Les mots

Ivan Noyer

Lycée Thiers

Crédits

- `grep` et expressions régulières
- Wikipedia : expressions régulières
- Nombres de Catalan ici
- Toujours le Mansuy.

Alphabet

Définition

On appelle *alphabet* (en général noté Σ) tout ensemble fini. Par convention, appelle *caractères* (ou *lettres*) ses éléments.

Exemple

- En anglais, l'alphabet est $\{a, b, \ldots, z\} \cup \{A, B, \ldots, Z\}$ (noté $[a-z]||[A-Z]$ dans le langage des expressions régulières POSIX).

Alphabet

Définition

On appelle *alphabet* (en général noté Σ) tout ensemble fini. Par convention, appelle *caractères* (ou *lettres*) ses éléments.

Exemple

- En anglais, l'alphabet est $\{a, b, \ldots, z\} \cup \{A, B, \ldots, Z\}$ (noté $[a-z]||[A-Z]$ dans le langage des expressions régulières POSIX).
- L'ADN est écrit dans l'alphabet $\{A, C, G, T\}$ et l'ARN $\{A, U, G, C\}$.

Alphabet

Définition

On appelle *alphabet* (en général noté Σ) tout ensemble fini. Par convention, appelle *caractères* (ou *lettres*) ses éléments.

Exemple

- En anglais, l'alphabet est $\{a, b, \ldots, z\} \cup \{A, B, \ldots, Z\}$ (noté $[a - z]||[A - Z]$ dans le langage des expressions régulières POSIX).
- L'ADN est écrit dans l'alphabet $\{A, C, G, T\}$ et l'ARN $\{A, U, G, C\}$.
- Les protéines sont écrites dans l'alphabet des 22 acides aminées.

Alphabet

Définition

On appelle *alphabet* (en général noté Σ) tout ensemble fini. Par convention, appelle *caractères* (ou *lettres*) ses éléments.

Exemple

- En anglais, l'alphabet est $\{a, b, \ldots, z\} \cup \{A, B, \ldots, Z\}$ (noté $[a-z]||[A-Z]$ dans le langage des expressions régulières POSIX).
- L'ADN est écrit dans l'alphabet $\{A, C, G, T\}$ et l'ARN $\{A, U, G, C\}$.
- Les protéines sont écrites dans l'alphabet des 22 acides aminées.
- L'alphabet de l'arithmétique est $\{0; 1; 2; 3; 7; 5; 6; 7; 8; 9; +; -; \times, \div; (;)\}$

Mots

Définition

- Un *mot* m sur un alphabet Σ est, soit le mot vide ε, soit une suite finie de lettres de l'alphabet $(m_1, m_2, \ldots, m_n)$ (souvent noté $m_1 m_2 \ldots m_n$).

Exemple

Avec le mot $m =$ langage, $|m| = 7$ et $|m|_g = 2$.

Mots

Définition

- Un *mot* m sur un alphabet Σ est, soit le mot vide ε, soit une suite finie de lettres de l'alphabet $(m_1, m_2, \ldots, m_n)$ (souvent noté $m_1 m_2 \ldots m_n$).
- La *longueur* d'un mot m, notée $|m|$ est le nombre de lettres ou zéro si le mot est vide.

Exemple

Avec le mot $m = \text{langage}$, $|m| = 7$ et $|m|_g = 2$.

Mots

Définition

- Un *mot* m sur un alphabet Σ est, soit le mot vide ε, soit une suite finie de lettres de l'alphabet $(m_1, m_2, \ldots, m_n)$ (souvent noté $m_1 m_2 \ldots m_n$).
- La *longueur* d'un mot m, notée $|m|$ est le nombre de lettres ou zéro si le mot est vide.
- La *longueur en* $x \in \Sigma$ d'un mot m, notée $|m|_x$ est le nombre d'occurences de x dans le mot.

Exemple

Avec le mot $m =$ langage, $|m| = 7$ et $|m|_g = 2$.

Σ^+ et Σ^*

- L'ensemble des mots de longueur n est noté Σ^n.

Σ^+ et Σ^*

- L'ensemble des mots de longueur n est noté Σ^n.
- $\Sigma^0 = \{\varepsilon\}$

Σ^+ et Σ^*

- L'ensemble des mots de longueur n est noté Σ^n.
- $\Sigma^0 = \{\varepsilon\}$
- L'ensemble de tous les mots est

$$\Sigma^* = \bigcup_{n\in\mathbb{N}} \Sigma^n = \{\varepsilon\} \cup \underbrace{\bigcup_{n\in\mathbb{N}^*} \Sigma^n}_{\text{noté } \Sigma^+}$$

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \dots m_p$ et $m' = m'_1 \dots m'_q$ est le mot de longueur $p + q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \dots m_p$ et $m' = m'_1 \dots m'_q$ est le mot de longueur $p + q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.
- Pas de symétrique d'un élément, mais des règles de *régularité* :

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \dots m_p$ et $m' = m'_1 \dots m'_q$ est le mot de longueur $p+q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.
- Pas de symétrique d'un élément, mais des règles de *régularité* :
  - Si $x \cdot y = x \cdot z$ alors $y = z$ (régularité à gauche)

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \dots m_p$ et $m' = m'_1 \dots m'_q$ est le mot de longueur $p + q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.
- Pas de symétrique d'un élément, mais des règles de *régularité* :
  - Si $x \cdot y = x \cdot z$ alors $y = z$ (régularité à gauche)
  - Si $y \cdot x = z \cdot x$ alors $y = z$ (régularité à droite)

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \ldots m_p$ et $m' = m'_1 \ldots m'_q$ est le mot de longueur $p + q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.
- Pas de symétrique d'un élément, mais des règles de *régularité* :
  - Si $x \cdot y = x \cdot z$ alors $y = z$ (régularité à gauche)
  - Si $y \cdot x = z \cdot x$ alors $y = z$ (régularité à droite)
- La longueur est un morphisme de $(\Sigma^*, \cdot)$ dans $(\mathbb{N}, +)$: $(|m \cdot m'| = |m| + |m'|)$

Concaténation

Définition

La *concaténation* de deux mots $m = m_1 \dots m_p$ et $m' = m'_1 \dots m'_q$ est le mot de longueur $p + q$ dont le caractère k est m_k si $k \leq p$ et m'_{k-p} sinon. On note $m \cdot m'$ (parfois mm') la concaténation de m et m'

Remarque

- La concaténation sur Σ^* est une LCI associative d'élément neutre ε mais non commutative. $(\Sigma^*, \cdot)$ est un *monoïde* non commutatif.
- Pas de symétrique d'un élément, mais des règles de *régularité* :
  - Si $x \cdot y = x \cdot z$ alors $y = z$ (régularité à gauche)
  - Si $y \cdot x = z \cdot x$ alors $y = z$ (régularité à droite)
- La longueur est un morphisme de $(\Sigma^*, \cdot)$ dans $(\mathbb{N}, +)$: $(|m \cdot m'| = |m| + |m'|)$
- On montre facilement que la longueur en x est aussi un morphisme additif entre les mêmes monoïdes $(|m \cdot m'|_x = |m|_x + |m'|_x)$

Préfixe, suffixe

Définition

- Le mot x est appelé un *préfixe* du mot m si il existe un mot y tel que $m = x \cdot y$.

