Chapitre 4 : Les langages

Option Informatique – MP

Lycée Chateaubriand

Table des matières

Plan

Dans ce chapitre nous allons étudier la recherche de motifs. Cela a beaucoup d'applications dans de nombreux domaines. Le principe est de rechercher un motif dans des données de dimension 1 (une suite de caractères) ou de dimension 2 (une image) . On peut penser aux exemples suivants :

Généralités

Dans ce chapitre nous allons étudier la recherche de motifs. Cela a beaucoup d'applications dans de nombreux domaines. Le principe est de rechercher un motif dans des données de dimension 1 (une suite de caractères) ou de dimension 2 (une image) . On peut penser aux exemples suivants :

- Un visage dans une photo

Généralités

Dans ce chapitre nous allons étudier la recherche de motifs. Cela a beaucoup d'applications dans de nombreux domaines. Le principe est de rechercher un motif dans des données de dimension 1 (une suite de caractères) ou de dimension 2 (une image) . On peut penser aux exemples suivants :

- Un visage dans une photo
- Un QR code

Généralités

Dans ce chapitre nous allons étudier la recherche de motifs. Cela a beaucoup d'applications dans de nombreux domaines. Le principe est de rechercher un motif dans des données de dimension 1 (une suite de caractères) ou de dimension 2 (une image) . On peut penser aux exemples suivants :

- Un visage dans une photo
- Un QR code
- Une suite de lettres (ou un motif) dans une chaine de de caractères

Généralités

Dans ce chapitre nous allons étudier la recherche de motifs. Cela a beaucoup d'applications dans de nombreux domaines. Le principe est de rechercher un motif dans des données de dimension 1 (une suite de caractères) ou de dimension 2 (une image) . On peut penser aux exemples suivants :

- Un visage dans une photo
- Un QR code
- Une suite de lettres (ou un motif) dans une chaine de de caractères

Dans la suite nous travaillerons essentiellement sur la recherche d'un motif (notion précisée plus loin) dans une chaine de caractères.

Dans la suite nous travaillerons essentiellement sur la recherche d'un motif (notion précisée plus loin) dans une chaine de caractères.

C'est en particulier cela qui sert à un compilateur pour identifier si un programme ne comporte pas d'erreurs de syntaxes.

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

- On parcourt le texte.

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

- On parcourt le texte.
- Pour chaque position possible du motif dans le texte on teste si la position est une occurence du motif.

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

- On parcourt le texte.
- Pour chaque position possible du motif dans le texte on teste si la position est une occurence du motif.
- Si, pour une position donnée on trouve le motif (en testant de gauche à droite) on retourne la position.

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

- On parcourt le texte.
- Pour chaque position possible du motif dans le texte on teste si la position est une occurence du motif.
- Si, pour une position donnée on trouve le motif (en testant de gauche à droite) on retourne la position.
- Dans le cas contraire on teste la position suivante.

Algorithme naïf

L'algorithme principal est le suivant (en prenant pour motif un mot) :

- On parcourt le texte.
- Pour chaque position possible du motif dans le texte on teste si la position est une occurence du motif.
- Si, pour une position donnée on trouve le motif (en testant de gauche à droite) on retourne la position.
- Dans le cas contraire on teste la position suivante.
- Si on est arrivé au bout des positions possibles le motif n'est pas détecté.

Implémentation en Caml

- En Caml il existe un type `char` pour les caractères. Un caractère est défini entre apostrophes.

Implémentation en Caml

- En Caml il existe un type `char` pour les caractères. Un caractère est défini entre apostrophes.
- Il existe aussi un type `string` pour les chaines de caractères. Une chaine de caractères est définie entre guillemets.

Implémentation en Caml

- En Caml il existe un type `char` pour les caractères. Un caractère est défini entre apostrophes.
- Il existe aussi un type `string` pour les chaines de caractères. Une chaine de caractères est définie entre guillemets.
- On accède à la i-ème lettre d'une chaine `w` par la commande `w.[i]`

Implémentation en Caml

- En Caml il existe un type `char` pour les caractères. Un caractère est défini entre apostrophes.
- Il existe aussi un type `string` pour les chaines de caractères. Une chaine de caractères est définie entre guillemets.
- On accède à la i-ème lettre d'une chaine `w` par la commande `w.[i]`
- Les chaines de caractères ne sont pas mutables.

Exercice

Ecrire une fonction `trouve : string -> string -> bool` telle que `trouve texte mot` renvoie un booléen selon selon que le mot `mot` se trouve ou non dans le texte `texte`.

Exercice

Ecrire une fonction `trouve : string -> string -> bool` telle que `trouve texte mot` renvoie un booléen selon selon que le mot `mot` se trouve ou non dans le texte `texte`.

```
let trouve texte mot =
  let trouve = ref false in
    let n = String.length texte and p = String.length mot in
      let i = ref 0 in
        while !i <= n - p && not (!trouve) do
          let j = ref 0 in
            let possible = ref true in
              while !possible && !j < p do
                if texte.[!i + !j] = mot.[!j] then j := !j + 1 else possible := false
              done;
              trouve := !possible;
              i := !i + 1
        done;
        !trouve;;
```

Exercice

Quelle est la complexité de la fonction `trouve` ?

