Université de Paris — UFR Informatique
Automates et Analyse Lexicale — Année 2021–2022

# Handout Cours 1

## Mots, Langages, et Expressions Rationnelles

## 1 Mots

### 1.1 Définitions de base

Un *alphabet* est un ensemble fini de *lettres* (ou *symboles*).

Un *mot* sur un alphabet $\Sigma$ est une séquence de lettres de l'alphabet $\Sigma$. On note

- $\epsilon$ (prononcé "epsilon") pour le *mot vide*. C'est la séquence de longueur 0 qui ne contient aucune lettre.
- $\Sigma^*$ pour l'ensemble de tous les mots sur l'alphabet $\Sigma$.
- $|w|$ pour la longueur du mot $w$. Pour tous les mots $w$, $|w|$ est un nombre entier non-négatif qui peut être 0.

Suivant la tradition dans les langages de programmation, les positions dans un mot $w$ sont numérotées de 0 jusqu'à $|w|-1$. Si $w$ est un mot et $0 \leq i < |w|$, nous écrivons $w[i]$ pour la lettre de $w$ à la position $i$.

Exemple: Soit l'alphabet $\Sigma = \{a, b, c, d\}$.

$w = ababca$ est un mot sur l'alphabet $\Sigma$, c.-à-d. $w \in \Sigma^*$. $|w| = 6$, et on a que $w[3] = b$.

### 1.2 Mesures sur des mots

Nous avons déjà vu la notion de *longueur* $|w|$ d'un mot $w$. Si $w$ est un mot sur $\Sigma$ et $a \in \Sigma$, alors $|w|_a$ est le *nombre d'occurrences de a dans w*. L'ensemble des *occurrences* de la lettre $a$ dans le mot $w$ est $\{i \mid w[i] = a\}$.

### 1.3 Opérations sur des mots

Soit un alphabet $\Sigma$. La *concaténation* de deux mots $v = a_0 \dots a_{n-1} \in \Sigma^*$ et $w = b_0 \dots b_{m-1} \in \Sigma^*$, est le mot $a_0 \dots a_{n-1} b_0 \dots b_{m-1}$ (attention les mots $v$ et $w$ peuvent être vides). On note $v \cdot w$ pour la concaténation de $v$ et $w$, très souvent on n'écrit pas l'opérateur $\cdot$, et on écrit $vw$ à la place de $v \cdot w$.

Par exemple, $abab \cdot cd = ababcd$

Propriétés importantes de la concaténation :

- pour tous mots $w$ : $\epsilon \cdot w = w$
- pour tous mots $w$ : $w \cdot \epsilon = w$
- pour tous mots $u, v, w$ : $u \cdot (v \cdot w) = (u \cdot v) \cdot w$

Si $w$ est un mot et $n$ un nombre non-négatif, alors nous écrivons $w^n$ pour

$$\underbrace{w \cdots w}_{n}$$

Si $w = a_0 \dots a_{n-1}$ est un mot de longueur $n$, alors son *miroir*, noté $\overline{w}$, est le mot $\overline{w} = a_{n-1} \cdots a_0$.

### 1.4 Relations entre des mots

Soit $\Sigma$ un alphabet. On a les relations suivantes entres des mots sur $\Sigma$ :

- $v$ est un *préfixe* de $w$ s'il existe un mot $u$ tel que $w = v \cdot u$. Il s'agit d'un *préfixe propre* quand $u \neq \epsilon$.
- $v$ est un *suffixe* de $w$ s'il existe un mot $u$ tel que $w = u \cdot v$. Il s'agit d'un *suffixe propre* quand $u \neq \epsilon$.
- $v$ est un *facteur* de $w$ s'il existe deux mots $x$, $y$ tels que $w = x \cdot v \cdot y$.
- $v$ est un *sous-mot* de $w$ quand $v$ est obtenu de $w$ en supprimant certains positions de $w$. Autrement dit, $v$ est un sous-mot de $w$ quand il y a une décomposition de $w$ en facteurs

  $$w = w_0 v_0 \cdots w_{n-1} v_{n-1} w_n$$

  tel que

  $$v = v_0 \cdots v_{n-1}$$

## 2 Langages

Un *langage* sur un alphabet $\Sigma$ est simplement un sous-ensemble de $\Sigma^*$. Autrement dit, un langage sur $\Sigma$ est un ensemble de mots sur $\Sigma$.

Toutes les notions ensemblistes s'appliquent donc aux langages : $\emptyset$ est le langage vide, $L_1 \cup L_2$ est l'union des deux langages $L_1$ et $L_2$, et $L_1 \cap L_2$ est leur intersection. Quand on parle du complément d'un langage $L$ il faut être précis sur l'alphabet sous-jacent, car le complément du langage $L$ par rapport à l'alphabet $\Sigma$ est

$$L^{comp} = \{w \in \Sigma^* \mid w \notin L\}$$

Plus intéressant sont les opérations qui sont spécifiques aux langages :
La *concaténation* de deux langages $L_1$ et $L_2$ est définie par

$$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

Cela nous permet de définir, pour un langage $L$ et $n \geq 0$ :

$$L^n = \underbrace{L \cdot \ldots \cdot L}_{n \text{ fois}}$$

et finalement

$$L^* = \bigcup_{n \geq 0} L^n$$

L'opérateur $*$ est appelé *l'étoile de Kleene*, en honneur du mathématicien Stephen C. Kleene.

## 3 Expressions Rationnelles

L'ensemble des *expressions rationnelles* sur un alphabet $\Sigma$ est défini par induction. C'est le plus petit ensemble RAT tel que

- $\epsilon \in \text{RAT}$
- $\emptyset \in \text{RAT}$
- pour tout $a \in \Sigma : a \in \text{RAT}$
- si $r_1, r_2 \in \text{RAT}$, alors $(r_1 + r_2) \in \text{RAT}$
- si $r_1, r_2 \in \text{RAT}$, alors $(r_1 r_2) \in \text{RAT}$
- si $r \in \text{RAT}$, alors $r^* \in \text{RAT}$

Les anglophones les appellent *regular expressions*.

On associe à chaque expression rationnelle $r$ un langage $\mathcal{L}(r)$, sa sémantique. Cette association est formalisée par une fonction définie par récurrence :

- $\mathcal{L}(\epsilon) = \{\epsilon\}$
- $\mathcal{L}(\emptyset) = \{\}$ (l'ensemble vide)
- si $a \in \Sigma$, alors $\mathcal{L}(a) = \{a\}$
- $\mathcal{L}((r_1 + r_2)) = \mathcal{L}(r_1) \cup \mathcal{L}(r_2)$
- $\mathcal{L}((r_1 r_2)) = \mathcal{L}(r_1) \cdot \mathcal{L}(r_2)$
- $\mathcal{L}(r^*) = (\mathcal{L}(r))^*$

Quand on écrit des expressions rationnelles on se permet d'omettre des parenthèses qui ne sont pas utiles, sachant que les deux opérateurs binaires $+$ et $\cdot$ sont associatives, et que $\cdot$ prend la priorité sur $+$.

Quelques raccourcis ("sucre syntaxique") fréquents :

- $r?$ est une abréviation pour $r + \epsilon$
- $r^+$ est une abréviation pour $rr^*$

Un langage $L$ est *rationnel* quand il existe une expression rationnelle $r$ telle que $L = \mathcal{L}(r)$.