Exemple

Préfixe, suffixe

Définition

- Le mot x est appelé un *préfixe* du mot m si il existe un mot y tel que $m = x \cdot y$.
- Le mot x est appelé un *suffixe* du mot m si il existe un mot y tel que $m = y \cdot x$.

Exemple

Préfixe, suffixe

Définition

- Le mot x est appelé un *préfixe* du mot m si il existe un mot y tel que $m = x \cdot y$.
- Le mot x est appelé un *suffixe* du mot m si il existe un mot y tel que $m = y \cdot x$.

Exemple

- ε, langage et lang sont des préfixes de langage,

Préfixe, suffixe

Définition

- Le mot x est appelé un *préfixe* du mot m si il existe un mot y tel que $m = x \cdot y$.
- Le mot x est appelé un *suffixe* du mot m si il existe un mot y tel que $m = y \cdot x$.

Exemple

- ε, langage et lang sont des préfixes de langage,
- ε, langage et gage sont des suffixes de langage,

Préfixe, suffixe

Définition

- Le mot x est appelé un *préfixe* du mot m si il existe un mot y tel que $m = x \cdot y$.
- Le mot x est appelé un *suffixe* du mot m si il existe un mot y tel que $m = y \cdot x$.

Exemple

- ε, langage et lang sont des préfixes de langage,
- ε, langage et gage sont des suffixes de langage,
- Si $xu = m$ et $xv = m$ alors, par régularité, $u = v$.

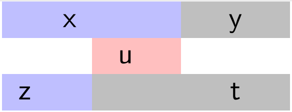
Propriété de Levi

Proposition (Levi)

Soient x, y, z, t 4 mots tels que $xy = zt$.
Alors il existe un unique mot u vérifiant :

1. $x = zu$ *et* $t = uy$ *ou*
2. $z = xu$ *et* $y = ut$

Schéma lorsque $|x| \geq |z|$:



Remarque

- Le cas 1 arrive lorsque $|x| \geq |z|$

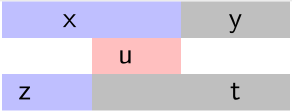
Propriété de Levi

Proposition (Levi)

Soient x, y, z, t 4 mots tels que $xy = zt$.
Alors il existe un unique mot u vérifiant :

1. $x = zu$ *et* $t = uy$ *ou*
2. $z = xu$ *et* $y = ut$

Schéma lorsque $|x| \geq |z|$:



Remarque

- Le cas 1 arrive lorsque $|x| \geq |z|$
- Le cas 2 lorsque $|x| \leq |z|$

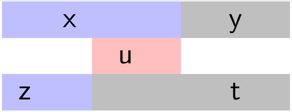
Propriété de Levi

Proposition (Levi)

Soient x, y, z, t 4 mots tels que $xy = zt$.
Alors il existe un unique mot u vérifiant :

1. $x = zu$ *et* $t = uy$ *ou*
2. $z = xu$ *et* $y = ut$

Schéma lorsque $|x| \geq |z|$:



Remarque

- Le cas 1 arrive lorsque $|x| \geq |z|$
- Le cas 2 lorsque $|x| \leq |z|$
- Si $|x| = |z|$, x et z sont deux préfixes de même longueur d'un même mot (puisque $xy = zt$). Ils sont donc égaux. Et, par régularité, $y = t$ aussi

Preuve

Unicité

Supposons que $xy = zt$ et qu'on puisse trouver u, u' vérifiant la propriété.

- Si $|x| \geq |z|$, ($x = zu'$ et $u'y = t$) et ($x = zu$ et $uy = t$), on a $x = zu = zu'$. Par régularité, $u = u'$.

Preuve

Unicité

Supposons que $xy = zt$ et qu'on puisse trouver u, u' vérifiant la propriété.

- Si $|x| \geq |z|$, $(x = zu'$ et $u'y = t)$ et $(x = zu$ et $uy = t)$, on a $x = zu = zu'$. Par régularité, $u = u'$.
- Si $|x| \leq |z|$ alors $(z = xu'$ et $u't = y)$ et $(z = xu$ et $ut = y)$: $u = u'$ aussi.

Preuve d'existence

Par récurrence sur la longueur de $|xy|$

- Pour le mot vide $xy = \varepsilon = zt$, il vient $x = y = u = z = t = \varepsilon$

Preuve d'existence

Par récurrence sur la longueur de $|xy|$

- Pour le mot vide $xy = \varepsilon = zt$, il vient $x = y = u = z = t = \varepsilon$
- On suppose que le résultat est vrai pour tous les mots de longueur $\leq n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

Preuve d'existence

Par récurrence sur la longueur de $|xy|$

- Pour le mot vide $xy = \varepsilon = zt$, il vient $x = y = u = z = t = \varepsilon$
- On suppose que le résultat est vrai pour tous les mots de longueur $\leq n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.
    - Si $x = \varepsilon$ alors $xy = y = zt$, et on pose $u = z$. Alors $xu = u = z$ et $y = ut$ (existence).

Preuve d'existence

Par récurrence sur la longueur de $|xy|$

- Pour le mot vide $xy = \varepsilon = zt$, il vient $x = y = u = z = t = \varepsilon$
- On suppose que le résultat est vrai pour tous les mots de longueur $\leq n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.
  - Si $x = \varepsilon$ alors $xy = y = zt$, et on pose $u = z$. Alors $xu = u = z$ et $y = ut$ (existence).
  - Idem si $z = \varepsilon$.

