Chapitre 0. Introduction

Définitions

- Alphabet
- Mot
- Langage

Définitions

Alphabet :

- Un alphabet (ou vocabulaire) est un ensemble fini non vide de symboles (ou caractères ou lettres). On le notera en général Σ , dans la suite du cours

Exemples :

Σ_1 = { •, ♣, ♦ }

Σ_2 = { a, b, c, …. , z }

Σ_3 = { if, then, else, id, nb, =, + }

Mot :

- Un mot, défini sur un alphabet $\sum$, est une suite (ou séquence) finie d'éléments de $\sum$

Exemples :

- sur l'alphabet $\sum_1$, les mots ♣, ♦♣, ♦♣•
- sur l'alphabet $\sum_2$, les mots programme, et, if, aba, sbl, cad, ple, …
- sur l'alphabet $\sum 3$, les mots if id = nb, idif, else, …

Remarque : les caractères a, b, c, …, z (en minuscule) seront utilisés pour designer un mot

Longueur d'un mot :

- La longueur d'un mot f défini sur un alphabet $\sum$, notée $|f|$, est le nombre de symboles qui composent f.

Par exemple :

- Sur $\sum_1$ le mot f = ♦♣•, |f| = 3
- Sur $\sum_2$ le mot f = if, |f| = 2
- Sur $\sum_3$ le mot f = if id = nb, |f| = 4

Mot vide :

- le mot vide, noté λ ou ε, est défini sur tous les alphabets et est le mot de longueur 0 (|ε| = 0).

Les ensembles $\sum^+$ et $\sum^*$:

- On note $\sum^+$ l'ensemble des mots de longueur supérieure ou égale à 1 que l'on peut construire à partir de l'alphabet $\sum$
- On note $\sum^*$ l'ensemble des mots que l'on peut construire à partir de $\sum$, y compris le mot vide :

$$\sum^* = \sum^+ \cup \{\varepsilon\}$$

Par exemple :

- Soit $\Sigma = \{a, b\}$ un alphabet,

➢ Σ^+ comprend :

- Longueur 1: a, b
- Longueur 2 2 : aa, ab, ba, bb
- Longueur 3 : aaa, aab, aba, abb, baa, bba, bab, bbb

………

➢ Σ^* comprend :

$$\Sigma^+ \cup \{\text{mot de longueur } 0 : \varepsilon\}$$

Opérations sur les mots

1°) Concaténation

- Soient deux mots f et g définis sur un alphabet ∑. La concaténation de f avec g, notée f.g (ou fg) est le mot formé en faisant suivre les symboles de f par les symboles de g.
- La puissance d'un mot f, notée f^n est la concaténation du mot f n fois. Cette puissance est calculée de manière itérative comme suit :

$$\begin{cases} f^0 = \varepsilon, \\ f^n = f.f^{n-1} \text{ pour } n \geq 1 \end{cases}$$

Exemple :

Sur l'alphabet ∑2 si u = ab et v = cd alors u.v = abcd, v.u = cdab et u^3 = ababab

Propriétés

- La concaténation est associative mais non commutative.
- La concaténation est une loi de composition interne de Σ
- Et ε est son élément neutre. Par conséquent, $(\Sigma^*, .)$ est un monoïde

2°) Préfixe, Suffixe et Facteur

- Soient deux mots f et g définis sur un alphabet Σ :

  - f est un préfixe (ou facteur gauche) de g si et seulement si $\exists$ $h \in \Sigma^*$ tel que $g = fh$
  - f est un suffixe (ou facteur droit) de g si et seulement si $\exists$ $h \in \Sigma^*$ tel que $g = hf$
  - f est un facteur de g si et seulement si $\exists$ h_1 et $h_2 \in \Sigma^*$ tel que $g = h_1 f h_2$

Remarque :

Si $f \neq g$ et $f \neq \varepsilon$ alors f est dit facteur propre de g

Le mot vide est préfixe, suffixe de tout mot

3°) Sous-mot

- Soient f et g deux mots $\in \Sigma^*$, f est un sous-mot de g s'il existe $h_1, h_2, \ldots, h_{n+1} \in \Sigma^*$ tels que :

$$f = f_1 f_2 \ldots\ldots\ldots f_n,\ \forall\ i\ 1 \leq i \leq n\ fi \in \Sigma \quad \text{et} \quad g = h_1 f_1\ h_2 f_2 \ldots\ldots\ldots f_n h_{n+1}$$

Remarque :

- Un facteur est un sous-mot mais l'inverse n'est pas vrai.

