Chapitre 2

Introduction aux langages

Dans ce chapitre, nous allons définir ce que nous appellerons *langage*. On connaît déjà certains types de langages : les langages de programmation, ou le langage naturel (français, anglais). En réalité, nous allons définir des langages bien plus généraux (et bien plus simples) que les langages de programmation.

Pour définir un langage, il existe plusieurs choix. Un choix très général est de définir un langage par sa *grammaire*. Nous n'étudierons pas ce cas, et nous nous limiterons à une définition très générale des langages, pour ensuite se concentrer plus tard dans l'année par des langages très simples : les langages rationnels, et les langages reconnus par des automates finis *.

2.1 Définitions générales des langages

Définition 2.1.1
On appelle alphabet *tout ensemble fini. Les éléments d'un alphabet sont appelés* lettres.

EXEMPLE

On peut considérer l'alphabet binaire $\{0,1\}$, l'alphabet ASCII $\{A,\dots,Z,a,\dots,z,0,\dots,9,\dots\}$ ou l'alphabet de Dyck $\{(,)\}$.

Maintenant, fixons un alphabet Σ.

Définition 2.1.2
Pour tout $p \geqslant 0$, on appellera mot sur Σ de longueur p *toute suite de p lettres de Σ.*

On appellera mot sur Σ *tout mot de longueur p pour tout $p \geqslant 0$.*

La longueur d'un mot p sera notée $|p|$, et l'unique mot de longueur 0 sera appelé le mot vide, *noté ε.*

*. Spoiler alert : ce sont en fait les mêmes langages !

L'ensemble de tous les mots sur l'alphabet Σ sera noté Σ^*.

NOTA

En abusant un peu, on dira que $\Sigma \subseteq \Sigma^*$, en confondant les lettres et les mots de une lettre.
On notera alors les mots omme juxtaposition de lettres, en omettant les parenthèses et les virgules. On fera en particulier attention si certains symboles sont communs entre les lettres (en réalité, on essayera d'éviter autant que possible ce genre d'alphabets).

EXEMPLE

Sur l'alphabet $\Sigma = \{a, b\}$, on notera aab ou $abbbaaaabbb$ plutôt que (a, a, b) ou $(a, b, b, b, a, a, a, a, b, b, b)$.
Sur l'alphabet $\{0, 00, 000\}$, on ne peut pas savoir si 000 est le mot (000), $(0, 00)$, $(0, 0, 0)$ ou $(00, 0)$. On utilisera donc de subtiles espaces : 0 00.

Venons-en finalement à la plus générale notion de langage.

Définition 2.1.3
Soit Σ un alphabet. On appelle langage sur Σ *toute partie de Σ^*.*

Il n'y a donc absolument aucune contrainte pour créer des langages.

EXEMPLE

Les langages les plus évidents sont Σ^* tout entier et $\emptyset$.
On peut citer d'autres exemples de langages sur $\{a, b\}$:

- Le langage des mots qui ont autant de a que de b
- Le langage des mots qui ont un nombre premier de a
- Le langage des palindromes
- Le langage des mots prononçables.

Définition 2.1.4
Soient $u = u_0 u_1 \cdots u_{m-1}$ et $v = v_0 v_1 \cdots v_{p-1}$ deux mots sur un alphabet Σ. On appelle concaténation *de u et v le mot, noté $u \cdot v$ ou uv*

$$uv = u_0 u_1 \cdots u_{m-1} v_0 v_1 \cdots v_{n-1}.$$

Attention, la concaténation de mots est le plus souvent non-commutative.

EXERCICE

À quelle condition nécessaire et suffisante la concaténation est-elle commutative ?

On dit que l'ensemble $(\Sigma^*, \cdots)$ est un monoïde (ensemble muni d'une loi de composition interne associative avec un élément neutre).

Proposition 2.1.5
Pour tous $u, v \in \Sigma^*$, on a

$$|uv| = |u| + |v|.$$

Démonstration. En exercice. □

Définition 2.1.6
Soient u et v deux mots sur Σ. On dit que v est un facteur *de u s'il existe $u_1, u_2 \in \Sigma^*$ tels que $u = u_1 v u_2$.*

Si $u_1 = \varepsilon$, alors on dit que v est un préfixe de u. Si $u_2 = \varepsilon$ convient, alors on dit que v est un suffixe de u.

On notera $v \sqsubseteq u$ pour noter "v est un préfixe de u".

EXEMPLE

Considérons le mot u="hippopotomonstrosesquippedaliophobie". Alors "hippopo" est un préfixe de u, "sesqui" est un facteur de u et "phobie" est un suffixe de u.

NOTA

Attention à la notation $u \sqsupseteq v$, qui pourrait vouloir dire que v est un préfixe de u, ou que u est un suffixe de v, ce qui est complètement différent.

Proposition 2.1.7
$\sqsubseteq$ est une relation d'ordre (partiel si $|\Sigma| \geqslant 2$) sur Σ^*.

Démonstration.
- u est bien un préfixe de u, car $u = \varepsilon u$.
- si $u \sqsubseteq v$ et $v \sqsubseteq w$, alors on peut écrire $v = uv_1$ et $w = vw_1$. On a donc $w = uv_1w_1$, et donc $u \sqsubseteq w$.
- si $u \sqsubseteq v$ et $v \sqsubseteq u$, alors $u = vu_1$ et $v = uv_1$. On a alors $u = uv_1u_1$, et donc $u_1 = v_1 = \varepsilon$. Donc $u = v$.