Exercice

Quelle est la complexité de la fonction `trouve` ?

Dans le pire des cas la boucle while sur la variable i va passer toutes les valeurs entre 0 et $n - p$ et dans cette boucle, la boucle sur la variable j va passer toutes les valeurs entre 0 et p. On en déduit une complexité en $O(np)$.

Exercice

Modifier la fonction précédente pour écrire une fonction `occurrences : string -> string -> int` qui renvoie le nombre d'occurrences du mot `mot` dans le texte `texte`.

Exercice

Modifier la fonction précédente pour écrire une fonction `occurrences : string -> string -> int` qui renvoie le nombre d'occurrences du mot `mot` dans le texte `texte`.

```
let occurrences texte mot =
  let n = String.length texte and p = String.length mot in
    let rec aux debut acc =
      if debut + p - 1 = n then acc else
        (
          let possible = ref true and j = ref 0 in
            while !possible && !j < p do
              if texte.[debut + !j] = mot.[!j] then j := !j + 1 else possible := false
            done;
            if !possible then aux (debut + 1) (acc + 1) else aux (debut + 1) acc
        )
    in
      aux 0 0;;
```

Définition 1 (Alphabet et mots)

Soit $\mathscr{A}$ un ensemble fini (que l'on appelle alphabet).

- *un mot sur $\mathscr{A}$ est*
  - *soit le mot vide noté ε ;*
  - *soit une suite finie $w_0 w_1 \dots w_{n-1}$ ou pour tout $k \in [\![0, n-1]\!]$, $w_k \in \mathscr{A}$.*
- *On note $\mathscr{A}^\star$ l'ensemble des mots sur $\mathscr{A}$.*

Terminologie

Soit w un mot sur $\mathscr{A}$

Terminologie

Soit w un mot sur $\mathscr{A}$

- On note $|w|$ la longueur du mot qui vaut 0 si le mot est vide et n si le mot est composé de n lettres.

Terminologie

Soit w un mot sur $\mathscr{A}$

- On note $|w|$ la longueur du mot qui vaut 0 si le mot est vide et n si le mot est composé de n lettres.
- Soit $k \in [\![0, |w| - 1]\!]$ on notera w_i ou $w[i]$ sa i-ème lettre.

Terminologie

Soit w un mot sur $\mathscr{A}$

- On note $|w|$ la longueur du mot qui vaut 0 si le mot est vide et n si le mot est composé de n lettres.
- Soit $k \in [\![0, |w| - 1]\!]$ on notera w_i ou $w[i]$ sa i-ème lettre.
- Soit $a \in \mathscr{A}$ on note $|w|_a$ et on appelle longueur en a du mot w le nombre d'occurrences de a dans le mot w.

Exemple

Pour l'alphabet $\mathscr{A} = \{a, b, c\}$.

Le mot $w = ababababac$ est de longueur 8, on a $|w|_a = 4$.

Concaténation

Définition 2 (Concaténation)

Soit w_1 et w_2 deux mots sur $\mathscr{A}$, la concaténation de w_1 et w_2 notée $w_1.w_2$ ou $w_1 w_2$ est le mot obtenu en mettant bout à bout les mots w_1 et w_2. On a donc pour $k \in [\![0, |w_1| + |w_2| - 1]\!]$:

$$(w_1 w_2)[k] = \begin{cases} w_1[k] & \text{si } k < |w_1| \\ w_2[k - |w_1|] & \text{si } k \geq |w_1| \end{cases}$$

Concaténation

Théorème 1

- L'opération . est une opération interne sur $\mathscr{A}^\star$.

Concaténation

Théorème 1

- L'opération . est une opération interne sur $\mathscr{A}^\star$.
- L'opération . est associative.

Concaténation

Théorème 1

- L'opération . est une opération interne sur $\mathscr{A}^\star$.
- L'opération . est associative.
- L'opération . admet le mot vide ε comme élément neutre.

Concaténation

Théorème 1

- L'opération . est une opération interne sur $\mathscr{A}^\star$.
- L'opération . est associative.
- L'opération . admet le mot vide ε comme élément neutre.

On dit que $(\mathscr{A}^\star, .)$ est un monoïde (ou magma associatif unifère).

Concaténation

$(\mathscr{A}^\star, .)$ est-il un groupe ?

Concaténation

$(\mathscr{A}^\star, .)$ est-il un groupe ?

Non (sauf si $\mathscr{A} = \emptyset$) car pour une lettre a dans $\mathscr{A}$, le mot a n'a pas d'inverse.