Preuve d'existence

Hérédité

On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.

Preuve d'existence

Hérédité

On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.
- On suppose maintenant que x et z possèdent au moins une lettre et s'écrivent $x = ax'$ et $z = a'z'$.

Preuve d'existence

Hérédité

On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.
- On suppose maintenant que x et z possèdent au moins une lettre et s'écrivent $x = ax'$ et $z = a'z'$.
  - Comme $xy = zt$, il vient que $ax'y = a'z't\,(1)$. La première lettre des expressions de gauche et droite est nécessairement la même donc $a = a'$.

Preuve d'existence

Hérédité

On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.
- On suppose maintenant que x et z possèdent au moins une lettre et s'écrivent $x = ax'$ et $z = a'z'$.
  - Comme $xy = zt$, il vient que $ax'y = a'z't\,(1)$. La première lettre des expressions de gauche et droite est nécessairement la même donc $a = a'$.
  - Par régularité,(1) est équivalente à $x'y = z't$. De plus $|x'y| = |z't| = n$. HR s'applique.

Preuve d'existence

Hérédité

On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.
- On suppose maintenant que x et z possèdent au moins une lettre et s'écrivent $x = ax'$ et $z = a'z'$.
  - Comme $xy = zt$, il vient que $ax'y = a'z't\,(1)$. La première lettre des expressions de gauche et droite est nécessairement la même donc $a = a'$.
  - Par régularité,(1) est équivalente à $x'y = z't$. De plus $|x'y| = |z't| = n$. HR s'applique.
  - Si $|x| \geq |z|$ alors $|x'| \geq |z'|$. Par HR, on a l'existence de u tel que $x' = z'u$ et $t = uy$. Donc $x = ax' = a'z'u = zu$ et $t = uy$.

Preuve d'existence

Hérédité

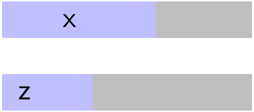
On suppose que le résultat est vrai pour tous les mots tels que $|xy| = n$ et on considère x, y, zt tels que $xy = zt$ et $|xy| = n + 1$.

- Vrai si $|x| = 0$ ou $|z| = 0$.
- On suppose maintenant que x et z possèdent au moins une lettre et s'écrivent $x = ax'$ et $z = a'z'$.
  - Comme $xy = zt$, il vient que $ax'y = a'z't\,(1)$. La première lettre des expressions de gauche et droite est nécessairement la même donc $a = a'$.
  - Par régularité,(1) est équivalente à $x'y = z't$. De plus $|x'y| = |z't| = n$. HR s'applique.
  - Si $|x| \geq |z|$ alors $|x'| \geq |z'|$. Par HR, on a l'existence de u tel que $x' = z'u$ et $t = uy$. Donc $x = ax' = a'z'u = zu$ et $t = uy$.
  - L'autre cas est laissé au lecteur.

Préfixe d'un préfixe

Corollaire

Si x et z sont deux préfixes (resp. suffixes) d'un même mot alors x est un préfixe de z ou z est un préfixe (resp. suffixe) de x.



Démonstration.

En effet si $m = xy = zt$, le théorème précédent s'applique. □

Facteur

Définition

On dit qu'un mot x est *facteur* d'un mot m s'il existe u, v, deux mots tels que $m = uxv$.

Exemple

Le mot sol est facteur de insolent.

Sous-mot

Un mot $x = x_1 \ldots x_k$ de k lettres est appelé un *sous-mot* d'un mot m s'il existe $k+1$ mots $u_0, \ldots, u_k$ vérifiant :

$$m = u_0 x_1 u_1 x_2 u_2 \ldots x_k u_k$$

Exemple

Le mot games est un sous-mot de zzghhhaaamtttezsdddd pour notre alphabet.

Mots de Dick

Cette section ne figure pas au programme officiel.

Définition

Un *$n-$chemin de Dyck* (ou encore un *chemin de Dyck de longueur* $2n$) est une suite de $2n+1$ points du plan $(M_k)_{0 \leq k \leq 2n}$ telle que

- M_0 a pour coordonnées $(0,0)$ et M_{2n} a pour coordonnées $(2n,0)$,

Mots de Dick

Cette section ne figure pas au programme officiel.

Définition

Un $n-$*chemin de Dyck* (ou encore un *chemin de Dyck de longueur* $2n$) est une suite de $2n+1$ points du plan $(M_k)_{0\leq k\leq 2n}$ telle que

- M_0 a pour coordonnées $(0,0)$ et M_{2n} a pour coordonnées $(2n,0)$,
- pour tout $k \in [\![0, 2n-1]\!]$ on a $\overrightarrow{M_kM_{k+1}} = \vec{i} + \vec{j}$ ou $\overrightarrow{M_kM_{k+1}} = \vec{i} - \vec{j}$,

Mots de Dick

Cette section ne figure pas au programme officiel.

Définition

Un *$n-$chemin de Dyck* (ou encore un *chemin de Dyck de longueur* $2n$) est une suite de $2n+1$ points du plan $(M_k)_{0\leq k\leq 2n}$ telle que

- M_0 a pour coordonnées $(0,0)$ et M_{2n} a pour coordonnées $(2n,0)$,
- pour tout $k \in [\![0,2n-1]\!]$ on a $\overrightarrow{M_kM_{k+1}} = \vec{i}+\vec{j}$ ou $\overrightarrow{M_kM_{k+1}} = \vec{i}-\vec{j}$,
- pour tout $k \in [\![0,2n-1]\!]$, $y_{M_k} \geq 0$.

