Chapitre 4

Langages Rationnels

Nous allons dans ce chapitre voir un type très simple de langage : les langages *rationnels*.

4.1 Recherche de motif

Le concept de motif est quelque chose de très vaste : on appellera *motif* un ensemble de propriétés vérifiées par certains objets. Par exemple, on peut reconnaître un rectangle par ses droites deux à deux parallèles. La recherche de motif a de nombreuses applications : reconnaissance faciale, recherche de séquences de gènes dans l'ADN, lecture de code-barres ou de QR-code, *etc.*

Nous ne nous intéresserons ici qu'à des motifs uni-dimensionnels, qu'on verra comme des langages sur un certain alphabet.

4.1.1 Recherche d'un mot dans un texte

La plus simple des recherche de motifs est celle de la recherche d'un unique mot dans un texte. On donne l'algorithme suivant :

```
let rec prefixe u v =






let rec est_dans u l =
```

Cet algorithme naïf n'est pas très efficace (complexité); on peut l'améliorer avec l'algorithme de Knuth-Morris-Pratt.

4.1.2 Multiples de 3 et base 2

Voyons l'exemple suivant : on veut savoir si un mot sur $\Sigma = \{0, 1\}$ est un multiple de 3 et base 2.

Commençons par écrire une fonction qui calcule le reste modulo 3, en remarquant que

Lemme 4.1.1
Soit $u = x_n \dots x_0$ un mot binaire. Notons $\varphi(u) = \sum xi2^i$ le nombre décimal représenté par u. Alors

Ainsi, si $\varphi(u) \equiv i[3]$, alors

On en déduit la fonction

```
let rec modulo3 u s =




let multiple3 u =
```

4.2 Langages rationnels

4.2.1 Expressions rationnelles

Définition 4.2.1
Soit Σ un alphabet. Les expressions rationnelles (ou régulières) sont définies inductivement par

-
-

•

EXEMPLE

Sur $\Sigma = \{0, 1\}$, les expressions ou sont des expressions rationnelles.

On peut alors simplement définir le type des expressions rationnelles :

```
type regexp =
```

4.2.2 Langages rationnels

Définition 4.2.2
On note $\mathcal{R}\text{at}(\Sigma)$

On pourrait évidemment donner une définition par le bas de $\mathcal{R}\text{at}$: on note $\mathcal{R}\text{at}_0$ l'ensemble , et

$\mathcal{R}\text{at}_{n+1} =$

Alors

$$\mathcal{R}\text{at} =$$

EXEMPLE

Sur $\Sigma = \{0, 1\}$, le langage est rationnel.

Il y a un lien entre les expressions et les langages rationnels, obtenu en définissant l'interprétation d'un langage rationnel : notons $\Phi(u)$ l'interprétation de l'expression rationnelle u

- $\Phi(\emptyset) =$
- $\Phi(\varepsilon) =$
- pour toute lettre a, $\Phi(a) =$
- pour toutes expressions rationnelle u et v,
  - $\Phi(u+v) =$
  - $\Phi(uv) =$
  - $\Phi(u^*) =$

La fonction Φ est alors bien définie, surjective dans $\mathcal{R}$at, mais pas injective.

EXERCICE

Donner deux expressions rationnelles dénotant le même langage rationnel.

On confondra donc dans la suite les langages rationnels et les expressions rationnelles dénotant ce langage, en faisant bien attention à la non-injectivité de l'interprétation.

On se passera souvent du symbole $\emptyset$ dans les expressions rationnelles grâce à la propriété suivante :

Proposition 4.2.3
Si le langage dénoté par une expression rationnelle u est non vide, alors

4.3 Langages locaux

On va dans cette partie voir un certain type de langages, qu'on appelle *langage local*. L'intérêt de ces langages restera mystérieux jusqu'au chapitre sur les automates finis. . .

Considérons un langage L sur un alphabet Σ. On note alors

- $P(L) =$
- $S(L) =$
- $F(L) =$
- $N(L) =$

Ces ensembles sont, dans l'ordre,

Il est alors clair que

$$L\backslash\{\varepsilon\} \subseteq$$

Définition 4.3.1
Soit L un langage sur Σ. On dit que L est local

Proposition 4.3.2
Si un langage L est local, alors

EXEMPLE

- Sur $\Sigma = \{0, 1\}$, le langage 0^* est local :
- Sur $\Sigma = \{0, 1, 2\}$, le langage $(012)^*$ est local :
- Sur $\Sigma = \{0, 1\}$, le langage $0^*(01)^*$ n'est pas local :

Les langages locaux ont des propriétés de clôture intéressantes :

Proposition 4.3.3
L'intersection de langages locaux est un langage local.

Démonstration. Soient donc L_1 et L_2 deux langages locaux. On note $P = P(L_1) \cap P(L_2)$, $S = S(L_1) \cap S(L_2)$ et $N = N(L_1) \cup N(L_2)$. On a alors

$$\begin{aligned}(L_1 \cap L_2)\backslash\{\varepsilon\} &= \\ &= \\ &= \\ &= \end{aligned}$$

□

Proposition 4.3.4
L'étoile de Kleene d'un langage local est un langage local.

