

# Solutions successives apportées (1)



**Théorème (Honkala, 1986)**

Le problème ULTIME PÉRIODICITÉ est décidable.

# Solutions successives apportées (1)



**Théorème (Honkala, 1986)**

Le problème ULTIME PÉRIODICITÉ est décidable.

**Théorème (Muchnik, 1991)**

Le problème SOUS-ENSEMBLE RATIONNEL† est décidable.

**Théorème (Leroux, 2005)**

Le problème SOUS-ENSEMBLE RATIONNEL† est décidable en complexité quadratique.

† SOUS-ENSEMBLE RATIONNEL est une généralisation d'ULTIME PÉRIODICITÉ: on considère un ensemble de *d*-uplets d'entiers.

# Solutions successives apportées (2)



### Théorème (Allouche, Rampersad, Shallit, 2009)

Le problème ULTIME PÉRIODICITÉ GÉNÉRALISÉ‡ est décidable en complexité exponentielle pour les U-systèmes dans lequel l'addition est réalisable par un automate.

### Théorème (Bell, Charlier, Fraenkel, Rigo, 2009)

Le problème ULTIME PÉRIODICITÉ GÉNÉRALISÉ‡ est décidable pour une large classe de $U$-systèmes.

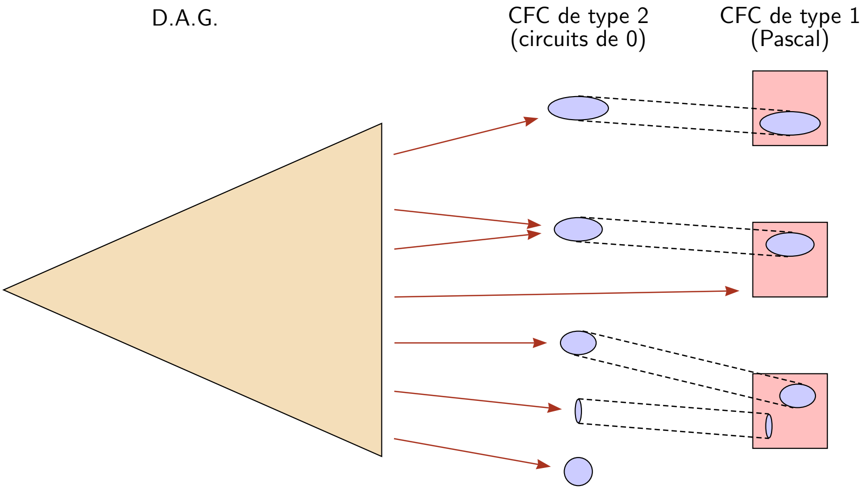
‡ Ce problème généralise d'ULTIME PÉRIODICITÉ à d'autres sytèmes de numérations.

**Théorème**

Le problème ULTIME PÉRIODICITÉ est décidable en complexité quasi-linéaire.

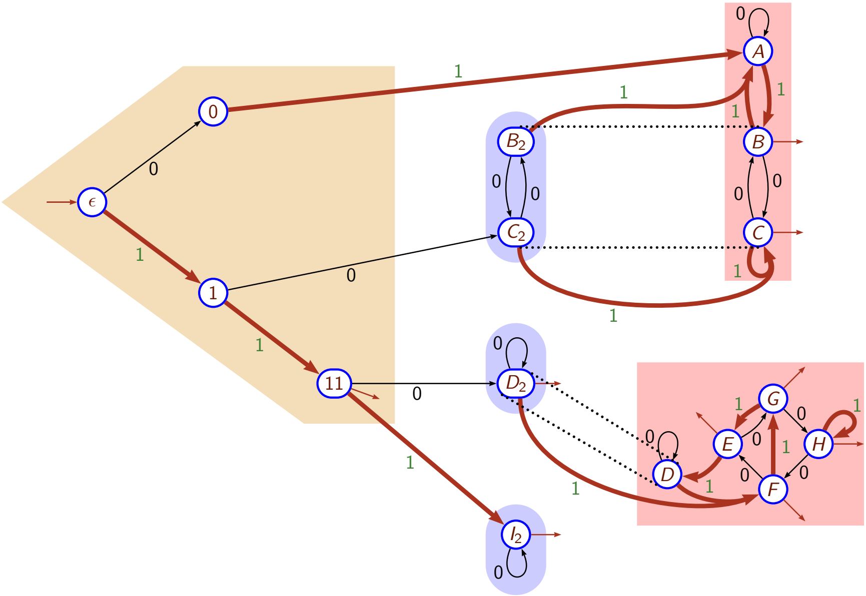
# La critère (UP)





# Un automate satisfaisant le critère (UP)





# Plan

# Perspectives



## Sur Ultime périodicité en base entière

- En utilisant la convention LSDF : quasi-linéaire.
- Sinon : exponentielle [ARS 2009].
- Pour des problèmes plus généraux :
  - en dimension supérieure : quadratique [L 2005] ;
  - aux U-systèmes : exponentielle [ARS 2009] ;
  - aux SNA réguliers : décidable [D 2013, M 2011].

## Sur les mots minimaux en base rationnelle

- La fonction successeur : transducteur infini proche de $L_{\frac{p}{q}}$.
- Si restreinte à une seule entrée : inconnu, transducteur fini?

# Perspectives (2)



## Sur les signatures

- Signatures s-morphique $\leftrightarrow$ SNA réguliers.
- Signatures périodique $\leftrightarrow$ bases entières et rationnelles.
- Classification des systèmes de numérations?
  - ? $\leftrightarrow$ U-systèmes.
  - ? $\leftrightarrow$ SNA algébrique.

## Sur la surminimisation

- Réduit le nombre d'états davantage que la minimisation.
- Conserve uniquement l'arbre ordonné sous-jacent.
- Induit une "forme normale" sur les SNA réguliers.