A propos de la relation k-binomiale

11 décembre 2019

Marie Lejeune

A propos de la relation k-binomiale

A propos de la relation k-binomiale

Alphabets et mots

Un **alphabet** est juste un ensemble fini dont les éléments sont appelés des lettres.
Un **mot (fini)** est une suite (finie) de lettres que l'on concatène.

Par exemple, $abacaba$ et $cababababab \cdots$ sont des mots sur l'alphabet $\{a, b, c\}$.

Alphabets et mots

Un **alphabet** est juste un ensemble fini dont les éléments sont appelés des lettres.
Un **mot (fini)** est une suite (finie) de lettres que l'on concatène.

Par exemple, $abacaba$ et $cabababab\cdots$ sont des mots sur l'alphabet $\{a, b, c\}$.

Un **sous-mot** du mot $u = u_1u_2\cdots u_m$ est une sous-suite finie de la suite $(u_j)_{j=1}^m$. Il est appelé **facteur** si la suite est constituée de termes consécutifs.

Par exemple, acb est un sous-mot de $u = abacaba$, mais pas un facteur.

Un **alphabet** est juste un ensemble fini dont les éléments sont appelés des lettres.
Un **mot (fini)** est une suite (finie) de lettres que l'on concatène.

Par exemple, $abacaba$ et $cababababab\cdots$ sont des mots sur l'alphabet $\{a, b, c\}$.

Un **sous-mot** du mot $u = u_1 u_2 \cdots u_m$ est une sous-suite finie de la suite $(u_j)_{j=1}^m$. Il est appelé **facteur** si la suite est constituée de termes consécutifs.

Par exemple, acb est un sous-mot de $u = abacaba$, mais pas un facteur. Le mot $acab$ est un facteur de u, donc aussi un sous-mot.

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = ?$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 1$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

> On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.
>
> La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = ?.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 1.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 2.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 3.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 4.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 5.$$

Nombre d'occurrences des facteurs et sous-mots

Un même mot peut apparaitre plusieurs fois comme sous-mot d'un autre mot. On peut donc compter combien de fois il apparait.

On note $\binom{\mathbf{u}}{\mathbf{x}}$ le nombre de fois que le mot x apparait comme sous-mot dans u et $|\mathbf{u}|_{\mathbf{x}}$ le nombre de fois qu'il apparait comme facteur dans u.

La quantité $\binom{u}{x}$ est appelée le **coefficient binomial** de u et x.

Considérons par exemple le mot $u = aababa$.

$$|u|_{ab} = 2 \quad \text{et} \quad \binom{u}{ab} = 5.$$

Une façon efficace de calculer un coefficient binomial

$$\binom{aababa}{aba} =$$

Une façon efficace de calculer un coefficient binomial

$$\binom{aababa}{aba} = \binom{ababa}{aba}$$

Une façon efficace de calculer un coefficient binomial

$$\binom{\color{red}{a}ababa}{\color{red}{a}ba} = \binom{ababa}{aba} + \binom{ababa}{ba}$$

Une façon efficace de calculer un coefficient binomial

$$\binom{aababa}{aba} = \binom{\color{lightgray}{a}\color{black}{baba}}{aba} + \binom{ababa}{ba}$$
$$= \binom{baba}{aba}$$

Une façon efficace de calculer un coefficient binomial

$$\binom{aababa}{aba} = \binom{\color{red}{a}baba}{\color{red}{a}ba} + \binom{ababa}{ba}$$
$$= \binom{baba}{aba} + \binom{baba}{ba}$$

Une façon efficace de calculer un coefficient binomial

$$\binom{aababa}{aba} = \binom{ababa}{aba} + \binom{ababa}{ba}$$
$$= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{\color{red}{a}baba}{\color{blue}{b}a} \\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$
\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{\textcolor{lightgray}{b}aba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba}
\end{aligned}
$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\ &= \binom{\color{blue}{b}aba}{\color{red}{a}ba} + \binom{baba}{ba} + \binom{baba}{ba} \\ &= \binom{aba}{aba}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\ &= \binom{aba}{aba} + 2\binom{aba}{ba}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\ &= \binom{baba}{aba} + \binom{\textcolor{blue}{b}aba}{\textcolor{blue}{b}a} + \binom{\textcolor{blue}{b}aba}{\textcolor{blue}{b}a} \\ &= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\ &= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\ &= \binom{ba}{aba}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba}\\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba}\\ &= \binom{{\color{red}a}ba}{{\color{red}a}ba} + 2\binom{aba}{ba} + 2\binom{aba}{a}\\ &= \binom{ba}{aba} + \binom{ba}{ba}\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba}
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{{\color{red}a}ba}{{\color{blue}b}a} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba}
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{\color{red}{a}b\color{red}{a}}{\color{red}{a}} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2 \cdot 2
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$
\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2 \cdot 2 \\
&= 0
\end{aligned}
$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{\textcolor{lightgray}{b}a}{ba} + 2\binom{\textcolor{lightgray}{b}a}{ba} + 2\cdot 2 \\
&= 0 + 3\binom{a}{ba}
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{{\color{blue}b}a}{{\color{blue}b}a} + 2\binom{{\color{blue}b}a}{{\color{blue}b}a} + 2 \cdot 2 \\
&= 0 + 3\binom{a}{ba} + 3\binom{a}{a}
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$
\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2 \cdot 2 \\
&= 0 + 3\binom{a}{ba} + 3\binom{a}{a} + 4
\end{aligned}
$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2 \cdot 2 \\
&= 0 + 3\binom{a}{ba} + 3\binom{a}{a} + 4 \\
&= 0 + 3 \cdot 0 + 3 \cdot 1 + 4 = 7
\end{aligned}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba}\\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba}\\ &= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a}\\ &= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2\cdot 2\\ &= 0 + 3\binom{a}{ba} + 3\binom{a}{a} + 4\\ &= 0 + 3\cdot 0 + 3\cdot 1 + 4 = 7\end{aligned}$$

Donc

$$\binom{\ell_1 u}{\ell_2 v} = \binom{u}{\ell_2 v}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}
\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\
&= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\
&= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\
&= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2 \cdot 2 \\
&= 0 + 3\binom{a}{ba} + 3\binom{a}{a} + 4 \\
&= 0 + 3 \cdot 0 + 3 \cdot 1 + 4 = 7
\end{aligned}$$

Donc

$$\binom{\ell_1 u}{\ell_2 v} = \binom{u}{\ell_2 v} + \delta_{\ell_1,\ell_2}\binom{u}{v}$$

Une façon efficace de calculer un coefficient binomial

$$\begin{aligned}\binom{aababa}{aba} &= \binom{ababa}{aba} + \binom{ababa}{ba} \\ &= \binom{baba}{aba} + \binom{baba}{ba} + \binom{baba}{ba} \\ &= \binom{aba}{aba} + 2\binom{aba}{ba} + 2\binom{aba}{a} \\ &= \binom{ba}{aba} + \binom{ba}{ba} + 2\binom{ba}{ba} + 2\cdot 2 \\ &= 0 + 3\binom{a}{ba} + 3\binom{a}{a} + 4 \\ &= 0 + 3\cdot 0 + 3\cdot 1 + 4 = 7\end{aligned}$$

Donc

$$\binom{\ell_1 u}{\ell_2 v} = \binom{u}{\ell_2 v} + \delta_{\ell_1,\ell_2}\binom{u}{v}$$

avec les cas de base $\binom{u}{v} = 0$ si $|u| < |v|$ et $\binom{u}{\ell} = |u|_\ell$.