Concaténation

Théorème 2

Les fonctions longueur $|.|$ et longueur en un caractère a : $|.|_a$ sont additives :

$$\forall (w_1, w_2) \in (\mathscr{A}^\star)^2, |w_1 w_2| = |w_1| + |w_2| \text{ et } |w_1 w_2|_a = |w_1|_a + |w_2|_a.$$

Concaténation

Théorème 2

Les fonctions longueur $|.|$ et longueur en un caractère a : $|.|_a$ sont additives :

$$\forall (w_1, w_2) \in (\mathscr{A}^\star)^2, |w_1 w_2| = |w_1| + |w_2| \text{ et } |w_1 w_2|_a = |w_1|_a + |w_2|_a.$$

On dit que ce sont des morphismes de monoïdes de $(\mathscr{A}^\star, .)$ dans $(\mathbb{N}, +)$

Concaténation

Définition 3 (Puissances)

Soit w un mot, on définit ses puissances par

$$w^0 = \varepsilon \ ; \ w^1 = w, \ \textit{et} \ \forall n \in \mathbb{N}, w^{n+1} = w.w^n.$$

Facteurs, préfixes et suffixes

Définition 4 (Facteurs, préfixes et suffixes)

Soit w un mot sur $\mathscr{A}^\star$.

- *S'il existe des α, β et u dans $\mathscr{A}^\star$ tels que $w = \alpha u \beta$, on dit que u est un facteur de w.*

Facteurs, préfixes et suffixes

Définition 4 (Facteurs, préfixes et suffixes)

Soit w un mot sur $\mathscr{A}^\star$.

- *S'il existe des α, β et u dans $\mathscr{A}^\star$ tels que $w = \alpha u \beta$, on dit que u est un facteur de w.*
- *Si de plus, $\alpha = \varepsilon$ on dit que u est un préfixe de w.*

Facteurs, préfixes et suffixes

Définition 4 (Facteurs, préfixes et suffixes)

Soit w un mot sur $\mathscr{A}^\star$.

- *S'il existe des α, β et u dans $\mathscr{A}^\star$ tels que $w = \alpha u \beta$, on dit que u est un facteur de w.*
- *Si de plus, $\alpha = \varepsilon$ on dit que u est un préfixe de w.*
- *A l'inverse si $\beta = \varepsilon$ on dit que u est un suffixe de w.*

Exemple

Si $w = ababababac$ alors :

- ab est un préfixe de w,
- ba est un facteur de w
- c un suffixe de w

Théorème 3

Le monoïde $\mathscr{A}^*$ est régulier à droite et à gauche. Cela signifie que si u, v et w appartiennent à $\mathscr{A}^*$ alors

$$uv = uw \Rightarrow v = w \text{ et } vu = wu \Rightarrow v = w$$

Exercice : Lemme de Levi

Soit A un alphabet et soient $u, v, x, y \in (A^*)^4$.

1. Montrer que $uv = xy$ si et seulement si il existe $t \in A^*$ tel que $\Big((ut = x \text{ et } v = ty) \text{ ou } (u = xt \text{ et } tv = y)\Big)$
2. En déduire que u et v commutent si et seulement s'il existe $w \in A^*$ et $p, q \in \mathbb{N}^2$ tels que $u = w^p$ et $v = w^q$.

Définition 5

Soit $\mathscr{A}$ un alphabet fini.

- *On appelle langage sur $\mathscr{A}$ une partie de $\mathscr{A}^\star$.*
- *L'ensemble des langages est noté $\mathscr{L}(\mathscr{A})$.*

Définition 5

Soit $\mathscr{A}$ un alphabet fini.

- *On appelle langage sur $\mathscr{A}$ une partie de $\mathscr{A}^\star$.*
- *L'ensemble des langages est noté $\mathscr{L}(\mathscr{A})$.*

On a donc $\mathscr{L}(\mathscr{A}) = \mathscr{P}(\mathscr{A}^\star)$.

Exemples

- L'ensemble vide $\emptyset$ est un langage

Exemples

- L'ensemble vide $\emptyset$ est un langage
- Le singleton $\{\varepsilon\}$ est un langage. **Il ne faut pas le confondre avec le précédent**.

Exemples

- L'ensemble vide $\emptyset$ est un langage
- Le singleton $\{\varepsilon\}$ est un langage. **Il ne faut pas le confondre avec le précédent**.
- Chaque singleton $\{a\}$ où a est un élément de $\mathscr{A}$ ou plus généralement $\{w\}$ où w est un élément de $\mathscr{A}^\star$ est un langage.

Exemples

- L'ensemble vide $\emptyset$ est un langage
- Le singleton $\{\varepsilon\}$ est un langage. **Il ne faut pas le confondre avec le précédent**.
- Chaque singleton $\{a\}$ où a est un élément de $\mathscr{A}$ ou plus généralement $\{w\}$ où w est un élément de $\mathscr{A}^\star$ est un langage.
- On prend $\mathscr{A} = \{a, b\}$. On peut considérer le langage des mots qui commencent par a et finissent par b.