Si Σ a au moins deux lettres a et b, alors ni ab ni ba n'est préfixe de l'autre. □

2.2 Quelques opérations sur les langages

Avec des langages, on peut en construire des nouveaux en ayant recours à quelques opérations.

Proposition 2.2.1
Soient L_1 et L_2 deux langages sur Σ. Alors $L_1 \cup L_2$ (le plus souvent notée $L_1 + L_2$), $L_1 \cap L_2$, $\Sigma^* \setminus L_1$ sont des langages sur Σ.

Ce sont les opérations ensemblistes classiques, et nous en reparlerons dans les chaptires ultérieurs.

Définition 2.2.2
Soient L_1 et L_2 deux langages sur Σ. On appelle concaténation *de L_1 et L_2 le langage, noté L_1L_2, formé des concaténations de mots de L_1 suivi de mots de L_2 :*

$$L_1L_2 = \{uv \mid u \in L_1,\ v \in L_2\}.$$

EXEMPLE

Sur $\Sigma = \{a, b\}$, soit L_1 le langage des mots ne contenant que des a, et L_2 le langage des mots ne contenant que des b. Alors

$$L_1 = \{a^n \mid n \in \mathbb{N}\},\ L_2 = \{b^n \mid n \in \mathbb{N}\} \text{ et } L_1L_2 = \{a^nb^m \mid n \in \mathbb{N},\ m \in \mathbb{N}\}.$$

Définition 2.2.3
Pour un langage L, on notera $L^0 = \{\varepsilon\}$ et $L^{n+1} = LL^n$.

Attention, les langages L^2 et $\{uu \mid u \in L\}$ sont en général différents.

EXERCICE

Reprenons L_1 de l'exemple précédent. Montrer que pour tout $n \in \mathbb{N}^*$, $L_1^n = L_1$.

Proposition-Définition 2.2.4
Soit L un langage sur Σ. Alors $L^* \sum_{n \geqslant 0} L^n$ est le plus petit langage sur Σ contenant L, le mot vide, et stable par concaténation.

L^*, prononcé "L étoile", s'appelle étoile de L *ou* fermeture de Kleene † de L.

†. Stephen Cole Kleene, 1909 – 1994, Américain

Démonstration. Il est assez clair que L^* satisfait aux conditions. Montrons que c'est le plus petit ; soit donc M un autre langage qui répond aux contraintes. Soit $u \in Ł^*$: il existe $n \in \mathbb{N}$ tel que $u \in L^n$. Mais alors il existe $u_1, \ldots, u_n \in L$ tels que $u = u_1 \cdots u_n$.

D'après les hypothèses, chaque u_i est dans M, et donc leur concaténation aussi, et donc $u \in M$.

D'où $L^* \subseteq M$. □

EXEMPLE

Si on considère $L = \Sigma$, alors $L^* = \Sigma^*$.

EXERCICE

Soit L le langage sur $\Sigma = \{0, 1\}$ des représentations binaires de nombres divisibles par 3. Montrer que

$$L^* = L \cup \{\varepsilon\}.$$

Définition 2.2.5
On appelle étoile stricte *d'un langage L le langage $L^+ = \sum_{n \in \mathbb{N}^*} L^n$. C'est le plus petit langage contenant L et stable par concaténation.*

EXERCICE

Montrer que $L^* = L^+$ si et seulement si $\varepsilon \in L$.

Proposition 2.2.6
On a en fait $L^+ = LL^*$.

2.3 Exercices

Arbres binaires

Exercice 1. (i) Montrer le lemme de Lévi :

> **Lemme 2.3.1**
> *Soient x, y, u, v quatre mots sur un alphabet Σ tels que $uv = xy$. Alors il existe $t \in \Sigma^*$ tel que*
> $$(u = xt \text{ et } y = tv) \text{ ou } (x = ut \text{ et } v = ty).$$

(ii) En déduire le résultat suivant :
Soient $u, v, w \in \Sigma^*$ tels que $u \sqsubseteq w$ et $v \sqsubseteq w$. Montrer que $u \sqsubseteq v$ ou $v \sqsubseteq u$.

Exercice 2. On appelle *langage de Dyck* le langage sur $\Sigma = \{(,)\}$ des mots bien parenthesés. C'est le plus petit langage contenant ε, stable par concaténation vérifiant

$$\forall u \in D,\ (u) \in D.$$

On définit la valuation d'un mot de Dyck en définissant $\sigma[(] = 1$ et $\sigma[)] = -1$, et en prolongeant à l'ensemble des mots par somme.

(i) Donner les valuations de $(()())$, $()()($ et $)(()$. Lesquels sont des mots de Dyck ?

(ii) Montrer que la valuation de tout mot de Dyck est nulle.

(iii) Montrer que la valuation de tout préfixe d'un mot de Dyck est positive.

(iv) Par récurrence sur $|u|$, montrer que tout mot sur Σ^* tel que $\sigma[u] = 0$ et $\forall v \sqsubseteq u,\ \sigma[v] \geqslant 0$ est un mot de Dyck.

Exercice 3. Reprenons les mots de Dyck de l'exercice précédent. Écrire une fonction OCaml qui vérifie si un mot donné est un mot de Dyck.

```
type parenthese = L | R;;
type mot = parenthese list;;
```

On peut aussi montrer qu'un mot est de Dyck si et seulement si en supprimant toutes les paires () récursivement, on arrive au mot vide.

Écrire une nouvelle fonction qui teste si un mot est de Dyck.