Différentes relations d'équivalence

Soient u et v deux mots finis. On définit plusieurs relations d'équivalence :

Différentes relations d'équivalence

Soient u et v deux mots finis. On définit plusieurs relations d'équivalence :

- l'égalité: $u \sim_= v \Leftrightarrow u = v$

Différentes relations d'équivalence

Soient u et v deux mots finis. On définit plusieurs relations d'équivalence :

- l'égalité: $u \sim_{=} v \Leftrightarrow u = v$
- l'équivalence abélienne : $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$

Différentes relations d'équivalence

Soient u et v deux mots finis. On définit plusieurs relations d'équivalence :

- l'égalité: $u \sim_{=} v \Leftrightarrow u = v$
- l'équivalence abélienne : $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$
- l'équivalence k-abélienne ($k \in \mathbb{N}$) : $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$

Soient u et v deux mots finis. On définit plusieurs relations d'équivalence :

- l'égalité: $u \sim_= v \Leftrightarrow u = v$
- l'équivalence abélienne : $u \sim_{ab,1} v \Leftrightarrow |u|_a = |v|_a \ \forall a \in A$
- l'équivalence k-abélienne ($k \in \mathbb{N}$) : $u \sim_{ab,k} v \Leftrightarrow |u|_x = |v|_x \ \forall x \in A^{\leq k}$
- l'équivalence k-binomiale ($k \in \mathbb{N}$) : $u \sim_k v \Leftrightarrow \binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \; \forall x \in A^{\leq k}.$$

L'équivalence k-binomiale

Définition (Rappel)
Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents.
En effet,

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 1 = \binom{v}{a}$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \; \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 1 = \binom{v}{b}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents.
En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 2 = \binom{v}{b}$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 3 = \binom{v}{b}$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa}$$

L'équivalence k-binomiale

Définition (Rappel)
Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa}, \binom{u}{bb} = 6 = \binom{v}{bb}$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 1 = \binom{v}{ab}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 2 = \binom{v}{ab}$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 3 = \binom{v}{ab}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 1 = \binom{v}{ba}.$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 2 = \binom{v}{ba}.$$

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 3 = \binom{v}{ba}.$$

Définition (Rappel)
Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 4 = \binom{v}{ba}.$$

L'équivalence k-binomiale

Définition (Rappel)

Soient u et v deux mots finis. Ils sont **k-binomialement équivalents** si

$$\binom{u}{x} = \binom{v}{x} \ \forall x \in A^{\leq k}.$$

Les mots $u = bbaabb$ et $v = babbab$ sont 2-binomialement équivalents. En effet,

$$\binom{u}{a} = 2 = \binom{v}{a}, \binom{u}{b} = 4 = \binom{v}{b}, \binom{u}{aa} = 1 = \binom{v}{aa},$$
$$\binom{u}{bb} = 6 = \binom{v}{bb}, \binom{u}{ab} = 4 = \binom{v}{ab}, \binom{u}{ba} = 4 = \binom{v}{ba}.$$

Complexité factorielle

Soit **w** un mot infini. Une fonction de complexité de **w** est une application liant chaque naturel n avec les facteurs de longueur n du mot **w**.

Complexité factorielle

Soit $\mathbf{w}$ un mot infini. Une fonction de complexité de $\mathbf{w}$ est une application liant chaque naturel n avec les facteurs de longueur n du mot $\mathbf{w}$.

Definition

La **complexité factorielle** du mot $\mathbf{w}$ est la fonction

$$p_{\mathbf{w}} : \mathbb{N} \to \mathbb{N} : n \mapsto \#\mathrm{Fac}_{\mathbf{w}}(n).$$

Complexité factorielle

Soit $\mathbf{w}$ un mot infini. Une fonction de complexité de $\mathbf{w}$ est une application liant chaque naturel n avec les facteurs de longueur n du mot $\mathbf{w}$.

Definition

La **complexité factorielle** du mot $\mathbf{w}$ est la fonction

$$p_{\mathbf{w}} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/ \sim_{=}).$$

Complexité factorielle

Soit **w** un mot infini. Une fonction de complexité de **w** est une application liant chaque naturel n avec les facteurs de longueur n du mot **w**.

Definition

La **complexité factorielle** du mot **w** est la fonction

$$p_{\mathbf{w}} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathsf{Fac}_{\mathbf{w}}(n)/\sim_{=}).$$

On peut remplacer $\sim_{=}$ par les autres relations d'équivalence dont on a parlé.

La **complexité factorielle** du mot $\mathbf{w}$ est la fonction

$$p_{\mathbf{w}} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_{=}).$$

La **complexité abélienne** du mot $\mathbf{w}$ est la fonction

$$\rho_{\mathbf{w}} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_{ab,1}).$$

La **complexité k-abélienne** du mot $\mathbf{w}$ est la fonction

$$\rho_{\mathbf{w}}^{(k)} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_{ab,k}).$$

La **complexité k-binomiale** du mot $\mathbf{w}$ est la fonction

$$b_{\mathbf{w}}^{(k)} : \mathbb{N} \to \mathbb{N} : n \mapsto \#(\mathrm{Fac}_{\mathbf{w}}(n)/\sim_{k}).$$

Un exemple...

Prenons le mot infini $\mathbf{t}$ généré par le morphisme $\varphi : a \mapsto ab, b \mapsto ba$. On a

$$\mathbf{t} = abbabaabbaababba \cdots$$

Les premières valeurs de la complexité factorielle sont les suivantes :

n	0	1	2	3	4	$\cdots$
$p_{\mathbf{t}}$	1	2	4	6	10	$\cdots$

Un exemple...

