Théorie des langages

Alexis Nasr

Symboles et mots

- Les symboles sont des éléments indivisibles qui vont servir de briques de base pour construire des mots.
- Un alphabet est un ensemble fini de symboles. On désigne conventionnellement un alphabet par la lettre grecque Σ.
- Une suite de symboles, appartenant à un alphabet Σ, mis bout à bout est appelé un mot (ou une *chaîne*) sur Σ.
- On note $|m|$ la longueur du mot m (le nombre de symboles qui le composent).
- On note $|m|_s$ le nombre de symboles s que possède le mot m.
- Le mot de longueur zéro, appelé mot vide, est noté ε.
- Si m est un mot et $1 \leq i \leq |m|$, on note $m[i]$ le i^{eme} symbole de m.

Concaténation

- La concaténation de deux mots α et β, notée $\alpha \cdot \beta$ ou simplement $\alpha\beta$ est le mot obtenu en juxtaposant les symboles de β à la suite de ceux de α.
- La concaténation est associative : $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
- ε est l'élément neutre pour la concaténation : $\varepsilon\alpha = \alpha\varepsilon = \alpha$
- L'itération de l'opération de concaténation d'un mot m permet d'obtenir les puissances de m :

$$m^0 = \varepsilon \; \forall n \in \mathcal{N} \; m^{n+1} = mm^n$$

Facteurs

Etant donné trois mots α, β et γ définis sur un alphabet Σ,

- β est un facteur (ou sous-chaîne) de $\alpha\beta\gamma$.
- α est un facteur gauche (ou préfixe) du mot $\alpha\beta$
- β est un facteur droit (ou suffixe) du mot $\alpha\beta$
- Si $\alpha \neq \beta$ et α est un préfixe de β, alors on dit que α est un préfixe propre de β.
- Si $\alpha \neq \beta$ et α est un suffixe de β, alors on dit que α est un suffixe propre de β.
- ε est un préfixe, un suffixe et un facteur de tout mot.

Langages

- L'ensemble de tous les mots que l'on peut construire sur un alphabet Σ est noté Σ^*.
- Un langage sur un alphabet Σ est un ensemble de mots construits sur Σ.
- Tout langage défini sur Σ est donc une partie de Σ^*.
- L'ensemble de tous les langages que l'on peut définir sur Σ^* est l'ensemble des parties de Σ^*, noté $\mathcal{P}(\Sigma^*)$.

Opérations sur les langages

Union	$L_1 \cup L_2$	$\{x \mid x \in L_1 \text{ ou } x \in L_2\}$
Intersection	$L_1 \cap L_2$	$\{x \mid x \in L_1 \text{ et } x \in L_2\}$
Différence	$L_1 - L_2$	$\{x \mid x \in L_1 \text{ et } x \notin L_2\}$
Complément	$\bar{L}$	$\{x \in \Sigma^* \mid x \notin L\}$
Concaténation	$L_1 L_2$	$\{xy \mid x \in L_1 \text{ et } y \in L_2\}$
Auto concaténation	$\overbrace{L \ldots L}^{n}$	L^n
Fermeture de Kleene	L^*	$\bigcup_{k \geq 0} L^k$

Exemples de langages

$$
\begin{array}{ll}
\Sigma = \{a\} & L_1 = \{\varepsilon, a, aa, aaa, \ldots\} \\
\Sigma = \{a, b\} & L_2 = \{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\} \\
\Sigma = \{a, b\} & L_2 = \{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\} \\
\Sigma = \{a, b\} & L_3 = \{\varepsilon, aa, bb, aaaa, abba, baab, bbbb, \ldots\} \\
\Sigma = \{a, b, c\} & L_4 = \{\varepsilon, abc, aabbcc, aaabbbccc, \ldots\}
\end{array}
$$

Exemples de langages

- $\Sigma = \{0 \ldots 9, +, \times, /, -, (,)\}$
- Une expression arithmétique peut être représentée par un mot $m \in \Sigma^*$
- L est l'ensemble des expressions arithmétiques bien formées
  - $(1 + 3) \times 45 \in L$
  - $/234 \notin L$
  - $(1 + 4456 \notin L$