Exemple :

Soit $\Sigma = \{a, b, c\}$ un alphabet et $f \in \Sigma^*$ tel que $f = ababc$

$f = u \cdot x \cdot v \mid u \in$ préfixes(f), $x \in$ sous-chaînes(f) et $v \in$ suffixes(f)

- préfixes(f) = $\{ u \mid f = u \cdot v \} = \{\lambda, a, ab, aba, abab, ababc \}$
- suffixes(f) = $\{ v \mid f = u \cdot v \} = \{\lambda, c, bc, abc, babc, ababc \}$
- sous-mot(f) = $\{\lambda, a, b, c, ab, ba, bc, aba, bab, abc, abab, babc, ababc \}$

4°) Mots conjugués

Soient f et g deux mots de Σ^*, f et g sont conjugués s'il existe u et v deux mots $\in \Sigma^*$ tel que

$$f = uv \text{ et } g = vu$$

5°) Miroir d'un mot

Soit f un mot de Σ^*, l'image miroir de $f = a_1a_2\ldots a_n$ où $a_i \in \Sigma\ \forall\ i,\ 1 \leq i \leq n$ est définie par

$$f^R = a_na_{n-1}\ldots a_1$$

Un mot est un palindrome s'il est égal à son miroir

Exemple :

$\Sigma = \{a, b\}$, $f = aba$ et $f^R = aba$

Langage

- Un langage, défini sur un alphabet $\sum$, est un ensemble de mots définis sur $\sum$. C'est donc un sous-ensemble de Σ^*

Deux langages particuliers sont indépendants de l'alphabet Σ :

- le langage vide ($L = \phi$),

- le langage contenant le seul mot vide ($L = \{\varepsilon\}$)

Opérations ensemblistes sur les langages :

Soient 2 langages L1 sur $\Sigma 1$ et L2 sur $\Sigma 2$:

- L'union de L1 et de L2, notée $L1 \cup L2$, est le langage défini sur $\Sigma 1 \cup \Sigma 2$ par l'ensemble de tous les mots contenus soit dans L1 ou dans L2

$$L1 \cup L2 = \{f \,/\, f \in L1 \text{ ou } f \in L2\}$$

- L'intersection de L1 et de L2 est le langage défini sur $\Sigma 1 \cap \Sigma 2$ par

$$L1 \cap L2 = \{f \,/\, f \in L1 \text{ et } f \in L2\}$$

- Le complémentaire de L1 est le langage défini sur $\Sigma 1$ par :

$$C(L1) = \{f \in \Sigma 1^* \text{ et } f \notin L1\}$$

- La différence de L1 et de L2 est le langage défini sur $\Sigma 1$ par :

$$L1 - L2 = \{f \,/\, f \in L1 \text{ et } f \notin L2\}$$

Opérations définies sur les langages :

Soient 2 langages L1 sur $\sum 1$ et L2 sur $\sum 2$:

- Le produit ou concaténation de L1 et de L2, est le langage défini sur $\sum 1 \cup \sum 2$ par :

$$L1.L2 = \{fg \ / \ f \in L1 \text{ et } g \in L2\}$$

Remarque :

Le produit de langages est associatif mais non commutatif

- La puissance d'un langage L est définie de manière récursive par :

$$L^0 = \{\varepsilon\}$$

$$L^n = L.L^{n-1} \text{ pour } n \geq 1$$

- La fermeture itérative (ou de Kleene) d'un langage L, notée L* est définie par :

$$L^* = \cup_{i=0}^{\infty} L^i$$

De même, on définit L^+ par :

$$L^+ = \cup_{i=1}^{\infty} L^i$$

Exemple :

$\Sigma = \{a, b\}$, $L = \{a.b\}$, Calculer L^*

Remarque :

- Un langage fini peut être décrit par l'énumération des mots qui le composent.
- Certains langages infinis peuvent être décrits par l'application d'opérations sur des langages plus simples.
- Certains langages infinis peuvent être décrits par un ensemble de règles appelé grammaire.