Prenons le mot infini $\mathbf{t}$ généré par le morphisme $\varphi : a \mapsto ab, b \mapsto ba$. On a

$$\mathbf{t} = abbabaabbaababba\cdots$$

Les premières valeurs de la complexité factorielle sont les suivantes :

n	0	1	2	3	4	$\cdots$
$p_{\mathbf{t}}$	1	2	4	6	10	$\cdots$

Les premières valeurs de la complexité abélienne sont les suivantes :

n	0	1	2	3	4	$\cdots$
$\rho_{\mathbf{t}}$	1	2	3	2	3	$\cdots$

1. Pour tous mots u, v et pout tout naturel k,

$$u \sim_{ab,k+1} v \Rightarrow u \sim_{ab,k} v \quad \text{et} \quad u \sim_{k+1} v \Rightarrow u \sim_k v.$$

Quelques propriétés

1. Pour tous mots u, v et pout tout naturel k,

$$u \sim_{ab,k+1} v \Rightarrow u \sim_{ab,k} v \quad \text{et} \quad u \sim_{k+1} v \Rightarrow u \sim_k v.$$

2. Pour tous mots u, v,

$$u \sim_1 v \Leftrightarrow u \sim_{ab,1} v.$$

Quelques propriétés

1. Pour tous mots u, v et pout tout naturel k,

$$u \sim_{ab,k+1} v \Rightarrow u \sim_{ab,k} v \quad \text{et} \quad u \sim_{k+1} v \Rightarrow u \sim_k v.$$

2. Pour tous mots u, v,

$$u \sim_1 v \Leftrightarrow u \sim_{ab,1} v.$$

3. Il y a un ordre entre les différentes fonctions de complexité :

$$\rho_{\mathbf{w}}(n) = b_{\mathbf{w}}^{(1)}(n) \leq b_{\mathbf{w}}^{(k)}(n) \leq b_{\mathbf{w}}^{(k+1)}(n) \leq p_{\mathbf{w}}(n) \quad \forall n \in \mathbb{N}, k \in \mathbb{N}.$$

A propos de la relation k-binomiale

Pour la complexité factorielle...

Théorème (Morse–Hedlund)

Soit $\mathbf{w}$ un mot infini construit sur un alphabet à ℓ lettres. Les trois affirmations suivantes sont équivalentes.

1. Le mot $\mathbf{w}$ est ultimement périodique : il existe des mots finis u, v tels que $\mathbf{w} = u \cdot v^\omega$.
2. Il existe $n \in \mathbb{N}$ tel que $p_{\mathbf{w}}(n) < n + \ell - 1$.
3. La fonction $p_{\mathbf{w}}$ est bornée par une constante.

Pour la complexité factorielle...

Théorème (Morse–Hedlund)

Soit $\mathbf{w}$ un mot infini construit sur un alphabet à ℓ lettres. Les trois affirmations suivantes sont équivalentes.

1. Le mot $\mathbf{w}$ est ultimement périodique : il existe des mots finis u, v tels que $\mathbf{w} = u \cdot v^\omega$.
2. Il existe $n \in \mathbb{N}$ tel que $p_\mathbf{w}(n) < n + \ell - 1$.
3. La fonction $p_\mathbf{w}$ est bornée par une constante.

Et pour la fonction $b^{(k)}$? L'une des implications reste évidente :

$$\mathbf{w} \text{ est ultimement périodique } \Rightarrow b_\mathbf{w}^{(k)} \text{ est borné par une constante,}$$

puisque $b_\mathbf{w}^{(k)}(n) \leq p_\mathbf{w}(n)$.

Le mot de Thue–Morse

Le **mot de Thue–Morse** est défini commme le point fixe du morphisme

$$\varphi : \{a,b\}^* \to \{a,b\}^* : \left\{ \begin{array}{lcl} a & \mapsto & ab; \\ b & \mapsto & ba, \end{array} \right.$$

On sait (M. Rigo, P. Salimov, 2015) qu'il a une complexité k-binomiale bornée.

Le mot de Thue–Morse

Le **mot de Thue–Morse** est défini commme le point fixe du morphisme

$$\varphi : \{a, b\}^* \to \{a, b\}^* : \left\{ \begin{array}{lcl} a & \mapsto & ab; \\ b & \mapsto & ba, \end{array} \right.$$

On sait (M. Rigo, P. Salimov, 2015) qu'il a une complexité k-binomiale bornée. La valeur exacte est connue :

Théorème (M. L., J. Leroy, M. Rigo, 2018)
Soit k un naturel non nul. Pour tout $n \leq 2^k - 1$, nous avons

$$b_{\mathbf{t}}^{(k)}(n) = p_{\mathbf{t}}(n),$$

tandis que pour tout $n \geq 2^k$,

$$b_{\mathbf{t}}^{(k)}(n) = \left\{ \begin{array}{ll} 3 \cdot 2^k - 3, & \text{si } n \equiv 0 \pmod{2^k}; \\ 3 \cdot 2^k - 4, & \text{sinon.} \end{array} \right.$$

Une autre famille de mots

Un **mot sturmien** est un mot infini ayant une complexité factorielle égale à $p(n) \ = \ n+1$ pour tout $n \in \mathbb{N}$.

Vu le théorème de Morse–Hedlund, il s'agit des mots apériodiques de complexité factorielle la plus faible possible.

Une autre famille de mots

Un **mot sturmien** est un mot infini ayant une complexité factorielle égale à $p(n) \ = \ n+1$ pour tout $n \in \mathbb{N}$.

Vu le théorème de Morse–Hedlund, il s'agit des mots apériodiques de complexité factorielle la plus faible possible.

Théorème (M. Rigo, P. Salimov, 2015)

Soit $\mathbf{w}$ un mot sturmien. Nous avons

$$\mathbf{b}_{\mathbf{w}}^{(k)}(n) = p_{\mathbf{w}}(n) = n+1,$$

pour tout $n \in \mathbb{N}$ et pour tout $k \geq 2$.

Une autre famille de mots

Un **mot sturmien** est un mot infini ayant une complexité factorielle égale à $p(n) \ = \ n+1$ pour tout $n \in \mathbb{N}$.

Vu le théorème de Morse–Hedlund, il s'agit des mots apériodiques de complexité factorielle la plus faible possible.

Théorème (M. Rigo, P. Salimov, 2015)
Soit $\mathbf{w}$ un mot sturmien. Nous avons

$$\mathbf{b}_{\mathbf{w}}^{(k)}(n) = p_{\mathbf{w}}(n) = n+1,$$

pour tout $n \in \mathbb{N}$ et pour tout $k \geq 2$.

