Les Langages

Chapitre 4

1 Introduction

1.1 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[1]) dans une chaine de de caractères

Dans la suite nous travaillerons essentiellement sur la recherche d'un motif 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.

1.2 Algorithme naif

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.

— 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.

Exercices :

1. 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`.

1. La notion de motif sera précisée plus loin.

2. Déterminer la complexité de cette fonction par rapport à la taille n du texte et la taille p du mot.

3. 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`.

2 Langages

2.1 Alphabet et mots

Définition 2.1.1

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 $\mathscr{A}$

- 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 $w = abababac$ est de longueur 8, on a $|w|_a = 4$.

Définition 2.1.2 (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_1w_2 est le mot obtenu en mettant bout à bout les mots w_1 et w_2. On a donc

$$\forall k \in [\![0\,;\, |w_1|+|w_2|-1]\!], (w_1w_2)[k] = \begin{cases} w_1[k] & si\ k < |w_1| \\ w_2[k-|w_1|] & si\ k \geqslant |w_1| \end{cases}$$

Proposition 2.1.3

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

Remarque : On dit que $(\mathscr{A}^\star, .)$ est un monoïde (ou magma associatif unifère).
Exercice : $(\mathscr{A}^\star, .)$ est-il un groupe ?

Proposition 2.1.4

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

$$\forall (w_1, w_2) \in (\mathscr{A}^\star)^2, |w_1w_2| = |w_1| + |w_2| \text{ et } |w_1w_2|_a = |w_1|_a + |w_2|_a.$$

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

Définition 2.1.5 (Puissances)

Soit w un mot, on définit ses puissances par

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

Définition 2.1.6 (Facteurs, préfixes et suffixes)

Soit w un mot sur $\mathscr{A}^\star$.
— *S'il existe des α, β et u 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 $w = abababac$ alors ab est un préfixe de w, ba est un facteur de w et c un suffixe de w.

Proposition 2.1.7

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

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

2.2 Langages

Définition 2.2.8

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

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

1. L'ensemble vide $\emptyset$ est un langage
2. Le singleton $\{\varepsilon\}$ est un langage. Il ne faut pas le confondre avec le précédent.
3. 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.
4. On prend $\mathscr{A} = \{a, b\}$. On peut considérer le langage des mots qui commencent par a et finissent par b.
5. 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

— 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 2.2.9 (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 L_1 et L_2 notée 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 :

1. La concaténation s'appelle aussi produit.
2. On peut aussi définir l'intersection et la différence de deux langages comme étant les opérations ensemblistes classiques mais c'est moins utile.
3. 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.

Proposition 2.2.10 (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. $L.\emptyset =$
4. $L.\{\varepsilon\} = \{\varepsilon\}.L =$

Démonstration :

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 : $a \cdots ab \cdots b$.

Définition 2.2.11 (Puissance d'un langage)

Soit L un langage, on définit ses puissances par

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

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

Définition 2.2.12 (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

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

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

Remarque : 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
— $aababac$
— $aabaac$

3 Langages rationnels et expressions régulières

3.1 Langages rationnels

Définition 3.1.13 (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 $\{\varepsilon\}$,*
— *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 :

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 $\star$.

3.2 Expressions régulières

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 3.2.14 (Expressions régulières)

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

1. On les appellera aussi expressions rationnelles ou motifs.
2. 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.
3. Pour minimiser les parenthèses, on considère que $\star$ est prioritaire sur le produit, lui même prioritaire sur la somme. Par exemple l'expression :
$$((a|(b\star)).b)|a \text{ se note } (a|b\star).b|a$$
4. Il arrive que l'on note + pour $|$.

On va associer à toute expression régulière un langage.

Définition 3.2.15 (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.

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

Exemples :

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

Notation : 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 3.2.16

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 $L = L(e)$.

Proposition 3.2.17

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

Démonstration :

□

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.

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 $\varepsilon|(a|b).(a|b)\star$

Définition 3.2.18

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

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

3.3 Implémentation des expressions régulières

On peut utiliser l'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"))`.