Exemples de langages

- $\Sigma = \{a, \ldots z, \vee, \wedge, \neg, \rightarrow, \leftrightarrow\}$
- Une formule de la logique des propositions peut être représentée par un mot $m \in \Sigma^*$
- L est l'ensemble des formules de la logique des propositions bien formées
  - $a \vee b \wedge c \in L$
  - $\vee b\ c \notin L$

Exemples de langages

- Σ est l'ensemble des identifiants, des constantes, des opérateurs et des mots clefs du langage C
- Un programme en langage C peut être représenté par un mot $m \in \Sigma^*$
- L est l'ensemble des programmes syntaxiquement corrects du langage C
  - `main(void){printf(‘‘hello world’’);}` $\in L$
  - `main(void){printf(‘‘hello world’’)}` $\notin L$

Exemples de langages

- Σ est l'ensemble des mots du français
- une phrase du français peut être représentée par un mot $m \in \Sigma^*$
- L est l'ensemble des phrases syntaxiquement correctes du français
  - *ceci est une phrase correcte* $\in L$
  - *ceci est une correcte phrase* $\notin L$
  - *d'incolores idées vertes dorment furieusement* $\in L$

Exemples de langages

- $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Soit une grille de sudoku avec a^i_j le chiffre situé en ligne i colonne j.
- Une telle grille peut être représentée par le mot $S = a^1_1 a^1_2 \dots a^2_1 a^2_2 \dots a^9_9 \in \Sigma^*$
- L est l'ensemble des grilles de sudoku valides

Exemples de langages

- $\Sigma = \mathcal{N} \cup \{(,), ', ,\}$
- Un graphe (S, A) peut être représenté par un mot $m \in \Sigma^*$
- L est l'ensemble des graphes connexes
  - $((1,2,3,4),((1,2),(2,3),(3,1),(1,4))) \in L$
  - $((1,2,3,4,5),((1,2),(2,3),(3,1),(4,5))) \notin L$

Généralisons

Problème de décision :
Entrée : $x \in S$
Question : x satisfait-il la propriété P ?

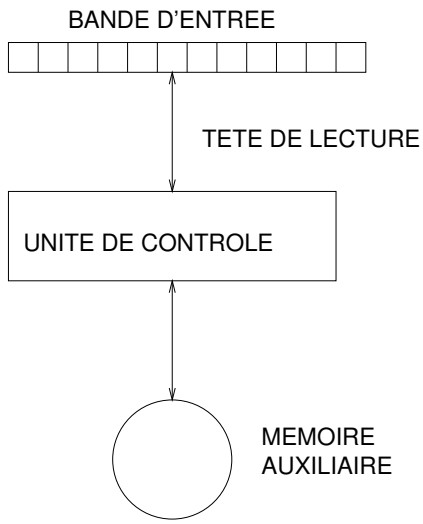
- S désigne un ensemble quelconque, dont les éléments sont appelés instances ou entrées du problème
- P est une propriété des éléments de S
- On représente $x \in S$ par un mot m
- On appelle L l'ensemble des mots qui représentent les instances qui vérifient P

Problème de décision :
Entrée : un mot m
Question : m appartient-il au langage L ?

Comment décrire un langage ?

- Enumération
  $L_2 = \{\varepsilon, aa, bb, ab, ba, aaaa, aaab, aaba, \ldots\}$
- Description littéraire
  Ensemble des mots construits sur l'alphabet $\{a, b\}$*, de longueur paire*
- Expression Régulière
  Formule permettant de dénoter de manière concise les mots du langage
  $((a + b)(a + b))*$
- Reconnaisseur
  Machine permettant de reconnaître tous les mots du langage
- Grammaire de réecriture
  Système permettant de générer tous les mots du langage

Représentation graphique d'un reconnaisseur



Eléments d'un reconnaisseur

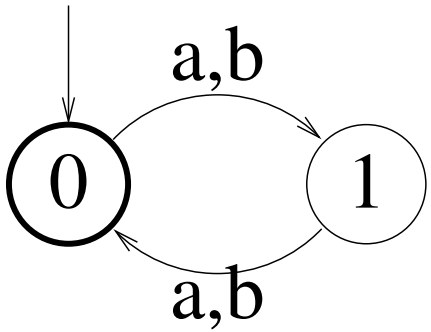
Un reconnaisseur est composé de quatre parties :