Longueur 6

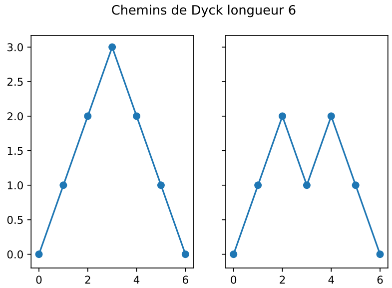


FIGURE – Longueur 6

Longueur 6

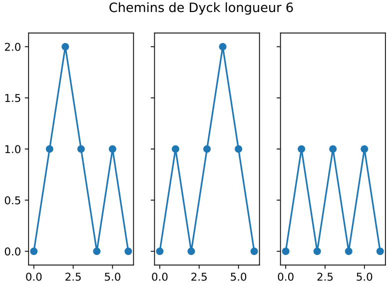


FIGURE – Longueur 6

Nombre de chemins de Dyck

Proposition

Il y a C_n chemins de Dyck de longueur $2n$ où C_n est la suite, dite de Catalan, définie par

$$C_0 = 1,\ C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$$

Remarque

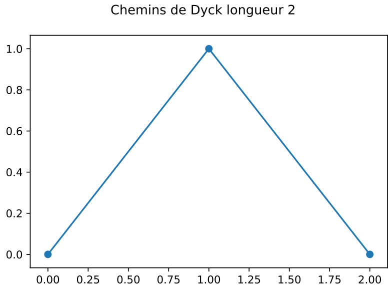
$C_1 = C_0 C_0 = 1\,;\ C_2 = C_0 C_1 + C_1 C_0 = 2.$

Preuve

On note D_n le nombre de chemins de Dyck de longueur $2n$ et on montre par récurrence que $D_n = C_n$.

- $D_1 = 1 = C_1$

FIGURE – Longueur 2

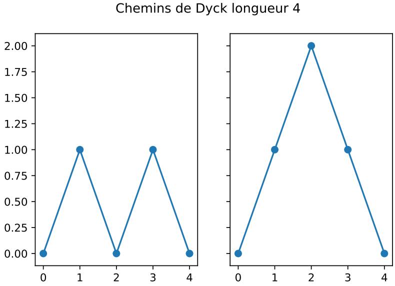


Preuve

On note D_n le nombre de chemins de Dyck de longueur $2n$ et on montre par récurrence que $D_n = C_n$.

- $D_1 = 1 = C_1$
- $D_2 = 2 = C_2$

Figure – Longueur 4



Preuve

Hérédité

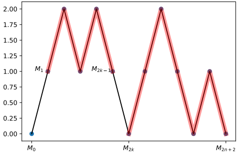
- Soit $n \in \mathbb{N}$ tel que pour tout $k \leq n$, $D_k = C_k$. Considérons $M_0, \ldots M_{2n+2}$ un chemin de Dyck de longueur $2(n+1)$.

Preuve

Hérédité

- Soit $n \in \mathbb{N}$ tel que pour tout $k \leq n$, $D_k = C_k$. Considérons $M_0, \ldots M_{2n+2}$ un chemin de Dyck de longueur $2(n+1)$.
- Soit $2k \neq 0$, le plus petit entier tel que $M_{2k} \in (Ox)$ $(2k \leq 2n+2)$. Alors $M_1 = M_{2k-1} = (1,1)$. À translation par $\vec{-i} - \vec{j}$ près, $P_k = M_1, M_2, \ldots, M_{2k-1}$ est un chemin de Dyck de longueur $2k-2$. De même, $P'_k = M_{2k}, M_{2k+1}, \ldots, M_{2n}$ est un chemin de Dyck à translation horizontale près.
  Remarque : vrai même si $2k = 2n+2$.

FIGURE – Premier point hors $(0,0)$ sur (Ox) : M_{2k}



Hérédité

- Il y a D_{k-1} façons de former P_k. Donc C_{k-1} façons par HR puisque $k-1 \leq n$.

Hérédité

- Il y a D_{k-1} façons de former P_k. Donc C_{k-1} façons par HR puisque $k-1 \leq n$.
- De plus $P'_k = M_{2k}, M_{2k+1}, \ldots, M_{2n+2}$ est un $(n+1-k)$-chemin de Dyck (car $2n+2-2k = 2(n+1-k)$).
  Il y a D_{n-k} façons de former P'_k. Donc C_{n-k} façons par HR puisque $n-k < n$ (rappelons que $k > 0$).

Hérédité

- Il y a D_{k-1} façons de former P_k. Donc C_{k-1} façons par HR puisque $k-1 \leq n$.
- De plus $P'_k = M_{2k}, M_{2k+1}, \dots, M_{2n+2}$ est un $(n+1-k)$-chemin de Dyck (car $2n+2-2k = 2(n+1-k)$).
  Il y a D_{n-k} façons de former P'_k. Donc C_{n-k} façons par HR puisque $n-k < n$ (rappelons que $k > 0$).
- Alors il y a $D_{k-1}D_{n-k}$ façons de former un chemin qui revient pour la première fois à l'axe des abscisses en M_{2k}.

Hérédité

- Il y a D_{k-1} façons de former P_k. Donc C_{k-1} façons par HR puisque $k-1 \leq n$.
- De plus $P'_k = M_{2k}, M_{2k+1}, \ldots, M_{2n+2}$ est un $(n+1-k)$-chemin de Dyck (car $2n+2-2k = 2(n+1-k)$).
  Il y a D_{n-k} façons de former P'_k. Donc C_{n-k} façons par HR puisque $n-k < n$ (rappelons que $k > 0$).
- Alors il y a $D_{k-1}D_{n-k}$ façons de former un chemin qui revient pour la première fois à l'axe des abscisses en M_{2k}.
- On valide l'hérédité car

$$D_{n+1} = \sum_{k=1}^{n+1} D_{k-1}D_{n+1-k} \underbrace{=}_{\text{HR}} \sum_{k=1}^{n+1} C_{k-1}C_{n+1-k} = \sum_{k=0}^{n} C_k C_{n-k} = \underbrace{C_{n+1}}_{\text{Hérédité C}}$$

Nombre explicite de chemins de Dyck

Construction du symétrique d'un chemin non-Dyck

- Considérons un chemin P de $M_0(0,0)$ à $M_{2n}(2n,0)$ avec n montées et n descentes. Il y en a $\binom{2n}{n}$ parmi lesquels des chemins de Dyck.

Nombre explicite de chemins de Dyck

Construction du symétrique d'un chemin non-Dyck

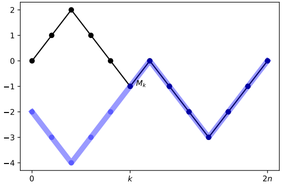
- Considérons un chemin P de $M_0(0,0)$ à $M_{2n}(2n,0)$ avec n montées et n descentes. Il y en a $\binom{2n}{n}$ parmi lesquels des chemins de Dyck.
- Si le chemin P n'est pas de Dyck (on dira *non-Dyck*), il y a un M_k $(0 < k < 2n)$ qui est le premier point d'ordonnée -1.