Puisque $\mathbf{b}_{\mathbf{w}}^{(k)}(n) \leq \mathbf{b}_{\mathbf{w}}^{(k+1)}(n) \leq p_{\mathbf{w}}(n)$, il suffit de prouver que

$$\mathbf{b}_{\mathbf{w}}^{(2)}(n) = p_{\mathbf{w}}(n).$$

Généralisations de ces résultats : cas de Thue–Morse

Le mot de Thue–Morse fait partie d'une famille de mots plus large.

Le mot de Thue–Morse fait partie d'une famille de mots plus large.

Un morphisme est **Parikh-constant** si les images de toutes ses lettres sont égales à permutation près.
Autrement dit, pour tous $a, b, c \in A$, $|\sigma(a)|_c = |\sigma(b)|_c$.

Le mot de Thue–Morse fait partie d'une famille de mots plus large.

Un morphisme est **Parikh-constant** si les images de toutes ses lettres sont égales à permutation près.
Autrement dit, pour tous $a, b, c \in A$, $|\sigma(a)|_c = |\sigma(b)|_c$.

Si σ est un morphisme pour lequel il existe une lettre $a \in A$ telle que

- $\sigma(a)$ commence par a,
- $\lim_{n\to+\infty} |\sigma^n(a)| = +\infty$,

alors on peut définir un mot infini

$$\mathbf{w} = \lim_{n\to+\infty} \sigma^n(a),$$

que l'on appelle un **point fixe** du morphisme σ.

Points fixes de morphismes Parikh-constants

Théorème (M. Rigo, P. Salimov, 2015)

Soit $\mathbf{w}$ un mot qui est point fixe d'un morphisme Parikh-constant. Alors il existe $C_{k,\mathbf{w}} > 0$ tel que

$$b_{\mathbf{w}}^{(k)}(n) < C_{k,\mathbf{w}}$$

pour tout $n \in \mathbb{N}$.

Points fixes de morphismes Parikh-constants

Théorème (M. Rigo, P. Salimov, 2015)

Soit $\mathbf{w}$ un mot qui est point fixe d'un morphisme Parikh-constant. Alors il existe $C_{k,\mathbf{w}} > 0$ tel que

$$b_{\mathbf{w}}^{(k)}(n) < C_{k,\mathbf{w}}$$

pour tout $n \in \mathbb{N}$.

Questions ouvertes :

1. Etant donné un $\mathbf{w}$ point fixe d'un morphisme Parikh-constant σ, peut-on calculer la valeur exacte de $b_{\mathbf{w}}^{(k)}$, connaissant juste σ ?

Points fixes de morphismes Parikh-constants

Théorème (M. Rigo, P. Salimov, 2015)
Soit $\mathbf{w}$ un mot qui est point fixe d'un morphisme Parikh-constant. Alors il existe $C_{k,\mathbf{w}} > 0$ tel que

$$b_{\mathbf{w}}^{(k)}(n) < C_{k,\mathbf{w}}$$

pour tout $n \in \mathbb{N}$.

Questions ouvertes :

1. Etant donné un $\mathbf{w}$ point fixe d'un morphisme Parikh-constant σ, peut-on calculer la valeur exacte de $b_{\mathbf{w}}^{(k)}$, connaissant juste σ ?
2. Existe-t-il un $\mathbf{w}$ point fixe d'un morphisme Parikh-constant pour lequel

$$b_{\mathbf{w}}^{(k)}(n) < b_{\mathbf{t}}^{(k)}(n)$$

pour tous les $n > N$?

Les mots sturmiens sont construits sur l'alphabet binaire $\{a_1, a_2\}$.

Soit $\mathbf{w}$ un mot infini sur l'alphabet $\{a_1, \ldots, a_d\}$. C'est un **mot d'Arnoux-Rauzy** si

Les mots sturmiens sont construits sur l'alphabet binaire $\{a_1, a_2\}$.

Soit $\mathbf{w}$ un mot infini sur l'alphabet $\{a_1, \ldots, a_d\}$. C'est un **mot d'Arnoux-Rauzy** si

- $p_{\mathbf{w}}(n) = (d-1)n + 1$;

Généralisations de ces résultats : cas des mots sturmiens

Les mots sturmiens sont construits sur l'alphabet binaire $\{a_1, a_2\}$.

Soit $\mathbf{w}$ un mot infini sur l'alphabet $\{a_1, \ldots, a_d\}$. C'est un **mot d'Arnoux-Rauzy** si

- $p_{\mathbf{w}}(n) = (d-1)n + 1$;
- il est récurrent ; i.e. chacun de ses facteurs apparait une infinité de fois dans $\mathbf{w}$;

Généralisations de ces résultats : cas des mots sturmiens

Les mots sturmiens sont construits sur l'alphabet binaire $\{a_1, a_2\}$.

Soit $\mathbf{w}$ un mot infini sur l'alphabet $\{a_1, \ldots, a_d\}$. C'est un **mot d'Arnoux-Rauzy** si

- $p_{\mathbf{w}}(n) = (d-1)n + 1$;
- il est récurrent ; i.e. chacun de ses facteurs apparait une infinité de fois dans $\mathbf{w}$;
- il possède exactement un facteur spécial à gauche de chaque longueur ; i.e. pour tout n, il existe un unique facteur u de $\mathbf{w}$ de longueur n qui peut être prolongé à gauche d'au moins deux façons différentes :

$$\forall n, \exists ! u \in \mathrm{Fac}_{\mathbf{w}}(n) \text{ t.q. } \exists a_i, a_j \in A, i \neq j : a_i u, a_j u \in \mathrm{Fac}_{\mathbf{w}}(n+1);$$

Les mots sturmiens sont construits sur l'alphabet binaire $\{a_1, a_2\}$.

Soit $\mathbf{w}$ un mot infini sur l'alphabet $\{a_1, \ldots, a_d\}$. C'est un **mot d'Arnoux-Rauzy** si

- $p_{\mathbf{w}}(n) = (d-1)n + 1$;
- il est récurrent ; i.e. chacun de ses facteurs apparait une infinité de fois dans $\mathbf{w}$;
- il possède exactement un facteur spécial à gauche de chaque longueur ; i.e. pour tout n, il existe un unique facteur u de $\mathbf{w}$ de longueur n qui peut être prolongé à gauche d'au moins deux façons différentes :

$$\forall n, \exists! u \in \mathrm{Fac}_{\mathbf{w}}(n) \text{ t.q. } \exists a_i, a_j \in A, i \neq j : a_i u, a_j u \in \mathrm{Fac}_{\mathbf{w}}(n+1);$$

- il possède exactement un facteur spécial à droite de chaque longueur.