Exemples

- L'ensemble vide $\emptyset$ est un langage
- Le singleton $\{\varepsilon\}$ est un langage. **Il ne faut pas le confondre avec le précédent**.
- Chaque singleton $\{a\}$ où a est un élément de $\mathscr{A}$ ou plus généralement $\{w\}$ où w est un élément de $\mathscr{A}^\star$ est un langage.
- On prend $\mathscr{A} = \{a, b\}$. On peut considérer le langage des mots qui commencent par a et finissent par b.
- Sur un alphabet $\mathscr{A}$ on peut considérer l'ensemble de mots de longueur impaire.

Remarque

La notion de langage va être ce qui va correspondre à nos motifs. On peut se poser la question de chercher dans un texte un mot appartenant à un langage donné ou inversement en se donnant un mot savoir s'il appartient ou non à un langage. On peut par exemple considérer les langages suivants

Remarque

La notion de langage va être ce qui va correspondre à nos motifs. On peut se poser la question de chercher dans un texte un mot appartenant à un langage donné ou inversement en se donnant un mot savoir s'il appartient ou non à un langage. On peut par exemple considérer les langages suivants

- Pour $\mathscr{A}$ qui désigne l'ensemble des signes ASCII. On peut considérer le langage des verbes conjugués à l'imparfait du subjonctif.

Remarque

La notion de langage va être ce qui va correspondre à nos motifs. On peut se poser la question de chercher dans un texte un mot appartenant à un langage donné ou inversement en se donnant un mot savoir s'il appartient ou non à un langage. On peut par exemple considérer les langages suivants

- Pour $\mathscr{A}$ qui désigne l'ensemble des signes ASCII. On peut considérer le langage des verbes conjugués à l'imparfait du subjonctif.
- Pour $\mathscr{A} = \{0, 1\}$ on peut considérer le langage des mots qui correspondent à un nombre (en binaire) divisible par 3.

Définition 6 (Opérations sur les langages)

On définit sur $\mathscr{L}(A)$ deux opérations :

- *L'union noté $\cup$: L'union des langages L_1 et L_2 est le langage $L_1 \cup L_2$. On le note aussi $L_1 + L_2$.*

Définition 6 (Opérations sur les langages)

On définit sur $\mathscr{L}(A)$ deux opérations :

- *L'union noté $\cup$: L'union des langages L_1 et L_2 est le langage $L_1 \cup L_2$. On le note aussi $L_1 + L_2$.*
- *La concaténation : La concaténation des langages L_1 et L_2 noté L_1L_2 (ou $L_1.L_2$) est l'ensemble des mots obtenus comme concaténation d'un mot de L_1 et d'un mot de L_2 :*

$$L_1L_2 = \{w_1w_2 \in \mathscr{A}^\star \mid w_1 \in L_1, w_2 \in L_2\}$$

Remarques

- L'opération de concaténation s'appelle aussi produit

Remarques

- L'opération de concaténation s'appelle aussi produit
- On peut aussi définir l'intersection et la différence de deux langages comme étant les opérations ensemblistes classiques mais c'est moins utile.

Remarques

- L'opération de concaténation s'appelle aussi produit
- On peut aussi définir l'intersection et la différence de deux langages comme étant les opérations ensemblistes classiques mais c'est moins utile.
- Il arrive que l'on note juste a le langage $\{a\}$. Par exemple on notera $a\mathscr{A}^\star$ pour la concaténation du langage $\{a\}$ et du langage $\mathscr{A}^\star$ qui est l'ensemble des mots qui commencent par a.

Règles de calculs

Théorème 4 (Règles de calculs)

Soit L, A et B trois langages,

1. $L.(A+B) = L.A + L.B$ et $(A+B).L = A.L + B.L$
2. $L + \emptyset =$
3. $L.\emptyset = \emptyset.L =$
4. $L.\{\varepsilon\} = \{\varepsilon\}.L =$

Règles de calculs

Théorème 4 (Règles de calculs)

Soit L, A et B trois langages,

1. $L.(A+B) = L.A + L.B$ et $(A+B).L = A.L + B.L$
2. $L + \emptyset = L$
3. $L.\emptyset = \emptyset.L =$
4. $L.\{\varepsilon\} = \{\varepsilon\}.L =$

Règles de calculs

Théorème 4 (Règles de calculs)

Soit L, A et B trois langages,

1. $L.(A+B) = L.A + L.B$ et $(A+B).L = A.L + B.L$
2. $L + \emptyset = L$
3. $L.\emptyset = \emptyset.L = \emptyset$
4. $L.\{\varepsilon\} = \{\varepsilon\}.L =$

Règles de calculs

Théorème 4 (Règles de calculs)

Soit L, A et B trois langages,

1. $L.(A + B) = L.A + L.B$ et $(A + B).L = A.L + B.L$
2. $L + \emptyset = L$
3. $L.\emptyset = \emptyset.L = \emptyset$
4. $L.\{\varepsilon\} = \{\varepsilon\}.L = L$

Démonstration :

- Soit $w \in L.(A+B)$, il s'écrit $w_1 w_2$ où $w_1 \in L$ et $w_2 \in A+B$. Si $w_2 \in A$, $w \in L.A$ et si $w_2 \in B$ alors $w \in L.B$. On en déduit que $w \in L.A + L.B$.
- Réciproquement, $L.A \subset L.(A+B)$ et $L.B \subset L.(A+B)$ donc $L.A + L.B \subset L.(A+B)$.