Système de génération des langages : Les Grammaires

Définition :

Une grammaire formelle est un quadruplet $G = (N, \sum, P, S)$ où :

- N est l'alphabet non terminal ou ensemble des symboles non-terminaux. Ces symboles sont utilisés dans le processus de génération mais n'aparaissent pas dans les mots générés,
- $\sum$ est l'alphabet terminal, l'alphabet sur lequel est défini le langage
- $S \in N$, est le symbole de départ ou axiome
- P est l'ensemble des règles de production de la forme :

$$\alpha \rightarrow \beta \text{ avec } \alpha \in (N \cup \sum)^+$$

$$\text{et } \beta \in (N \cup \sum)^*$$

Langage généré par une grammaire :

Soit $G = (N, \sum, P, S)$ une grammaire, le langage généré par G, noté L(G) est défini par :

$$L(G) = \{f \in \sum^* / \text{ S dérive f après plusieurs étapes}\}$$

Remarque :

- Une grammaire définit un seul langage
- Un même langage peut être généré par plusieurs grammaires différentes

<u>Exercice :</u>

1°) Soit $G = (N, \Sigma, P, S)$ avec $N = \{S\}$, $\Sigma = \{b, c\}$ et $P = \{S \rightarrow bS \mid cc\}$

Déterminer L(G)

2°) Construire une grammaire qui génère $L = \{ba^nb \ / \ n \geq 0\}$

Classification de Chomsky

En introduisant des restrictions sur la forme des règles de production, on obtient différentes classes de grammaire, ordonnées par inclusion. On distingue les 4 types suivants :

- Type 0 : pas de restriction sur les règles.
- Type 1 : grammaires dépendantes du contexte ou contextuelles. Les règles de P sont de la forme :

$\alpha A \beta \rightarrow \alpha \gamma \beta$ avec $A \in N$, α et $\beta \in (N \cup \Sigma)^*$ et $\gamma \in (N \cup \Sigma)^+$

Autrement dit que le symbole non-terminal A est remplacé par γ si on a les contextes α à gauche et β à droite.

- Type 2 : grammaires algébriques ou hors-contexte. Les règles de P sont de la forme :

$A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (N \cup \Sigma)^*$

- Type 3 : grammaires régulières
  - à droite. Les règles de P sont de la forme : $A \rightarrow fB \,/\, f$ avec $f \in \Sigma^*$ et A et $B \in N$
  - à gauche. Les règles de R sont de la forme : $A \rightarrow Bf \,/\, f$ avec $f \in \Sigma^*$ et A et $B \in N$

A chaque type de grammaire est associé un type de langage :

- les grammaires de type 3 génèrent les langages réguliers,
- les grammaires de type 2 génèrent les langages algébriques,
- les grammaires de type 1 génèrent les langages contextuels,
- les grammaires de type 0 permettent de générer tous les langages “décidables”, autrement dit, tous les langages qui peuvent être reconnus en un temps fini par une machine. Les langages qui ne peuvent pas être générés par une grammaire de type 0 sont dits “indécidables”.

Ces langages sont ordonnés par inclusion :

$$\mathbf{T3 \subset T2 \subset T1 \subset T0}$$

A chaque type de grammaire est associé un type d'automate qui permet de reconnaître les langages de sa classe :

- Les langages réguliers sont reconnus par des automates finis,
- Les langages algébriques sont reconnus par des automates à pile
- Les autres langages, décrits par des grammaires de type 1 ou 0, sont reconnus par des machines de Turing.