Les mots d'Arnoux-Rauzy à 2 lettres sont exactement les mots sturmiens.

Mots d'Arnoux-Rauzy

Les mots d'Arnoux-Rauzy à 2 lettres sont exactement les mots sturmiens.

Rappel :

Pour tout mot sturmien $\mathbf{w}$, on a $b_{\mathbf{w}}^{(k)} = p_{\mathbf{w}}$, pour tout $k \geq 2$.

Les mots d'Arnoux-Rauzy à 2 lettres sont exactement les mots sturmiens.

Rappel :

Pour tout mot sturmien $\mathbf{w}$, on a $b_{\mathbf{w}}^{(k)} = p_{\mathbf{w}}$, pour tout $k \geq 2$.

Conjecture :

Pour tout mot d'Arnoux-Rauzy $\mathbf{w}$, on a $b_{\mathbf{w}}^{(k)} = p_{\mathbf{w}}$, pour tout $k \geq 2$.

Les mots d'Arnoux-Rauzy à 2 lettres sont exactement les mots sturmiens.

Rappel :
Pour tout mot sturmien $\mathbf{w}$, on a $b_{\mathbf{w}}^{(k)} = p_{\mathbf{w}}$, pour tout $k \geq 2$.

Conjecture :
Pour tout mot d'Arnoux-Rauzy $\mathbf{w}$, on a $b_{\mathbf{w}}^{(k)} = p_{\mathbf{w}}$, pour tout $k \geq 2$.

La conjecture a pu être démontrée (M. L., M. Rigo, M. Rosenfeld, 2019) pour le mot de Tribonacci, point fixe du morphisme

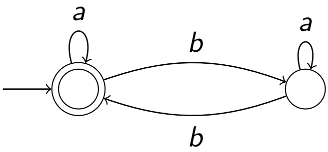
$$\tau(0) = 01, \tau(1) = 02, \tau(2) = 0.$$

Digression : les languages réguliers

Un **language** est un ensemble de mots.
Il est dit **régulier** s'il est accepté par un automate.

Un **language** est un ensemble de mots.
Il est dit **régulier** s'il est accepté par un automate.

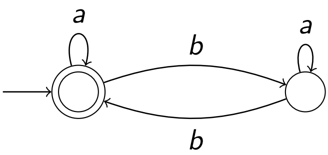
Exemple : L'automate suivant accepte l'ensemble des mots sur $\{a, b\}$ ayant un nombre pair de b.



Digression : les languages réguliers

Un **language** est un ensemble de mots.
Il est dit **régulier** s'il est accepté par un automate.

Exemple : L'automate suivant accepte l'ensemble des mots sur $\{a, b\}$ ayant un nombre pair de b.



Les languages réguliers ont en général une expression "simple". Ici, le language accepté par l'automate peut être écrit $a^*(a^*ba^*ba^*)^*$.

Les languages LL et Sing...

Soit $\sim$ une relation d'équivalence quelconque définie sur les mots. On définit

- $\mathrm{LL}(\sim, A) = \{u \in A^* \,:\, \forall v \in [u]_\sim, u \leq_{lex} v\}$, le language des plus petits représentants de chaque classe;

Les languages LL et Sing...

Soit $\sim$ une relation d'équivalence quelconque définie sur les mots. On définit

- $\mathrm{LL}(\sim, A) = \{u \in A^* \,:\, \forall v \in [u]_\sim, u \leq_{lex} v\}$, le language des plus petits représentants de chaque classe;
- $\mathrm{Sing}(\sim, A) = \{u \in A^* \,:\, \#[u]_\sim = 1\}$, le language des mots seuls dans leur classe.

Les languages LL et Sing...

Soit $\sim$ une relation d'équivalence quelconque définie sur les mots. On définit

- $\mathrm{LL}(\sim, A) = \{u \in A^* \,:\, \forall v \in [u]_\sim, u \leq_{lex} v\}$, le language des plus petits représentants de chaque classe;
- $\mathrm{Sing}(\sim, A) = \{u \in A^* \,:\, \#[u]_\sim = 1\}$, le language des mots seuls dans leur classe.

Exemple : Considérons la relation abélienne. On a $[aaa]_{\sim_{ab,1}} = \{aaa\}$ et $[baa]_{\sim_{ab,1}} = \{baa, aba, aab\}$. Donc

$$\begin{aligned} aaa \in \mathrm{Sing}(\sim_{ab,1}, \{a,b\}) \quad &\text{et} \quad baa, aba, aab \notin \mathrm{Sing}(\sim_{ab,1}, \{a,b\}) \\ aaa, aab \in \mathrm{LL}(\sim_{ab,1}, \{a,b\}) \quad &\text{et} \quad baa, aba \notin \mathrm{LL}(\sim_{ab,1}, \{a,b\}). \end{aligned}$$

...pour la relation k-abélienne

Théorème (J. Cassaigne, J. Karhumäki, S. Puzynina, M. A. Whiteland, 2017) :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_{ab,k}, A)$ et $\mathrm{Sing}(\sim_{ab,k}, A)$ sont réguliers.

...pour la relation k-abélienne

Théorème (J. Cassaigne, J. Karhumäki, S. Puzynina, M. A. Whiteland, 2017) :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_{ab,k}, A)$ et $\mathrm{Sing}(\sim_{ab,k}, A)$ sont réguliers.

En général, on "aime bien" les languages réguliers, car on peut les exprimer de façon très simple.

Théorème (J. Cassaigne, J. Karhumäki, S. Puzynina, M. A. Whiteland, 2017) :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_{ab,k}, A)$ et $\mathrm{Sing}(\sim_{ab,k}, A)$ sont réguliers.

En général, on "aime bien" les languages réguliers, car on peut les exprimer de façon très simple.

Corollaire :
Il existe une opération "assez simple", appelée k-switch, et dénotée $\equiv_k$, telle que

$$u \sim_{ab,k} v \quad \Leftrightarrow \quad u \equiv_k^* v,$$

i.e. u et v sont équivalents ssi on peut passer d'un mot à l'autre en appliquant un nombre fini de fois un k-switch.

Question naturelle :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ sont-ils réguliers ?

Question naturelle :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ sont-ils réguliers ?

1. Le cas facile : si $k = 1$. On a vu que $\sim_{1,ab}$ et $\sim_1$ étaient la même relation. Donc OUI.

...pour la relation k-binomiale

Question naturelle :
Soit A un alphabet quelconque et soit k un naturel. Les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ sont-ils réguliers ?