Exemple

Si on considère $L_1 = \{a\}^\star$ les mots qui ne s'écrivent qu'avec a et $L_2 = \{b\}^\star$ les mots qui ne s'écrivent qu'avec b. Le langage $L_1.L_2$ est l'ensemble des mots de la forme : $a \cdots ab \cdots b$.

On a donc $L_1.L_2 = \{a^n b^m \mid (n, m) \in \mathbb{N}^2\}$

Puissance

Définition 7 (Puissance d'un langage)

Soit L un langage, on définit ses puissances par

$$L^0 = \{\varepsilon\} \ ; \ L^1 = L, \ \textit{et} \ \forall n \in \mathbb{N}, L^{n+1} = L.L^n.$$

Exercice

Soit L un langage. Comparer L^2 et $L' = \{w^2 \mid w \in L\}$.

Exercice

Soit L un langage. Comparer L^2 et $L' = \{w^2 \mid w \in L\}$.

On a $L' \subset L^2$. Par contre il n'y a pas égalité en général. Si w_1 et w_2 sont deux mots distincts de L, $w_1 w_2 \in L^2$ mais $w_1 w_2 \notin L'$ (en général).

Étoile de Kleene

Définition 8 (Opération étoile)

Soit L un langage on note $L^\star$ le langage des mots obtenus en concaténant un nombre fini (éventuellement nul) de mots de L. On a

$$L^\star = \bigcup_{k\in\mathbb{N}} L^k.$$

En particulier, on a toujours $\varepsilon \in L^\star$.

Cette opération s'appelle aussi étoile de Kleene.

Exemple

Pour $\mathscr{A} = \{a, b, c\}$. Si on pose $L_1 = \{a\}$, $L_2 = \{ab, ac\}$ et $L = L_1.(L_2)^\star$. Les mots suivants appartiennent-ils à L?

- ab
- a
- $aababac$
- $aabaac$

Exemple

Pour $\mathscr{A} = \{a, b, c\}$. Si on pose $L_1 = \{a\}$, $L_2 = \{ab, ac\}$ et $L = L_1.(L_2)^\star$. Les mots suivants appartiennent-ils à L?

- ab NON
- a
- $aababac$
- $aabaac$

Exemple

Pour $\mathscr{A} = \{a, b, c\}$. Si on pose $L_1 = \{a\}$, $L_2 = \{ab, ac\}$ et $L = L_1.(L_2)^\star$. Les mots suivants appartiennent-ils à L?

- ab NON
- a OUI
- $aababac$
- $aabaac$

Exemple

Pour $\mathscr{A} = \{a, b, c\}$. Si on pose $L_1 = \{a\}$, $L_2 = \{ab, ac\}$ et $L = L_1.(L_2)^\star$. Les mots suivants appartiennent-ils à L?

- ab NON
- a OUI
- $aababac$ OUI
- $aabaac$

Exemple

Pour $\mathscr{A} = \{a, b, c\}$. Si on pose $L_1 = \{a\}$, $L_2 = \{ab, ac\}$ et $L = L_1.(L_2)^\star$. Les mots suivants appartiennent-ils à L?

- ab NON
- a OUI
- $aababac$ OUI
- $aabaac$ NON

Exercice : Lemme d'Arden

Soient deux langages L et M. On s'intéresse à l'équation $X = L.X + M$ où l'inconnue X est un langage.

1. Vérifier que $X = L^*.M$ est solution de cette équation.
2. Montrer que si X est solution alors $L^*.M \subset X$.
3. Supposons de plus que $\varepsilon \notin L$. Montrer qu'alors la seule solution est $X = L^*.M$ Indication : raisonner par l'absurde et considérer un mot $u \in X \setminus L^*.M$ de longueur minimum.
4. Dans le cas où $\varepsilon \in L$, montrer que tout langage de la forme $L^*.N$ où $M \subset N$ est solution de l'équation $X = L.X + M$
5. Résoudre l'équation $X = X.L + M$ dans le cas où $\epsilon \notin L$.

Langages

Langages rationnels

Définition 9 (Langages rationnels)

Soit $\mathscr{A}$ un alphabet fini.
On note $R(\mathscr{A})$ le plus petit sous-ensemble de $\mathscr{L}(\mathscr{A})$ stable par les opérations suivantes :

- *union de deux langages (et donc d'un nombre fini)*
- *concaténation de deux langages (et donc d'un nombre fini)*
- *étoile*

et contenant

- *le langage vide*
- *le langage $\{\varepsilon\}$ et les langages $\{a\}$ pour $a \in \mathscr{A}$.*

Les langages de $R(\mathscr{A})$ s'appellent les langages rationnels.