Nombre explicite de chemins de Dyck

Construction du symétrique d'un chemin non-Dyck

- Considérons un chemin P de $M_0(0,0)$ à $M_{2n}(2n,0)$ avec n montées et n descentes. Il y en a $\binom{2n}{n}$ parmi lesquels des chemins de Dyck.
- Si le chemin P n'est pas de Dyck (on dira *non-Dyck*), il y a un M_k $(0 < k < 2n)$ qui est le premier point d'ordonnée -1.
- On prend le symétrique des points à gauche de M_k par rapport à l'axe $\Delta\ :\ x = -1$. On note $S(P) = M'_0, \ldots, M'_{k-1}, M'_k, M'_{k+1}, \ldots, M'_{2n}$ le chemin ainsi construit ($M'_{k+q} = M_{k+q}$ et $M'_{k-q} = s_\Delta(M_{k-q})$).

FIGURE – Chemin $S(P)$ construit à partir d'un chemin non-Dyck



Nombre explicite de chemins de Dyck

- L'application S des chemins non-Dyck vers les chemins de $M'_0 = (0, -2)$ à $M'_{2n} = (2n, 0)$ à $n+1$ montées et $n-1$ descentes est bijective sur les chemins de M_0 à M_{2n} comme involution.

Nombre explicite de chemins de Dyck

- L'application S des chemins non-Dyck vers les chemins de $M'_0 = (0, -2)$ à $M'_{2n} = (2n, 0)$ à $n+1$ montées et $n-1$ descentes est bijective sur les chemins de M_0 à M_{2n} comme involution.
- Pour un chemin P non-Dyck, $S(P)$ est de longueur $2n$ et possède $n+1$ montées et $n-1$ descentes. Il y a $\binom{2n}{n-1}$ possibilité de placer les $n-1$ descentes dans $S(P)$.

Nombre explicite de chemins de Dyck

- L'application S des chemins non-Dyck vers les chemins de $M'_0 = (0, -2)$ à $M'_{2n} = (2n, 0)$ à $n+1$ montées et $n-1$ descentes est bijective sur les chemins de M_0 à M_{2n} comme involution.
- Pour un chemin P non-Dyck, $S(P)$ est de longueur $2n$ et possède $n+1$ montées et $n-1$ descentes. Il y a $\binom{2n}{n-1}$ possibilité de placer les $n-1$ descentes dans $S(P)$.
- Puisque S est bijective, il y a $\binom{2n}{n-1}$ chemins non-Dyck.

Nombre explicite de chemins de Dyck

- L'application S des chemins non-Dyck vers les chemins de $M'_0 = (0, -2)$ à $M'_{2n} = (2n, 0)$ à $n+1$ montées et $n-1$ descentes est bijective sur les chemins de M_0 à M_{2n} comme involution.
- Pour un chemin P non-Dyck, $S(P)$ est de longueur $2n$ et possède $n+1$ montées et $n-1$ descentes. Il y a $\binom{2n}{n-1}$ possibilité de placer les $n-1$ descentes dans $S(P)$.
- Puisque S est bijective, il y a $\binom{2n}{n-1}$ chemins non-Dyck.
- Alors le nombre de chemins de *Dyck* est

$$\binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n} = C_n$$

Mots de Dyck

Définition

On appelle *mot de Dyck* sur l'alphabet $\{X, Y\}$ tout mot formé de n lettres X et de n lettres Y, telle qu'aucun préfixe ne contienne plus de Y que de X.
Autrement dit, lorsque nous parcourons un mot de Dyck de gauche à droite, le nombre de X rencontrés est toujours supérieur ou égal au nombre de Y.

Exemple

Par exemple, les mots de Dyck de la longueur 6 sont :
$XXXYYY, \quad XYXXYY, \quad XYXYXY, \quad XXYYXY, \quad XXYXYY.$

Nombre de mots de Dyck

- A un mot m de Dyck de longueur $2n$ on associe de manière injective le chemin $f(m) = (M_k)_{0 \leq k \leq 2n}$ tel que $M_0(0,0)$, $M_{2n}(2n,0)$ et $\overrightarrow{M_k M_{k+1}}$ est une montée si la lettre k est un X et une descente sinon.

Nombre de mots de Dyck

- A un mot m de Dyck de longueur $2n$ on associe de manière injective le chemin $f(m) = (M_k)_{0\leq k\leq 2n}$ tel que $M_0(0,0)$, $M_{2n}(2n,0)$ et $\overrightarrow{M_kM_{k+1}}$ est une montée si la lettre k est un X et une descente sinon.
- La condition sur les préfixes assure qu'il y a toujours plus de montées que de descentes dans les sous-chemins partant de M_0 et donc que jamais les M_k ne sont d'abscisses négatives, ce qui donne que $f(m)$ est un chemin de Dyck.

Nombre de mots de Dyck

- A un mot m de Dyck de longueur $2n$ on associe de manière injective le chemin $f(m) = (M_k)_{0 \leq k \leq 2n}$ tel que $M_0(0,0)$, $M_{2n}(2n,0)$ et $\overrightarrow{M_k M_{k+1}}$ est une montée si la lettre k est un X et une descente sinon.
- La condition sur les préfixes assure qu'il y a toujours plus de montées que de descentes dans les sous-chemins partant de M_0 et donc que jamais les M_k ne sont d'abscisses négatives, ce qui donne que $f(m)$ est un chemin de Dyck.
- Il est immédiat que f est bijective.

Nombre de mots de Dyck

- A un mot m de Dyck de longueur $2n$ on associe de manière injective le chemin $f(m) = (M_k)_{0 \leq k \leq 2n}$ tel que $M_0(0,0)$, $M_{2n}(2n,0)$ et $\overrightarrow{M_k M_{k+1}}$ est une montée si la lettre k est un X et une descente sinon.
- La condition sur les préfixes assure qu'il y a toujours plus de montées que de descentes dans les sous-chemins partant de M_0 et donc que jamais les M_k ne sont d'abscisses négatives, ce qui donne que $f(m)$ est un chemin de Dyck.
- Il est immédiat que f est bijective.
- Donc le nombre de mots de Dyck est C_n.

Nombre de bons parenthésage

- Soit un *magma* muni d'une loi notée multiplicativement avec élément neutre mais non nécessairement associative.

Nombre de bons parenthésage

- Soit un *magma* muni d'une loi notée multiplicativement avec élément neutre mais non nécessairement associative.
- Le nombre de façons d'écrire un produit de n termes, correspond au nombre de façons de placer correctement n parenthèses ouvrantes et n fermantes.

