Langages, expressions rationnelles, langages locaux

Ex. 1 – Concaténation et intersection

Soient L, M et P trois langages.

Montrer que $L(M \cap P) \subset LM \cap LP$, mais qu'en général on n'a pas égalité.

Ex. 2 – Expressions rationnelles et alphabet binaire

On considère l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Donner une expression rationnelle du langage des mots de Σ^* :

1. commençant par `aa`, se terminant par `a`, et contenant exactement deux `b` ;

2. d'au moins deux lettres où les éventuels `a` précèdent les éventuels `b` ;

3. contenant au plus deux `b` ;

4. contenant à la fois le facteur `aa` et le facteur `bb` ;

5. contenant à la fois le facteur `ab` et le facteur `ba` ;

6. contenant le facteur `aa` ou le facteur `bb` mais pas les deux ;

7. contenant exactement deux fois le facteur `aa`.

Ex. 3 – Relations sur le nombre d'occurrences

On s'intéresse aux langages sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

Soit f une application définie sur $\mathbb{N}$ à valeurs dans $\mathbb{N}$. On note $L(f)$ le langage des mots u de Σ^* vérifiant $|u|_{\mathtt{a}} = f(|u|_{\mathtt{b}})$.

Soit f_1 définie par $\forall n \in \mathbb{N},\ f_1(n) = 2$.

1. Proposer une expression rationnelle s'interprétant en $L(f_1)$.

Soit f_2 définie par $\forall n \in \mathbb{N},\ f_2(n) = 1$ si n est pair, et $f_2(n) = 0$ sinon.

2. Proposer une expression rationnelle s'interprétant en $L(f_2)$.

Ex. 4 – Expressions rationnelles et alphabet ternaire

On se place sur l'alphabet ternaire $\Sigma = \{0, 1, 2\}$.

1. Déterminer une expression rationnelle des mots qui correspondent à l'écriture ternaire d'un entier qui n'est pas divisible par 3.

2. Déterminer une expression rationnelle des mots qui correspondent à l'écriture ternaire d'un entier pair.

3. Déterminer une expression rationnelle des mots où deux symboles consécutifs sont toujours différents.

Ex. 5 – Plis de feuille

On s'intéresse à l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. On définit sur Σ^* l'application $r \mapsto \overline{r}$ par

$$\begin{cases} \overline{\varepsilon} \mapsto \varepsilon \\ \forall w \in \sigma^*, \quad \overline{\mathtt{a}w} = \overline{w}\mathtt{b} \quad \text{et} \quad \overline{\mathtt{b}w} = \overline{w}\mathtt{a} \end{cases}$$

1. Déterminer $\overline{\mathtt{aababa}}$.

2. Montrer que, pour tout $u, v \in (\Sigma^*)^2$, on a $\overline{uv} = \overline{v}\,\overline{u}$.

On définit Σ en OCaml par le type

```ocaml
type alphabet = A | B;;
```

Un mot de Σ^* sera un objet de type `alphabet list`.

3. Définir la fonction $r \mapsto \overline{r}$ en OCaml. On cherchera une solution de complexité linéaire en la longueur de la liste.

On prend une feuille de papier, et on replie n fois la moitié droite sur la moitié gauche. Puis on redéplie la feuille. On observe des plis qui forment une série de creux et de bosses. En assimilant un creux à `a` et une bosse à `b`, et en les notant de la gauche vers la droite, on peut associer le pliage à un mot de Σ^*.

On notes w_n cette suite de mots. Pour $n \in [\![0\,..\,3]\!]$, il s'agit de ε, `a`, `aab` et `aabaabb`.

4. Déterminer le lien entre w_n et w_{n+1}, et en déduire une fonction OCaml de signature `int -> alphabet list` calculant w_n.

Un mot w_n étant le préfixe de tous les mots le suivant dans la suite, on peut les considérer tous comme les préfixes d'un « mot infini » $w = s_1 s_2 s_3 \ldots$ où s_k désigne le k^{e} symbole de w.

5. Que vaut s_k si k est impair ?

6. Déterminer un lien entre les symboles s_{2k} et s_k.

7. En déduire une fonction de signature `int -> alphabet` retournant s_k.

8. Que vaut s_{2025} ?

Ex. 6 – Détermination

Parmi les expressions rationnelles suivantes, lesquelles s'interprêtent en un langage local ? Lorsque c'est le cas, exhiber des ensembles P, S et N qui les caractérisent.

`aa*` `ab*` `(a|b)*` `(ab)*` `(abc)*` `(aba)*` `(aa|b)*` `(ab|a)*`

Ex. 7 – Cardinaux

1. Proposer un langage local L sur $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ tel que $|L| = 2$.

2. Même question pour $|L| = 3$ et $|L| = 4$.

3. Proposer un majorant du nombre de langages locaux distincts sur $\{\mathtt{a}, \mathtt{b}\}^*$ (l'énumération exacte est longue).

Ex. 8 – Facteurs

Montrer que si L est un langage local, le langage des facteurs des mots de L est local.

Ex. 9 – Caractérisation

Soit un langage L sur un alphabet Σ. Montrer que L est un langage local si et seulement si pour tous $u, v, u', v' \in (\Sigma^*)^4$ et pour tout $s \in \Sigma$, on a

$$\left(u \cdot s \cdot v \in L \text{ et } u' \cdot s \cdot v' \in L\right) \implies u \cdot s \cdot v' \in L$$

Ex. 10 – Pompage

Soit Σ un alphabet[1] et L un langage local sur Σ.

1. Montrer que si L contient un mot w tel que $w = x \cdot y \cdot y \cdot z$ avec $x, y, z \in \Sigma^* \times \Sigma^+ \times \Sigma^*$, alors L est infini.

2. Montrer que si L contient un mot w incluant deux fois un même symbole de Σ, alors L est infini.

3. En déduire un majorant sur la taille des mots de L si L est fini, et proposer un langage local fini ayant un mot de cette taille.

Ex. 11 – Codes

On appelle *code* sur un alphabet Σ tout langage L tel que toute factorisation d'un mot de L^* est unique, c'est-à-dire qui vérifie

$$\left(u_1 u_2 \dots u_n = v_1 v_2 \dots v_p \quad \text{avec} \quad u_i, v_j \in L\right) \Rightarrow \left(p = q \quad \text{et} \quad \forall i \in [\![1\,..\,n]\!], u_i = v_i\right)$$

1. On rappelle que l'on travaille sur des alphabets finis.

1. Proposer un code de quatre mots sur $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

2. Déterminer quels sont les codes parmi :

$$\{\mathtt{a}, \mathtt{ab}\}$$

$$\{\mathtt{a}, \mathtt{ab}, \mathtt{ba}\}$$

$$\{\mathtt{ab}, \mathtt{baa}, \mathtt{abba}, \mathtt{aabaa}\}$$

$$\{\mathtt{b}, \mathtt{ab}, \mathtt{baa}, \mathtt{abaa}, \mathtt{aaaa}\}$$

$$\{\mathtt{a}, \mathtt{ba}, \mathtt{bba}, \mathtt{baab}\}$$

3. Montrer que L ne contient pas le mot-vide ;

4. Montrer que si tous les mots de L ont même longueur (non nulle), alors L est un code.

5. Montrer que si $L = \{u, v\}$, L est un code si et seulement si $u \cdot v \neq v \cdot u$.

6. Soit un langage L tel qu'aucun mot de L ne soit un préfixe propre d'un mot de L. Montrer que L est un code.