Remarque

Cela signifie que les langages rationnels sont les langages que l'on obtient à partir des « atomes », $\emptyset$, $\{\varepsilon\}$ et $\{a\}$ pour $a \in \mathscr{A}$ en appliquant les trois opérations.

Exemples

- Pour tout $w \in \mathscr{A}^\star$ le langage $L = \{w\}$ est rationnel car si $w = w_1 \cdots w_n$, le langage L s'obtient par concaténation des langages $\{w_i\}$ pour $i \in [\![1, n]\!]$.

Exemples

- Pour tout $w \in \mathscr{A}^\star$ le langage $L = \{w\}$ est rationnel car si $w = w_1 \cdots w_n$, le langage L s'obtient par concaténation des langages $\{w_i\}$ pour $i \in [\![1, n]\!]$.
- Tout langage fini est rationnel. Il est obtenu par l'union de ses singletons.

Exemples

- Le langage $\mathscr{A}^\star$ est rationnel.

Exemples

- Le langage $\mathscr{A}^\star$ est rationnel.

  En effet le langage $L = \bigcup_{a \in \mathscr{A}} \{a\}$ des mots de longueur 1 est un langage rationnel car c'est une union (finie) de singletons.
  Maintenant $\mathscr{A}^\star = L^\star$.

Exemples

- Pour $\mathscr{A} = \{a, b\}$ le langage L des mots qui finissent par ab est rationnel.

Exemples

- Pour $\mathscr{A} = \{a, b\}$ le langage L des mots qui finissent par ab est rationnel. On a $L = \mathscr{A}^\star.(\{ab\})$.

Exemples

- Pour $\mathscr{A} = \{a, b\}$ le langage L des mots qui finissent par ab est rationnel. On a $L = \mathscr{A}^\star.(\{ab\})$.
- Le langage des palindromes n'est pas rationnel. Nous le verrons plus tard.

ATTENTION

Attention

1. Si L est un langage rationnel. Un sous-langage de L n'est pas nécessairement rationnel.

ATTENTION

Attention

1. Si L est un langage rationnel. Un sous-langage de L n'est pas nécessairement rationnel.
2. Un langage rationnel n'est pas nécessairement stable par . et $\star$.

Expressions régulières

Définitions

Si on se donne un langage L, il n'est pas aisé de trouver un mot du langage dans un texte ou savoir si un mot donné appartient (ou pas) au langage.

On va donc représenter les langages par des formules.

Commençons par la définition formelle de ces formules. Elle sont définies par induction comme les formules logiques.

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

- *$\emptyset$ est une expression régulière*

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

- *$\emptyset$ est une expression régulière*
- *ε et toute lettre de l'alphabet sont des expressions régulières que l'on dira atomiques*

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

- *$\emptyset$ est une expression régulière*
- *ε et toute lettre de l'alphabet sont des expressions régulières que l'on dira atomiques*
- *si e_1 et e_2 sont des expressions régulières la somme $(e_1|e_2)$ est une expression régulière*

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

- *$\emptyset$ est une expression régulière*
- *ε et toute lettre de l'alphabet sont des expressions régulières que l'on dira atomiques*
- *si e_1 et e_2 sont des expressions régulières la somme $(e_1|e_2)$ est une expression régulière*
- *si e_1 et e_2 sont des expressions régulières le produit $(e_1.e_2)$ est une expression régulière*

Définition 10 (Expressions régulières)

Soit $\mathscr{A}$ un alphabet fini. On définit récursivement l'ensemble $\mathscr{E}(\mathscr{A})$ des expressions régulières par :

- *$\emptyset$ est une expression régulière*
- *ε et toute lettre de l'alphabet sont des expressions régulières que l'on dira atomiques*
- *si e_1 et e_2 sont des expressions régulières la somme $(e_1|e_2)$ est une expression régulière*
- *si e_1 et e_2 sont des expressions régulières le produit $(e_1.e_2)$ est une expression régulière*
- *si e est une expression régulière $e\star$ est une expression régulière.*

Remarques

- On les appellera aussi expressions régulières ou motifs.

Remarques

- On les appellera aussi expressions régulières ou motifs.
- On peut voir $\mathscr{E}(\mathscr{A})$ comme la clôture inductive de l'ensemble $B = \mathscr{A} \cup \{\varepsilon, \emptyset\}$ par les constructeurs $|$ et $.$ d'arité 2 et le constructeur $\star$ d'arité 1.