Nombre de bons parenthésage

- Soit un *magma* muni d'une loi notée multiplicativement avec élément neutre mais non nécessairement associative.
- Le nombre de façons d'écrire un produit de n termes, correspond au nombre de façons de placer correctement n parenthèses ouvrantes et n fermantes.
- C'est le nombre de mots de Dyck de longueur $2n$ sur l'alphabet $\{')',' ('\}$ et il vaut C_n.

Nombre de bons parenthésage

- Soit un *magma* muni d'une loi notée multiplicativement avec élément neutre mais non nécessairement associative.
- Le nombre de façons d'écrire un produit de n termes, correspond au nombre de façons de placer correctement n parenthèses ouvrantes et n fermantes.
- C'est le nombre de mots de Dyck de longueur $2n$ sur l'alphabet $\{')',' ('\}$ et il vaut C_n.
- Il est d'usage de ne pas écrire les parenthèses externes (on écrit $a+b$ plutôt que $(a+b)$), mais cela ne change rien au nombre de bons parenthésages.

Exemples

Conformément au programme, on se contente de donner des exemples sans établir de théorie générale

- Recherche de toutes les occurences d'un mot dans un texte par exemple pour remplacer Toto par Gogo. Cela correspond à la recherche d'un facteur (Toto) dans un mot (le texte).

Exemples

Conformément au programme, on se contente de donner des exemples sans établir de théorie générale

- Recherche de toutes les occurences d'un mot dans un texte par exemple pour remplacer Toto par Gogo. Cela correspond à la recherche d'un facteur (Toto) dans un mot (le texte).
- On peut traquer les fautes d'orthographes qui correspondent à la présence de plus de deux s consécutifs dans un texte : un tel mot n'existe pas en français. On cherche donc tous les facteurs de longueur au moins 3 qui ne sont constitués que de s.

Code barre

- Un code barre monodimensionnel est une succession de barres verticales et d'espaces d'épaisseurs variables.

FIGURE – code barre

0 36000 29145 2

Code barre

- Un code barre monodimensionnel est une succession de barres verticales et d'espaces d'épaisseurs variables.

FIGURE – code barre

- La signification de ces séquences de barres peut varier selon l'usage.

Code barre

- Un code barre monodimensionnel est une succession de barres verticales et d'espaces d'épaisseurs variables.

FIGURE – code barre

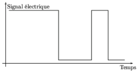
- La signification de ces séquences de barres peut varier selon l'usage.
- Par exemple, on peut imaginer que « une bande noire ayant trois fois la taille d'une bande élémentaire, suivie d'une bande blanche du double de la largeur élémentaire, suivie d'une bande noire et d'une bande blanche toutes deux de largeur élémentaire » désignent le chiffre 0.

FIGURE – Zéro en code barre

Code barre

- ≪ Lu ≫ par un lecteur optique, la séquence précédente est transformée en signal électrique en fonction du temps, ressemblant à ceci :

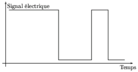
FIGURE – Signal électrique en fonction du temps.



Code barre

- ≪ Lu ≫ par un lecteur optique, la séquence précédente est transformée en signal électrique en fonction du temps, ressemblant à ceci :

FIGURE – Signal électrique en fonction du temps.

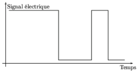


- C'est traduit ensuite en une séquence de bits. Par exemple 0001101 (avec 1 pour blanc, 0 pour noir).

Code barre

- « Lu » par un lecteur optique, la séquence précédente est transformée en signal électrique en fonction du temps, ressemblant à ceci :

FIGURE – Signal électrique en fonction du temps.



- C'est traduit ensuite en une séquence de bits. Par exemple 0001101 (avec 1 pour blanc, 0 pour noir).
- Si on veut connaître, par exemple, le nombre de zéros contenus dans un code barre, on cherche donc les occurences du motif 0001101 dans la succession de bits renvoyée par le lecteur optique.

Code barre

- « Lu » par un lecteur optique, la séquence précédente est transformée en signal électrique en fonction du temps, ressemblant à ceci :

FIGURE – Signal électrique en fonction du temps.

- C'est traduit ensuite en une séquence de bits. Par exemple 0001101 (avec 1 pour blanc, 0 pour noir).
- Si on veut connaître, par exemple, le nombre de zéros contenus dans un code barre, on cherche donc les occurences du motif 0001101 dans la succession de bits renvoyée par le lecteur optique.

La distance de la tête du lecteur optique à la surface du code barre peut varier. C'est pour cela que, dans un code barre, il y a une partie qui ne sert pas au stockage de l'information mais à l'étalonnage du lecteur afin qu'il reconnaisse la largeur standard.

Ribosome

- Les ribosomes sont des molécules composées de protéines et d'ARN (conservés au cours de l'évolution) présents dans les cellules eucaryotes et procaryotes. Leur fonction est de synthétiser les protéines en décodant l'information contenue dans l'ARN messager (ARNm).

Ribosome

- Les ribosomes sont des molécules composées de protéines et d'ARN (conservés au cours de l'évolution) présents dans les cellules eucaryotes et procaryotes. Leur fonction est de synthétiser les protéines en décodant l'information contenue dans l'ARN messager (ARNm).
- Le ribosome ≪ lit ≫ l'ARNm et transforme chaque *codon* de celui-ci (succession de trois nucléotides) en acide aminé. Les assemblages de ces acides aminés forment alors la protéine fabriquée par le ribosome.

Ribosome

- Les ribosomes sont des molécules composées de protéines et d'ARN (conservés au cours de l'évolution) présents dans les cellules eucaryotes et procaryotes. Leur fonction est de synthétiser les protéines en décodant l'information contenue dans l'ARN messager (ARNm).
- Le ribosome ≪ lit ≫ l'ARNm et transforme chaque *codon* de celui-ci (succession de trois nucléotides) en acide aminé. Les assemblages de ces acides aminés forment alors la protéine fabriquée par le ribosome.
- Toutefois, il faut bien s'arrêter un jour d'assembler ces acides (les protéines sont finies).