Démonstration. Soit L un langage local. Il suffit de noter que la première lettre d'un mot de L^* est nécessairement
Les facteurs de longueur 2 sont
On pose donc

$$P = \qquad , \quad S = \qquad , \quad F = \qquad ,$$

et il suffit de montrer qu'on a bien l'égalité voulue. □

Pour l'union et la concaténation, c'est un peu plus compliqué :

Proposition 4.3.5
L'union de deux langages locaux définis sur des alphabets disjoints est un langage local.

Démonstration. Notons Σ l'union des deux alphabets. On pose ensuite

$P = \qquad , \quad S = \qquad , \quad F =$

Soit donc $u = u_1 \dots u_p \in (P\Sigma^* \cap \Sigma^* S) \backslash (\Sigma^* N \Sigma^*)$. Alors nécessairement, $u_1 \in P$, et donc $u_1 \in P(L_2)$ ou $u_1 \in P(L_2)$. Supposons qu'on est dans le premier cas.

On a alors . Mais $u_1 \in \Sigma_1$, et $F(L_1) \subset \Sigma_1^2$, et donc , puis

Une récurrence immédiate montre que , et donc

On a donc $u \in L_1$. □

Proposition 4.3.6
La concaténation de deux langages locaux définis sur des alphabets disjoints est un langage local.

Démonstration. En exercice. □

Définition 4.3.7
Une expression rationnelle est dite linéaire *si*

Un langage qui peut être dénoté par une expression rationnelle linéaire sera dit

EXEMPLE

Les expressions $(0+1)^*$ et $01+2$ sont linéaires, mais pas $11+0$ ou $(1+1)^*$.

Théorème 4.3.8
Tout langage linéaire est local. La réciproque est fausse.

Démonstration. Par induction sur l'expression rationnelle linéaire u dénotant le langage :

- si $u = \emptyset$ ou $u = \varepsilon$,
- si $u = a \in \Sigma$, alors
- si $u = v + w$ (resp. $u = vw$), alors v et w sont linéaires, et donc dénotent des langages locaux par hypothèse d'induction. Mais
- si $u = v^*$, alors v est linéaire et dénote donc un langage local.

Pour la réciproque, on peut considérer le langage sur $\Sigma = \{a\}$. Il est bien local , mais ne peut pas être dénoté par une expression linéaire. □

Donnons maintenant les algorithmes permettant, étant donnée une expression rationnelle linéaire, de calculer les ensembles P, S et F du langage local associé.

Il faut commencer par vérifier si un mot est vide :

```
let rec contient_epsilon(u) =
```

On peut maintenant facilement calculer les ensembles P et S :

```
let rec prefixe(u) =
```

```
let rec suffixe(u) =
```

Pour l'ensemble F, il suffit de remarquer que les seuls cas non triviaux sont la concaténation et l'étoile de Kleene. Dans le premier cas, un facteur de longueur 2 est soit un facteur de longueur 2 d'un des langages, soit une dernière lettre du premier langage et une première du second. Dans le deuxième cas, c'est soit un facteur de longueur 2 du langage, soit une dernière lettre et une première lettre. Il faut donc une fonction calculant tous les couples (dernier, premier) possible.

```
let rec produit l1 l2 =
```

On en déduit la dernière fonction :

```
let rec facteur u =
```

4.4 Exercices

Exercice 1. Montrer que les langages suivants sont des langages rationnels :

- sur $\Sigma = \{0, 1\}$, le langage des mots qui représentent un nombre pair
- sur $\Sigma = \{0, 1\}$, le langage des mots qui représentent une puissance de 2
- sur $\Sigma = \{0, 1\}$, le langage des mots normalisés, *i.e.* les mots qui commencent par 1 ou aui sont exactement 0

Exercice 2. Soit $\Sigma = \{0, 1\}$. On définit la suite de mots (f_n) par $f_0 = 1$, $f_1 = 0$ et $f_{n+2} = f_{n+1} f_n$

a) Montrer que pour $n \geqslant 2$, f_n se termine par 01 si n est pair et par 10 sinon.

b) Montrer qu'en enlevant les deux dernières lettres de f_n, on obtient un palindrome.

Exercice 3. Soient les langages $L_1 = (b^* a^2 b^*)^*$ et $L_2 = (a^* b^2 a^*)^*$. Montrer que l'intersection de L_1 et L_2 est rationnelle. Montrer que le complémentaire de L_1 est rationnel.

Exercice 4. On veut montrer le lemme d'Arden :

Lemme 4.4.1
Soient A et B deux langages sur le même alphabet, $\varepsilon \notin A$. Alors l'équation $L = AL + B$ admet pour unique solution $L = A^* B$.

a) Sur $\Sigma = \{a, b\}$, montrer qu'il existe un unique langage L vérifiant $L = aL + b$, et qu'il s'agit de $L = a^* b$.

b) Montrer le lemme d'Arden.

c) On se place sur $\Sigma = \{a, b\}$. Soient P le langage des mots ayant un nombre pair de b, et I celui des mots ayant un nombre impair de b.
Déterminer P et I avec le lemme d'Arden.

Exercice 5. Montrer qu'un langage L sur Σ est local si et seulement si

$$\forall u, v, u', v' \in \Sigma^*,\ \forall a \in \Sigma,\ (uav \in L \land u'av' \in L) \Rightarrow uav' \in L.$$