Remarques

- On les appellera aussi expressions régulières ou motifs.
- On peut voir $\mathscr{E}(\mathscr{A})$ comme la clôture inductive de l'ensemble $B = \mathscr{A} \cup \{\varepsilon, \emptyset\}$ par les constructeurs $|$ et $.$ d'arité 2 et le constructeur $\star$ d'arité 1.
- Pour minimiser les parenthèses, on considère que $\star$ est prioritaire sur le produit, lui même prioritaire sur la somme. Par exemple l'expression :

$$((a|(b\star)).b)|a \text{ se note } (a|b\star).b|a$$

Remarques

- On les appellera aussi expressions régulières ou motifs.
- On peut voir $\mathscr{E}(\mathscr{A})$ comme la clôture inductive de l'ensemble $B = \mathscr{A} \cup \{\varepsilon, \emptyset\}$ par les constructeurs $|$ et $.$ d'arité 2 et le constructeur $\star$ d'arité 1.
- Pour minimiser les parenthèses, on considère que $\star$ est prioritaire sur le produit, lui même prioritaire sur la somme. Par exemple l'expression :

$$((a|(b\star)).b)|a \text{ se note } (a|b\star).b|a$$

- Il arrive que l'on note $+$ pour $|$.

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

- $L(\emptyset) = \emptyset$

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$ *et pour tout* a *de* $\mathscr{A}$, $L(a) = \{a\}$

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$ *et pour tout* a *de* $\mathscr{A}$, $L(a) = \{a\}$
- $\forall (e_1, e_2) \in \mathscr{E}(\mathscr{A})^2,\ L(e_1|e_2) = L(e_1) \cup L(e_2)$

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$ *et pour tout* a *de* $\mathscr{A}$, $L(a) = \{a\}$
- $\forall (e_1, e_2) \in \mathscr{E}(\mathscr{A})^2,\ L(e_1|e_2) = L(e_1) \cup L(e_2)$
- $\forall (e_1, e_2) \in \mathscr{E}(\mathscr{A})^2,\ L(e_1.e_2) = L(e_1).L(e_2)$

Sémantique des expressions régulières

On va associer à toute expression régulière un langage.

Définition 11 (Sémantique des expressions régulières)

On définit par induction structurelle une application L de $\mathscr{E}(\mathscr{A})$ dans $\mathscr{L}(\mathscr{A})$ qui associe à chaque expression régulière un langage.

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$ *et pour tout* a *de* $\mathscr{A}$, $L(a) = \{a\}$
- $\forall (e_1, e_2) \in \mathscr{E}(\mathscr{A})^2,\ L(e_1|e_2) = L(e_1) \cup L(e_2)$
- $\forall (e_1, e_2) \in \mathscr{E}(\mathscr{A})^2,\ L(e_1.e_2) = L(e_1).L(e_2)$
- $\forall e \in \mathscr{E}(\mathscr{A}),\ L(e\star) = L(e)^\star$

Exemples

- Si $\mathscr{A} = \{a, b\}$, $L(a.b\star)$ est le langage $\{a\}.\{b\}^\star$ des mots de la forme ab^k pour $k \in \mathbb{N}$ (éventuellement $k = 0$)

Exemples

- Si $\mathscr{A} = \{a, b\}$, $L(a.b\star)$ est le langage $\{a\}.\{b\}^\star$ des mots de la forme ab^k pour $k \in \mathbb{N}$ (éventuellement $k = 0$)

- Si $\mathscr{A} = \{a, b\}$, $L((a|b)\star)$ est $\mathscr{A}^*$.

Exercice

- Décrire les mots du langage $L(ab\star a\star)$.

- Définir une expression régulière décrivant le langage $\{a^n b^p \mid n \in \mathbb{N}, p \in \mathbb{N}^\star\}$.

Exercice

- Décrire les mots du langage $L(ab\star a\star)$.

  Ce sont les mots de la forme $ab^n a^m$ où $n \geq 0$ et $m \geq 0$.

- Définir une expression régulière décrivant le langage $\{a^n b^p \mid n \in \mathbb{N}, p \in \mathbb{N}^\star\}$.

Exercice

- Décrire les mots du langage $L(ab\star a\star)$.

  Ce sont les mots de la forme $ab^n a^m$ où $n \geq 0$ et $m \geq 0$.

- Définir une expression régulière décrivant le langage $\{a^n b^p \mid n \in \mathbb{N}, p \in \mathbb{N}^\star\}$.

  On peut prendre $a\star bb\star$.

Notations

Il existe de nombreuses versions d'expressions régulières avec des syntaxes plus enrichies. Malheureusement il n'existe pas de normalisation universelle. Voici quelques exemples les plus courants :

Notations

Il existe de nombreuses versions d'expressions régulières avec des syntaxes plus enrichies. Malheureusement il n'existe pas de normalisation universelle. Voici quelques exemples les plus courants :

- Le symbole . désigne toute lettre de l'alphabet. C'est l'expression qui est la somme de toutes les constantes atomiques sauf le mot vide. Par exemple pour $\mathscr{A} = \{a, b, c\}$, . désigne $(a|b|c)$.