- 1 - une bande de lecture
  - elle est composée d'une succession de cases.
  - Chaque case pouvant contenir un seul symbole d'un alphabet d'entrée.
  - C'est dans les cases de cette bande de lecture qu'est écrit le mot à reconnaître.
- 2 - une tête de lecture
  - Elle peut lire une case à un instant donné.
  - La case sur laquelle se trouve la tête de lecture à un moment donné s'appelle la case courante.
  - La tête peut être déplacée par le reconnaisseur pour se positionner sur la case immédiatement à gauche ou à droite de la case courante.
- 3 - une mémoire
  - Elle peut prendre des formes différentes.
  - La mémoire permet de stocker des éléments d'un alphabet de mémoire.

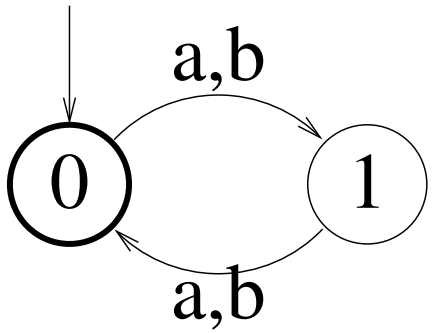
Eléments d'un reconnaisseur

- 4 - une unité de contrôle
  - Elle constitue le cœur d'un reconnaisseur.
  - Elle peut être vue comme un progamme qui dicte au reconnaisseur son comportement.
  - Elle est définie par un ensemble fini d'états ainsi que par une fonction de transition qui décrit le passage d'un état à un autre en fonction du contenu de la case courante de la bande de lecture et du contenu de la mémoire.
  - L'unité de contrôle décide aussi de la direction dans laquelle déplacer la tête de lecture et choisit quels symboles stocker dans la mémoire.
  - Parmi les états d'un reconnaisseur, on distingue
    - des états initiaux, qui sont les états dans lesquels doit se trouver le reconnaisseur avant de commencer à reconnaître un mot
    - des états d'acceptation qui sont les états dans lequel doit se trouver le reconnaisseur après avoir reconnu un mot.

Reconnaisseur simple



$ababaa \in L$?



$$
\begin{aligned}
& (0, ababaa) \\
\vdash \quad & (1, babaa) \\
\vdash \quad & (0, abaa) \\
\vdash \quad & (1, baa) \\
\vdash \quad & (0, aa) \\
\vdash \quad & (1, a) \\
\vdash \quad & (0, \varepsilon)
\end{aligned}
$$

Grammaires de réécriture

Une grammaire de réécriture est un 4-uplet $\langle N, \Sigma, P, S \rangle$ où :

- N est un ensemble de symboles non terminaux, appelé l'alphabet non terminal.
- Σ est un ensemble de symboles terminaux, appelé l'alphabet terminal, tel que N et Σ soient disjoints.
- P est un sous ensemble fini de :

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$$

  un élément (α, β) de P, que l'on note $\alpha \to \beta$ est appelé une règle de production ou règle de réécriture.
- S est un élément de N appelé l'axiome de la grammaire.
- Exemple :

$$G = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$$

$ababaa \in L$?

$$G = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$$

$$\begin{aligned} P &\Rightarrow aI \\ &\Rightarrow abP \\ &\Rightarrow abaI \\ &\Rightarrow ababP \\ &\Rightarrow ababaI \\ &\Rightarrow ababaaP \\ &\Rightarrow ababaa \end{aligned}$$

Reconnaissance

Etant donné un mot m et un langage L, on veut répondre à la question :

$$m \stackrel{?}{\in} L$$

La difficulté de répondre à cette question dépend-t-elle de la nature de L ?

- $L_0 = \{m \in \{a,b\}^* | \ |m| \ mod \ 2 = 0\}$
- $L_1 = \{m \in \{a,b\}^* | \ |m|_a = |m|_b\}$
- $L_2 = \{m \in \{a,b\}^* | \ m = uu\}$