1. Le cas facile : si $k = 1$. On a vu que $\sim_{1,ab}$ et $\sim_1$ étaient la même relation. Donc OUI.
2. Deuxième cas facile : si $k = 2$ et $A = \{0, 1\}$. Déjà connu donc OUI (détails slides suivants).

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

01100110

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$01100110 \quad \equiv \quad 10010110$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 &\equiv 10010110 \\ &\equiv 10100101 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 &\equiv 10010110 \\ &\equiv 10100101 \\ &\equiv 01101001 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 &\equiv 10010110 \\ &\equiv 10100101 \\ &\equiv 01101001 \\ &\equiv 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 &\equiv 10010110 \\ &\equiv 10100101 \\ &\equiv 01101001 \\ &\equiv 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 \quad &\equiv \quad 10010110 \\ &\equiv \quad 10100101 \\ &\equiv \quad 01101001 \\ &\equiv \quad 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 &\equiv 10010110 \\ &\equiv 10100101 \\ &\equiv 01101001 \\ &\equiv 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\ &\equiv \quad 10100101 \\ &\equiv \quad 01101001 \\ &\equiv \quad 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\ &\equiv \quad 10100101 \\ &\equiv \quad 01101001 \\ &\equiv \quad 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\ &\equiv \quad 10100101 \\ &\equiv \quad 01101001 \\ &\equiv \quad 01011010 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{ccccc}
01100110 & \equiv & 10010110 & \equiv & 10011001 \\
 & \equiv & 10100101 & \equiv & 11000011 \\
 & \equiv & 01101001 & & \\
 & \equiv & 01011010 & &
\end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{ccccc} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & & \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{cclcl}
01100110 & \equiv & 10010110 & \equiv & 10011001 \\
 & \equiv & 10100101 & \equiv & 11000011 \\
 & \equiv & 01101001 & & \\
 & \equiv & 01011010 & &
\end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{ccccc} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & & \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned}
01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\
&\equiv \quad 10100101 \quad \equiv \quad 11000011 \\
&\equiv \quad 01101001 \\
&\equiv \quad 01011010
\end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{llllll}
01100110 & \equiv & 10010110 & \equiv & 10011001 \\
 & \equiv & 10100101 & \equiv & 11000011 \\
 & \equiv & 01101001 & & \\
 & \equiv & 01011010 & &
\end{array}$$

Si $k = 2$ et $A = \{0,1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned}
01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\
&\equiv \quad 10100101 \quad \equiv \quad 11000011 \\
&\equiv \quad 01101001 \\
&\equiv \quad 01011010
\end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{lclcl} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & & \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{cccccc} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & & \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{lclcl} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & \equiv & 00111100 \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{cccccc} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & \equiv & 00111100 \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{aligned} 01100110 \quad &\equiv \quad 10010110 \quad \equiv \quad 10011001 \\ &\equiv \quad 10100101 \quad \equiv \quad 11000011 \\ &\equiv \quad 01101001 \\ &\equiv \quad 01011010 \quad \equiv \quad 00111100 \end{aligned}$$

Si $k = 2$ et $A = \{0, 1\}$

Définissons le **switch**, dénoté $\equiv$:
Soit u un mot de la forme $x01y10z$ (resp., $x10y01z$). On dit qu'on applique un switch si on le transforme en le mot $x10y01z$ (resp., $x01y10z$).

Théorème : Nous avons

$$u \sim_2 v \quad \Leftrightarrow \quad u \equiv^* v.$$

Exemple : Générons la classe $\sim_2$ du mot 01100110.

$$\begin{array}{cccccc} 01100110 & \equiv & 10010110 & \equiv & 10011001 \\ & \equiv & 10100101 & \equiv & 11000011 \\ & \equiv & 01101001 & & \\ & \equiv & 01011010 & \equiv & 00111100 \end{array}$$

Si $k = 2$ et $A = \{0, 1\}$: $\mathrm{LL}(\sim_2, \{0, 1\})$

Corollaire : Un mot u est lexicographiquement minimal dans $[u]_{\sim_2}$ ssi une occurrence de 10 comme facteur dans u implique qu'il n'y a aucune occurrence de 01 comme facteur à droite de ce 10.

Si $k = 2$ et $A = \{0,1\}$: $\mathrm{LL}(\sim_2, \{0,1\})$

Corollaire : Un mot u est lexicographiquement minimal dans $[u]_{\sim_2}$ ssi une occurrence de 10 comme facteur dans u implique qu'il n'y a aucune occurrence de 01 comme facteur à droite de ce 10.

Retour à l'exemple : Trouvons le mot minimal dans

$$[01100110]_{\sim_2} = \{01100110, 10010110, 10011001, 10100101, \\ 11000011, 01101001, 01011010, 00111100\}.$$

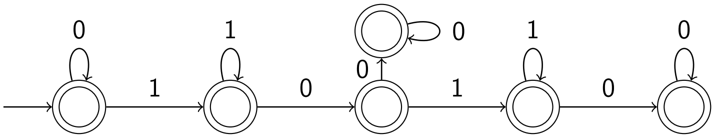
Si $k = 2$ et $A = \{0,1\}$: $\mathrm{LL}(\sim_2, \{0,1\})$

Corollaire : Un mot u est lexicographiquement minimal dans $[u]_{\sim_2}$ ssi une occurrence de 10 comme facteur dans u implique qu'il n'y a aucune occurrence de 01 comme facteur à droite de ce 10.

Retour à l'exemple : Trouvons le mot minimal dans

$$[01100110]_{\sim_2} = \{01100110, 10010110, 10011001, 10100101, 11000011, 01101001, 01011010, 00111100\}.$$

Voici un automate acceptant $\mathrm{LL}(\sim_2, \{0,1\})$.



Si $k = 2$ et $A = \{0, 1\}$: $\mathrm{Sing}(\sim_2, \{0, 1\})$

Remarque : Un mot u est seul dans $[u]_{\sim_2}$ ssi

- une occurrence de 10 comme facteur dans u implique qu'il n'y a aucune occurrence de 01 comme facteur à droite de ce 10 (i.e., il est minimal)

ET

- une occurrence de 01 comme facteur dans u implique qu'il n'y a aucune occurrence de 10 comme facteur à droite de ce 01 (i.e., il est maximal).

Si $k = 2$ et $A = \{0, 1\}$: $\text{Sing}(\sim_2, \{0, 1\})$

Remarque : Un mot u est seul dans $[u]_{\sim_2}$ ssi

- une occurrence de 10 comme facteur dans u implique qu'il n'y a aucune occurrence de 01 comme facteur à droite de ce 10 (i.e., il est minimal)

ET

- une occurrence de 01 comme facteur dans u implique qu'il n'y a aucune occurrence de 10 comme facteur à droite de ce 01 (i.e., il est maximal).