Notations

Il existe de nombreuses versions d'expressions régulières avec des syntaxes plus enrichies. Malheureusement il n'existe pas de normalisation universelle. Voici quelques exemples les plus courants :

- Le symbole . désigne toute lettre de l'alphabet. C'est l'expression qui est la somme de toutes les constantes atomiques sauf le mot vide. Par exemple pour $\mathscr{A} = \{a, b, c\}$, . désigne $(a|b|c)$.
- Si e est une expression, le symbole $e+$ désigne la répétition au moins une fois d'un motif de e. C'est donc $ee^\star$.

Notations

Il existe de nombreuses versions d'expressions régulières avec des syntaxes plus enrichies. Malheureusement il n'existe pas de normalisation universelle. Voici quelques exemples les plus courants :

- Le symbole . désigne toute lettre de l'alphabet. C'est l'expression qui est la somme de toutes les constantes atomiques sauf le mot vide. Par exemple pour $\mathscr{A} = \{a, b, c\}$, . désigne $(a|b|c)$.
- Si e est une expression, le symbole $e+$ désigne la répétition au moins une fois d'un motif de e. C'est donc $ee\star$.
- On note pour finir $e?$ pour $(e|\varepsilon)$ à savoir la répétition au plus une fois d'un motif.

Définition 12

Soit $L \subset \mathscr{P}(\mathscr{A}^*)$ un langage.
On dit que L est un langage régulier s'il existe une expression régulière e telle que $L = L(e)$.

Théorème 5

Soit $L \subset \mathscr{P}(\mathscr{A}^*)$ un langage.
Il est régulier si et seulement s'il est rationnel.

Démonstration

Si on note B l'ensemble des langages réguliers.

- Si e est une expression régulière alors $L(e)$ se déduit des langages atomiques par les opérations $\cup, .$ et $\star$. On en déduit que $L(e) \in \mathscr{R}(\mathscr{A})$. Cela signifie que $B \subset \mathscr{R}(\mathscr{A})$; un langage régulier est rationnel.

Démonstration

Si on note B l'ensemble des langages réguliers.

- Si e est une expression régulière alors $L(e)$ se déduit des langages atomiques par les opérations $\cup, .$ et $\star$. On en déduit que $L(e) \in \mathscr{R}(\mathscr{A})$. Cela signifie que $B \subset \mathscr{R}(\mathscr{A})$; un langage régulier est rationnel.

- L'ensemble B contient le langage vide, le langage $\{\varepsilon\}$ et les langages $\{a\}$ pour $a \in \mathscr{A}$. De plus, il est stable par $\cup, .$ et $\star$ donc $\mathscr{R}(\mathscr{A}) \subset B$; un langage rationnel est régulier.

Exemple

On a $\mathscr{A} = \{a, b, c\}$. Déterminer une expression régulière e telle $L(e)$ soit le langage des mots dans lesquels c n'apparaît jamais juste à gauche d'un b.

Exemple

Il faut s'assurer qu'après un (ou plusieurs c) il doit y avoir un (ou plusieurs) a sauf à la fin du mot.

On peut prendre $(c\star a|b|a)\star c\star$.

ATTENTION

Attention

Comme dans le cas des formules logiques, il est important de faire la différence entre la syntaxe d'une expression régulière (son écriture) et la sémantique (le langage qui lui est associé).

Deux expressions régulières différentes peuvent définir le même langage. Par exemple $(a|b)\star$ et $\varepsilon|(a|b).(a|b)\star$

Equivalence sémantique

Définition 13

Deux expressions régulières définissant le même langage sont dites sémantiquement équivalentes.

Equivalence sémantique

Définition 13

Deux expressions régulières définissant le même langage sont dites sémantiquement équivalentes.

Si e_1 et e_2 sont des expressions régulières équivalentes (c'est-à-dire $L(e_1) = L(e_2)$) on note $e_1 \equiv e_2$.

Implémentation des expressions régulières

Implémentation

On peut utiliser l'implémentation

```
type expr_rat = Vide
   | Epsilon
   | Mot of string
   | Produit of expr_rat list
   | Somme of expr_rat list
   | Etoile of expr_rat ;;
```

Dans l'implémentation ci-dessus :

- On utilise `Mot of string`plutôt que `Lettre of char`ce qui permet d'utiliser `Mot("ab")`plutôt que `Produit(Lettre('a'),Lettre('b'))`.

Implémentation

Dans l'implémentation ci-dessus :

- On utilise `Mot of string`plutôt que `Lettre of char`ce qui permet d'utiliser `Mot("ab")`plutôt que `Produit(Lettre('a'),Lettre('b'))`.
- On utilise des listes pour les produits et les sommes en utilisant l'associativité. La encore on peut utiliser `Somme([Mot("a"); Mot("b") ; Mot("c")]`plutôt que `Somme(Somme(Mot("a"),Mot("b")),Mot("c"))`.