Ribosome

- Les ribosomes sont des molécules composées de protéines et d'ARN (conservés au cours de l'évolution) présents dans les cellules eucaryotes et procaryotes. Leur fonction est de synthétiser les protéines en décodant l'information contenue dans l'ARN messager (ARNm).
- Le ribosome ≪ lit ≫ l'ARNm et transforme chaque *codon* de celui-ci (succession de trois nucléotides) en acide aminé. Les assemblages de ces acides aminés forment alors la protéine fabriquée par le ribosome.
- Toutefois, il faut bien s'arrêter un jour d'assembler ces acides (les protéines sont finies).
- Le Ribosome sait qu'il est arrivé à la fin de la protéine lorsqu'il retrouve l'un des trois codons (motifs) suivant dans l'ARNm : UAA, UAG, UGA.

Motifs de dimensions plus grande que 1

- Lors d'une avalanche, on cherche sur des photographies thermiques les motifs ressemblant à des corps ensevelis.

Motifs de dimensions plus grande que 1

- Lors d'une avalanche, on cherche sur des photographies thermiques les motifs ressemblant à des corps ensevelis.
- Reconnaissance de forme après le scan d'un document pour retrouver les mots du texte d'après une image.

QR code

- Le code appelé ≪ Quick Response ≫ en raison de sa facilité et sa rapidité à être décodé est un type de code barre en deux dimensions, présenté sous forme de pictogramme constitué de pixels noirs sur carré de fond blanc.

QR code

- Le code appelé ≪ Quick Response ≫ en raison de sa facilité et sa rapidité à être décodé est un type de code barre en deux dimensions, présenté sous forme de pictogramme constitué de pixels noirs sur carré de fond blanc.
- La lecture de ce code délivre une information déchiffrée par des lecteurs spécifiques (smartphones, téléphones mobiles, webcam, tablette mobile).

QR code

- Le code appelé ≪ Quick Response ≫ en raison de sa facilité et sa rapidité à être décodé est un type de code barre en deux dimensions, présenté sous forme de pictogramme constitué de pixels noirs sur carré de fond blanc.
- La lecture de ce code délivre une information déchiffrée par des lecteurs spécifiques (smartphones, téléphones mobiles, webcam, tablette mobile).
- Le QR est photographié via la caméra embarquée de l'appareil qui le déchiffre grâce à une application préinstallée ou à installer.

QR code

- Le code appelé ≪ Quick Response ≫ en raison de sa facilité et sa rapidité à être décodé est un type de code barre en deux dimensions, présenté sous forme de pictogramme constitué de pixels noirs sur carré de fond blanc.
- La lecture de ce code délivre une information déchiffrée par des lecteurs spécifiques (smartphones, téléphones mobiles, webcam, tablette mobile).
- Le QR est photographié via la caméra embarquée de l'appareil qui le déchiffre grâce à une application préinstallée ou à installer.
- Les informations sont stockées verticalement et horizontalement. Gain de place important par rapport au flash-code.

QR code

- Le code appelé « Quick Response » en raison de sa facilité et sa rapidité à être décodé est un type de code barre en deux dimensions, présenté sous forme de pictogramme constitué de pixels noirs sur carré de fond blanc.
- La lecture de ce code délivre une information déchiffrée par des lecteurs spécifiques (smartphones, téléphones mobiles, webcam, tablette mobile).
- Le QR est photographié via la caméra embarquée de l'appareil qui le déchiffre grâce à une application préinstallée ou à installer.
- Les informations sont stockées verticalement et horizontalement. Gain de place important par rapport au flash-code.
- 30% environ de la surface est consacrée à la détection et à la correction d'erreur (redondances)

QR code

FIGURE – Un QR code. Les gros carrés noirs servent au positionnement.

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide
  - toutes les lettres de l'alphabet

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide
  - toutes les lettres de l'alphabet
- Etant données deux expressions régulières R et S, les opérations suivantes produisent des expressions régulières :

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide
  - toutes les lettres de l'alphabet
- Etant données deux expressions régulières R et S, les opérations suivantes produisent des expressions régulières :
  - La concaténation de R et S notée $(R \cdot S)$ ou (RS), ou encore RS

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide
  - toutes les lettres de l'alphabet
- Etant données deux expressions régulières R et S, les opérations suivantes produisent des expressions régulières :
  - La concaténation de R et S notée $(R \cdot S)$ ou (RS), ou encore RS
  - La réunion (aussi appelée choix) notée $(R|S)$.

Présentation

Les expressions régulières constituent un système très puissant et très rapide pour faire des recherches dans des chaînes de caractères (des phrases, par exemple).

Définition

Les expressions régulières sur un alphabet Σ sont :

- Les constantes :
  - $\emptyset$: l'ensemble vide.
  - ε : la chaîne de caractère vide
  - toutes les lettres de l'alphabet
- Etant données deux expressions régulières R et S, les opérations suivantes produisent des expressions régulières :
  - La concaténation de R et S notée $(R \cdot S)$ ou (RS), ou encore RS
  - La réunion (aussi appelée choix) notée $(R|S)$.
  - L'itération de de R notée $(R^\star)$ (et qui signifie $\varepsilon, R, RR, \ldots, RRRRRRRRRRR, \ldots$. On l'appelle aussi *étoile (de Kleene)*.

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| \, b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| \, b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| \, b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

- 1. Etoile (plus forte priorité)

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| \, b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

- 1. Etoile (plus forte priorité)
  2. Concaténation

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| \, b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

- 1. Etoile (plus forte priorité)
  2. Concaténation
  3. Choix (plus faible priorité)

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbbba, \dots$
- $(((ab)^\star))|\, b)$ désigne $b, ab, abab, ababab, \dots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

- 1. Etoile (plus forte priorité)
  2. Concaténation
  3. Choix (plus faible priorité)
- $a \cdot b^\star |\, b$ désigne

$$((a \cdot (b^\star))|\, b)$$

Exemples

Sur $\{a, b\}$

- $((a \cdot (b^\star)) \cdot a)$ désigne $aa, aba, abbbbbbbbbbbba, \ldots$
- $(((ab)^\star))| b)$ désigne $b, ab, abab, ababab, \ldots$

Remarque

Des règles de priorité permettent de limiter l'usage des parenthèses

- 1. Etoile (plus forte priorité)
  2. Concaténation
  3. Choix (plus faible priorité)
- $a \cdot b^\star | b$ désigne
$$((a \cdot (b^\star))| b)$$
- Plus la priorité est forte, plus, si on remet des parenthèses, le nombre de parenthèses entourant l'opérateur est élevé.

