Langages formels (11X003) - Automne 2025

1. Concepts de bases

Enseignant: Arnaud Casteigts

Assistants: A.-Q. Berger & M. Marseloo
Monitrices: N. Beghdadi & E. Bussod

1.1 Alphabet et mot

Un **alphabet** Σ est un ensemble fini de symboles (aussi appelés caractères) comme par exemple des lettres ou des chiffres. Par exemple,

- L'alphabet binaire $\Sigma_1 = \{\mathtt{0}, \mathtt{1}\}$
- L'alphabet conventionnel $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- L'alphabet arithmétique $\Sigma_3 = \{+, -, *, /, (,), 0, \ldots, 9\}$
- Un alphabet quelconque $\Sigma_4 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

Un **mot** défini sur un alphabet Σ est une suite finie de symboles de Σ. On parle aussi de chaîne de caractère. Par exemple, $u = \mathtt{abba}$ et $v = \mathtt{baba}$ sont deux mots sur l'alphabet $\{\mathtt{a}, \mathtt{b}\}$. La **longueur** d'un mot u est notée $|u|$, par exemple $|\mathtt{abba}| = 4$. Il existe un mot de longueur zéro, appelé **mot vide** et noté ε.

La **concaténation** de deux mots $u = a_1 a_2 \ldots a_n$ et $v = b_1 b_2 \ldots b_m$ est l'opération qui consiste à coller v à la fin de u en formant un nouveau mot $a_1 \ldots a_n b_1 \ldots b_m$. On écrit cette opération $u \cdot v$ ou simplement uv. La concaténation est une opération *associative*, c'est à dire que $(u \cdot v) \cdot w = u \cdot (v \cdot w)$, mais elle n'est pas *commutative*, car en général $u \cdot v \neq v \cdot u$. Enfin, le mot vide ε est l'élément neutre de la concaténation : pour tout mot w, on a bien $w \cdot \varepsilon = \varepsilon \cdot w = w$.

On peut concaténer un mot avec lui-même plusieurs fois, on parle alors de **puissance** (ou d'exposant) d'un mot w, notée w^n où $n \geq 0$, définie par :

1. $w^0 = \varepsilon$,
2. $w^{n+1} = w \cdot w^n$.

Par exemple, le mot $w = abbc$ élevé à la puissance 3 vaut $w^3 = abbcabbcabbc$. Si un mot w peut s'écrire comme la concaténation de deux mots $u \cdot v$, alors u est un **préfixe** de w et v est un **suffixe** de w. Plus généralement, si x peut s'écrire $u \cdot v \cdot w$, alors v est un **facteur** (ou sous-chaîne) de x. Les préfixes et les suffixes sont des cas particuliers de facteurs (en posant $u = \varepsilon$ ou $v = \varepsilon$). De même pour le mot lui-même.

Enfin, l'**inverse** d'un mot $w = \mathtt{a_1 a_2 \ldots a_n}$ est le mot $w^R = \mathtt{a_n \ldots a_2 a_1}$. Dans le cas particulier où $w = w^R$, le mot w est appelé un **palindrome**. Par exemple les mots `radar` ou `esoperesteicietserepose`.

1.2 Langage

Un **langage** est un ensemble de mots. Par exemple,

- $L_1 = \{\mathtt{aab}, \mathtt{aba}, \mathtt{abb}, \mathtt{baa}, \mathtt{bab}, \mathtt{bba}\}$ sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.
  Ce langage consiste en tous les mots de trois lettres composés de `a` et de `b` ayant au moins un `a` et un `b`. C'est un langage fini car le nombre de mot qu'il contient est fini.
- $L_2 = \{\mathtt{acbb}, \mathtt{accbb}, \mathtt{acccbb}, \ldots\}$ sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.
  Ce langage consiste en tous les mots commencant par `a`, suivi d'un ou plusieurs `c` et se terminant par `bb`. C'est un langage infini.

La **taille d'un langage** L, également notée $|L|$ est le nombre de mots qu'il contient. Par exemple ci-dessus $|L_1| = 6$ et $|L_2| = \infty$. Le **langage vide** $L = \{\}$ est noté $\emptyset$. Attention à ne pas confondre le mot vide et le langage vide. Par exemple le langage $L = \{\varepsilon\}$ n'est pas vide : il contient un mot (le mot vide), sa taille est donc 1.

Étant donné un alphabet Σ, on note Σ^* l'ensemble (et donc, le langage) de tous les mots définis sur cet alphabet, quelle que soit leur taille. Par exemple, pour $\Sigma = \{a, b\}$, on a :

$\Sigma^* = \{\varepsilon, \mathtt{a}, \mathtt{b}, \mathtt{aa}, \mathtt{ab}, \mathtt{ba}, \mathtt{bb}, \mathtt{aaa}, \mathtt{aab}, \mathtt{aba}, \mathtt{abb}, \mathtt{baa}, \mathtt{bab}, \ldots\}$

On note aussi Σ^+ le même langage privé de ε. Observons que Σ^* et Σ^+ sont des langages infinis (du moment que l'alphabet Σ contient au moins une lettre).