On peut donc obtenir un automate acceptant $\text{Sing}(\sim_2, \{0, 1\})$ en faisant l'intersection de l'automate du slide précédent et son "complémenté" (obtenu en échangeant les 0 et les 1).

Le cas général : $k \geq 2$ et $\#A \geq 3$

Réponse : NON, les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ ne sont pas réguliers ; ce qui diffère donc des résultats connus pour $\sim_{ab,k}$.

Le cas général : $k \geq 2$ et $\#A \geq 3$

Réponse : NON, les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ ne sont pas réguliers ; ce qui diffère donc des résultats connus pour $\sim_{ab,k}$.

Idées-clés du raisonnement:

- Ces languages sont polynomiaux. Un **language** L est dit **polynomial** si la fonction

$$n \mapsto L \cap A^n$$

est majorée par un polynôme en n.

Le cas général : $k \geq 2$ et $\#A \geq 3$

Réponse : NON, les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ ne sont pas réguliers ; ce qui diffère donc des résultats connus pour $\sim_{ab,k}$.

Idées-clés du raisonnement:

- Ces languages sont polynomiaux. Un **language** L est dit **polynomial** si la fonction
$$n \mapsto L \cap A^n$$
est majorée par un polynôme en n.
- Ces languages sont non bornés. Un **language** L est dit **borné** s'il existe des mots finis $u_1, \ldots, u_p$ tels que
$$L \subset u_1^* \cdots u_p^*.$$

Le cas général : $k \geq 2$ et $\#A \geq 3$

Réponse : NON, les languages $\mathrm{LL}(\sim_k, A)$ et $\mathrm{Sing}(\sim_k, A)$ ne sont pas réguliers ; ce qui diffère donc des résultats connus pour $\sim_{ab,k}$.

Idées-clés du raisonnement:

- Ces languages sont polynomiaux. Un **language** L est dit **polynomial** si la fonction
$$n \mapsto L \cap A^n$$
est majorée par un polynôme en n.
- Ces languages sont non bornés. Un **language** L est dit **borné** s'il existe des mots finis $u_1, \ldots, u_p$ tels que
$$L \subset u_1^* \cdots u_p^*.$$
- On conclut que les 2 languages ne sont pas réguliers car : *Tout language polynomial qui est régulier est aussi borné.*

Le cas manquant : $\#A = 2$ et $k > 2$

Conjecture :
Pour tout $k > 2$, les languages $\mathrm{LL}(\sim_k, \{0,1\})$ et $\mathrm{Sing}(\sim_k, \{0,1\})$ ne sont pas réguliers.

Ces languages sont polynomiaux, mais la technique utilisée dans le cas général pour montrer qu'ils sont non bornés ne fonctionne plus.

Le cas manquant : $\#A = 2$ et $k > 2$

Conjecture :
Pour tout $k > 2$, les languages $\mathrm{LL}(\sim_k, \{0,1\})$ et $\mathrm{Sing}(\sim_k, \{0,1\})$ ne sont pas réguliers.

Ces languages sont polynomiaux, mais la technique utilisée dans le cas général pour montrer qu'ils sont non bornés ne fonctionne plus.

Remarque :
Lorsqu'un language est régulier, on le trouve souvent "sympathique". On peut lui associer une expression "simple". Le fait que les languages $\mathrm{LL}(\sim_k, \{0,1\})$ et $\mathrm{Sing}(\sim_k, \{0,1\})$ ne soient pas réguliers dans le cas général indique qu'il y a peu de chance de trouver une opération similaire au k-switch dans le cas k-abélien.

A propos de la relation k-binomiale

Comment obtenir tous les mots d'une classe $\sim_2$?

Soit $u = u_1 \cdots u_n$ et A un alphabet d'au moins 3 lettres. On souhaite calculer $[u]_{\sim_2}$ rapidement. Comment faire ?

Comment obtenir tous les mots d'une classe $\sim_2$?

Soit $u = u_1 \cdots u_n$ et A un alphabet d'au moins 3 lettres. On souhaite calculer $[u]_{\sim_2}$ rapidement. Comment faire ?

Idée 1 : Générer toutes les permutations de $\{u_1, \ldots, u_n\}$, les mots ainsi obtenus sont $\sim_1$-équivalents à u. Il faut alors calculer les coefficients binomiaux de 2 lettres.

Comment obtenir tous les mots d'une classe $\sim_2$?

Soit $u = u_1 \cdots u_n$ et A un alphabet d'au moins 3 lettres. On souhaite calculer $[u]_{\sim_2}$ rapidement. Comment faire ?

Idée 1 : Générer toutes les permutations de $\{u_1, \ldots, u_n\}$, les mots ainsi obtenus sont $\sim_1$-équivalents à u. Il faut alors calculer les coefficients binomiaux de 2 lettres.

Idée 2 : Généraliser le switch utilisé dans le cas d'un alphabet binaire : soient a, b deux lettres de A. Le switch est défini comme suit :

$$x\,ab\,y\,ba\,z \equiv x\,ba\,y\,ab\,z.$$

Comment obtenir tous les mots d'une classe $\sim_2$?

Soit $u = u_1 \cdots u_n$ et A un alphabet d'au moins 3 lettres. On souhaite calculer $[u]_{\sim_2}$ rapidement. Comment faire ?

Idée 1 : Générer toutes les permutations de $\{u_1, \ldots, u_n\}$, les mots ainsi obtenus sont $\sim_1$-équivalents à u. Il faut alors calculer les coefficients binomiaux de 2 lettres.

Idée 2 : Généraliser le switch utilisé dans le cas d'un alphabet binaire : soient a, b deux lettres de A. Le switch est défini comme suit :

$$x\,ab\,y\,ba\,z \equiv x\,ba\,y\,ab\,z.$$

Nous avons $u \equiv^* v \Rightarrow u \sim_2 v$ mais malheureusement nous n'avons plus la réciproque : $1223312 \sim_2 2311223$ mais $1223312 \not\equiv^* 2311223$.

Si notre alphabet a d lettres, on suppose que $A = \{1, \ldots, d\}$.

Remarques : Soient $a, b \in A$. Lorsque l'on passe d'un mot $u = xaby$ à un mot $v = xbay$, ce que l'on va noter $u \xrightarrow{ab} v$, Si notre alphabet a d lettres, on suppose que $A = \{1, \ldots, d\}$.