Exemples

Exercice

Sur $\{a, b, c\}$

1. Décrire l'ensemble des mots qui commencent par a, se poursuivent par un nombre arbitraire de ba et peuvent éventuellement se terminer par un c
2. Décrire les mots qui ne sont formés que de a en nombre impair

Réponses

1. $a(ba)^\star(c|\varepsilon)$
2. $(aa)^\star a$

Description en OCAML

Un type

```
type regexp =
  | Emptyset
  | Const of string (*contient déjà la chaîne vide ""*)
  | Concat of regexp * regexp
  | Choice of regexp * regexp
  | Star of regexp;;

(*  e=a/bb*a   *)
let e = Choice(Const "a",
      Concat(Concat(Const "b", Star (Const "b")),Const "a"))
         ;;
```

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.
- la troisième Conversation de Vienne de 1994 en recommanda officiellement l'adoption.

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.
- la troisième Conversation de Vienne de 1994 en recommanda officiellement l'adoption.
- Le 1er juillet 1996, l'Allemagne, la Suisse, l'Autriche et le Liechtenstein s'engagèrent à adopter la réforme.

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.
- la troisième Conversation de Vienne de 1994 en recommanda officiellement l'adoption.
- Le 1er juillet 1996, l'Allemagne, la Suisse, l'Autriche et le Liechtenstein s'engagèrent à adopter la réforme.
- Parmi les nouvelles règles, le « eszett » ne s'emploie plus qu'après une voyelle longue ou une diphtongue :

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.
- la troisième Conversation de Vienne de 1994 en recommanda officiellement l'adoption.
- Le 1er juillet 1996, l'Allemagne, la Suisse, l'Autriche et le Liechtenstein s'engagèrent à adopter la réforme.
- Parmi les nouvelles règles, le « eszett » ne s'emploie plus qu'après une voyelle longue ou une diphtongue :
  - on l'utilise dans *Grüße, bloß, beißen*

Expressions régulières étendues

- Le *Cercle de travail international pour l'orthographe (de la langue allemande)* fut fondé en 1980 et publia ses propositions de réformes en 1992.
- la troisième Conversation de Vienne de 1994 en recommanda officiellement l'adoption.
- Le 1er juillet 1996, l'Allemagne, la Suisse, l'Autriche et le Liechtenstein s'engagèrent à adopter la réforme.
- Parmi les nouvelles règles, le « eszett » ne s'emploie plus qu'après une voyelle longue ou une diphtongue :
  - on l'utilise dans *Grüße, bloß, beißen*
  - mais plus dans *bisschen (bißchen), Fass (Faß), du musst (du mußt)*

Expressions régulières étendues

- A partir de 1996, les éditeurs durent traquer les fautes apparues dans leurs ouvrages du fait des règles nouvelles.

Expressions régulières étendues

- A partir de 1996, les éditeurs durent traquer les fautes apparues dans leurs ouvrages du fait des règles nouvelles.
- Par exemple, il faut remplacer *i*ß par *iss* mais laisser *ei*ß tel quel.

Expressions régulières étendues

- A partir de 1996, les éditeurs durent traquer les fautes apparues dans leurs ouvrages du fait des règles nouvelles.
- Par exemple, il faut remplacer *i*ß par *iss* mais laisser *ei*ß tel quel.
- Or cette sélection (la différence ensembliste des mots contenant ß moins les mots pour lesquels ß est autorisé) n'est pas modélisable (facilement) avec les expressions régulières.
  En fait, si elle l'est, mais au prix d'une complexité très importante.

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :
  1. la différence ensembliste,

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :
  1. la différence ensembliste,
  2. l'intersection (pour gérer la superposition de deux contraintes).

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :
  1. la différence ensembliste,
  2. l'intersection (pour gérer la superposition de deux contraintes).
- Muni de ces deux nouvelles règles, l'ensemble des expressions régulières devient celui des *expressions régulières étendues*.

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :
  1. la différence ensembliste,
  2. l'intersection (pour gérer la superposition de deux contraintes).
- Muni de ces deux nouvelles règles, l'ensemble des expressions régulières devient celui des *expressions régulières étendues*.
- En fait seule l'opération de complémentation notée R^c (pour désigner les expressions qui décrivent les mots de Σ^* ne vérifiant pas R) est vraiment nécessaire car en théorie des ensembles $A \cap B = \overline{(\overline{A} \cup \overline{B})}$ et $A \setminus B = A \cap \overline{B}$.

Expressions régulières étendues

- Il est donc tentant d'ajouter aux constructeurs d'expressions régulières :
  1. la différence ensembliste,
  2. l'intersection (pour gérer la superposition de deux contraintes).
- Muni de ces deux nouvelles règles, l'ensemble des expressions régulières devient celui des *expressions régulières étendues*.
- En fait seule l'opération de complémentation notée R^c (pour désigner les expressions qui décrivent les mots de Σ^* ne vérifiant pas R) est vraiment nécessaire car en théorie des ensembles $A \cap B = \overline{(\overline{A} \cup \overline{B})}$ et $A \setminus B = A \cap \overline{B}$.
- On peut montrer que l'opération de complémentation est redondante car elle peut toujours être exprimée avec les autres opérateurs mais avec une complexité très importante (en double exponentielle, c'est à dire exponentielle d'exponentielle).

Expressions régulières linéaires

Définition

On dit qu'une expression régulière est *linéaire* lorsque chaque caractère de Σ n'y apparaît qu'une fois au plus.

Exemple

- $ab^\star|c$ mais pas

Expressions régulières linéaires

Définition

On dit qu'une expression régulière est *linéaire* lorsque chaque caractère de Σ n'y apparaît qu'une fois au plus.

Exemple

- $ab^\star|c$ mais pas
- $(ab)^\star|(ba)^\star$

POSIX

Les expression régulières POSIX utilisées sous UNIX avec (par exemple) `egrep` sont bien plus expressives que celles que nous venons de décrire :

POSIX

Regexp	Signification
`A`	Lettre A
`[a-z]`	Lettres minuscules de a à z
`[A-Ea-e]`	Lettres de a à e, majuscules et minuscules
`[0-9]`	Chiffres
`[^aeiou]`	Tout caractère sauf une voyelle minuscule
`^`	Début de ligne
`$`	Fin de ligne
`ab*`	a suivi de zéro ou plusieurs b
`ab+`	a suivi de au moins un b
`b{2,4}`	entre 2 et 4 b
`b?`	zéro ou un b
`a|b`	a ou b