Les langages étant des ensembles, on peut leur appliquer les opérations ensemblistes classiques. On note donc $L_1 \cup L_2$ l'**union** de deux langages, et $L_1 \cap L_2$ leur **intersection**. Enfin, étant donné un langage L sur l'alphabet Σ, on note $\overline{L}$ le **complément** de ce langage, c'est à dire l'ensemble des mots sur Σ qui n'en font pas partie. Autrement dit, $\overline{L} = \Sigma^* \setminus L$.

Il existe aussi des opérations plus spécifiques sur les langages. Soient L_1 et L_2 deux langages, l'opération de **concaténation** est définie comme suit :

$$L_1 \circ L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1 \text{ et } w_2 \in L_2\}$$

Autrement dit, $L_1 \circ L_2$ est l'ensemble des mots que l'on peut obtenir en concaténant un mot de L_1 avec un mot de L_2. De même que pour les mots, on peut concaténer un langage plusieurs fois avec lui-même, on parle alors de **puissance** d'un langage, noté L^n.

Voici quelques exemples pour $L_1 = \{\varepsilon, \mathtt{ab}\}$ et $L_2 = \{\mathtt{c}, \mathtt{bc}, \mathtt{abc}\}$ sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$:

- $L_1 \cup L_2 = \{\varepsilon, c, ab, bc, abc\}$

- $L_1 \cap L_2 = \emptyset$
- $L_1 \circ L_2 = \{c, bc, abc, abbc, ababc\}$
- $L_1^3 = \{\varepsilon, ab, abab, ababab\}$
- $L_2^2 = \{cc, cbc, cabc, bcc, bcbc, bcabc, abcc, abcbc, abcabc\}$

Observons que dans l'exemple de concaténation, le mot `abc` peut être obtenu de deux manières différentes : par $\texttt{ab} \cdot \texttt{c}$ (avec $\texttt{ab} \in L_1, \texttt{c} \in L_2$) ou par $\varepsilon \cdot \texttt{abc}$ (avec $\varepsilon \in L_1, \texttt{abc} \in L_2$). Par ailleurs, attention à ne pas confondre la concaténation de mots $(\cdot)$ et la concaténation de langages $(\circ)$. Même remarque pour le produit.

De même que ε est l'élément neutre pour la concaténation de *mots*, le langage $\{\varepsilon\}$, noté L_ε, est l'élément neutre pour la concaténation de *langages*. En effet, pour tout langage L, on a bien $L_\varepsilon \circ L = L \circ L_\varepsilon = L$. Le langage vide $\emptyset$, quant à lui, n'est pas neutre, c'est un élément **absorbant** (comme le zéro de la multiplication) qui vérifie $L \circ \emptyset = \emptyset \circ L = \emptyset$ pour tout L.

En utilisant l'élément neutre, on peut définir plus rigoureusement la puissance d'un langage L comme :

1. $L^0 = L_\varepsilon = \{\varepsilon\}$,
2. $L^{n+1} = L^n \circ L$.

Enfin, L^* désigne l'ensemble des mots résultant d'une concaténation d'un nombre arbitraire de mots de L (appelé **fermeture itérative** de L), à savoir [1] :

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \ldots = \bigcup_{i \geq 0} L^i$$

et L^+ désigne les mots résultant d'une concaténation d'*au moins* un mot de L ; autrement dit, $L^+ = L \circ L^*$. Le mot vide appartient donc à L^*, qu'il soit ou non dans L, mais il n'appartient à L^+ que s'il appartient à L. On notera ici la signification intuitive des exposants $^+$ et *, qui comme pour Σ^+ et Σ^*, indique une répétition un nombre arbitraire de fois (mais au moins une pour $^+$).

1.2.1 Observations

D'un point de vu algébrique, on peut voir l'opération $\cup$ comme une addition (dont l'élément neutre est $\emptyset$) et l'opération $\circ$ comme une multiplication (dont l'élément neutre est $\{\varepsilon\}$ et l'élément absorbant est $\emptyset$). On a donc :

- $L \circ \emptyset = \emptyset$
- $L \circ \varepsilon = L$
- $L \cup \emptyset = L$

Par ailleurs, $L \cup \varepsilon$ revient à ajouter au langage L le mot ε (s'il n'y était pas déjà).

1. ATTENTION, en classe, j'avais d'abord écrit cela, puis autre chose, mais au final, c'était bien une union. Désolé pour la confusion, merci de corriger vos notes de cours (si applicable).

1.3 Formalismes de spécification des langages

Pour spécifier un langage, c'est-à-dire le décrire formellement, plusieurs formalismes sont à disposition. La première solution consiste à énumérer de manière exhaustive les mots qu'il contient, ce qui est souvent inadapté. Pour décrire des langages infinis il faut alors utiliser des formalismes plus riches comme les *automates*, les *expressions régulières*, les *grammaires*, ou les *machines de Turing*, que nous découvrirons plus tard.