Remarques : Soient $a, b \in A$. Lorsque l'on passe d'un mot $u = xaby$ à un mot $v = xbay$, ce que l'on va noter $u \xrightarrow{ab} v$,

1. Les mots u et v sont toujours 1-binomialement équivalents.

L'idée intelligente

Si notre alphabet a d lettres, on suppose que $A = \{1, \ldots, d\}$.

Remarques : Soient $a, b \in A$. Lorsque l'on passe d'un mot $u = xaby$ à un mot $v = xbay$, ce que l'on va noter $u \xrightarrow{ab} v$,

1. Les mots u et v sont toujours 1-binomialement équivalents.
2. Si c, d sont des lettres différentes de a, b, $\binom{v}{cd} = \binom{u}{cd}$.

L'idée intelligente

Si notre alphabet a d lettres, on suppose que $A = \{1, \dots, d\}$.

Remarques : Soient $a, b \in A$. Lorsque l'on passe d'un mot $u = xaby$ à un mot $v = xbay$, ce que l'on va noter $u \xrightarrow{ab} v$,

1. Les mots u et v sont toujours 1-binomialement équivalents.
2. Si c, d sont des lettres différentes de a, b, $\binom{v}{cd} = \binom{u}{cd}$.
3. On a $\binom{v}{ab} = \binom{u}{ab} - 1$ et $\binom{v}{ba} = \binom{u}{ba} + 1$.

L'algorithme

Soit $u = u_1 \cdots u_n$. Pour calculer $[u]_{\sim_2}$:

1. On démarre du mot "trié"

$$w = 1^{|u|_1} 2^{|u|_2} \cdots d^{|u|_d},$$

qui est le mot lexicographiquement minimal dans $[u]_{\sim_1}$.

L'algorithme

Soit $u = u_1 \cdots u_n$. Pour calculer $[u]_{\sim_2}$:

1. On démarre du mot "trié"

   $$w = 1^{|u|_1} 2^{|u|_2} \cdots d^{|u|_d},$$

   qui est le mot lexicographiquement minimal dans $[u]_{\sim_1}$.

2. Pour tous $a < b \in A$, on calcule $\binom{u}{ba}$. Remarquons que $\binom{w}{ba} = 0$.

Soit $u = u_1 \cdots u_n$. Pour calculer $[u]_{\sim_2}$:

1. On démarre du mot "trié"

$$w = 1^{|u|_1} 2^{|u|_2} \cdots d^{|u|_d},$$

   qui est le mot lexicographiquement minimal dans $[u]_{\sim_1}$.

2. Pour tous $a < b \in A$, on calcule $\binom{u}{ba}$. Remarquons que $\binom{w}{ba} = 0$.

3. On va générer tous les mots obtenus en appliquant, pour tous $a < b$, exactement $\binom{u}{ba}$ transformations du type $\xrightarrow{ab}$ au départ de w.

L'algorithme

Soit $u = u_1 \cdots u_n$. Pour calculer $[u]_{\sim_2}$:

1. On démarre du mot "trié"

$$w = 1^{|u|_1} 2^{|u|_2} \cdots d^{|u|_d},$$

   qui est le mot lexicographiquement minimal dans $[u]_{\sim_1}$.

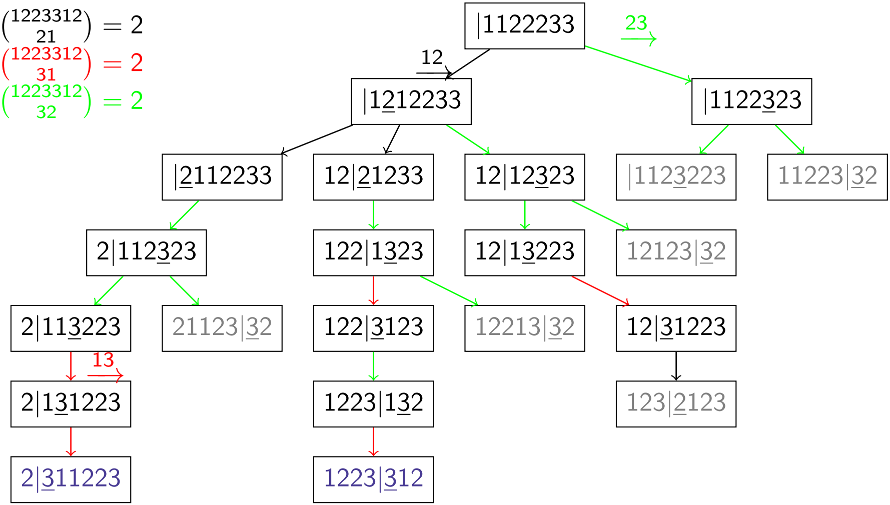
2. Pour tous $a < b \in A$, on calcule $\binom{u}{ba}$. Remarquons que $\binom{w}{ba} = 0$.

3. On va générer tous les mots obtenus en appliquant, pour tous $a < b$, exactement $\binom{u}{ba}$ transformations du type $\xrightarrow{ab}$ au départ de w.

4. On fait cela "de manière intelligente" afin de ne pas tourner en rond.

Exemple : Générer $[1223312]_{\sim_2}$

$$\binom{1223312}{21} = 2$$
$$\binom{1223312}{31} = 2$$
$$\binom{1223312}{32} = 2$$



Et pour générer une classe $\sim_k$?

On peut déjà générer la classe $\sim_2$ facilement, et il suffit donc de calculer les coefficients binomiaux de plus de 2 lettres.

Et pour générer une classe $\sim_k$?

On peut déjà générer la classe $\sim_2$ facilement, et il suffit donc de calculer les coefficients binomiaux de plus de 2 lettres.

Remarque :
Il suffit de calculer les coefficients binomiaux en les mots de Lyndon, càd les mots w tels que, pour tous mots $u, v \in A^+$ tels que $w = uv$, alors $w <_{lex} vu$. On peut déduire les autres coefficients de ceux-ci.

Et pour générer une classe $\sim_k$?

On peut déjà générer la classe $\sim_2$ facilement, et il suffit donc de calculer les coefficients binomiaux de plus de 2 lettres.

Remarque :

Il suffit de calculer les coefficients binomiaux en les mots de Lyndon, càd les mots w tels que, pour tous mots $u, v \in A^+$ tels que $w = uv$, alors $w <_{lex} vu$. On peut déduire les autres coefficients de ceux-ci.

Par exemple, le mot *bab* n'est pas Lyndon, et le coefficient peut donc s'exprimer

$$\binom{w}{bab} = \binom{w}{ab}\left[\binom{w}{b} - 1\right] - 2\binom{w}{abb},$$

en fonction de ceux des mots ab, b, abb qui sont Lyndon.

Place aux questions...