Langages Formels

Anne Grazon – L3 Info Rennes 1

2015 – 2016, S1

1 Monoïdes libres, langages

1.1 Monoïdes

La théorie des langages est née dans les années 60 de la volonté des linguistes de formaliser la notion de grammaire (des langages naturels). Parmi eux, Noam Chomsky a défini quatre types de grammaires associées à quatre types de langages (type 0, 1, 2 et 3). Dans ce cours, nous étudierons les langages algébriques (type 2) et rationnels (type 3).

def 1 Quelques définitions :

1. Alphabet — Ensemble Σ fini non vide de symboles, appelés lettres.
2. Mot sur Σ — Suite finie de lettres. On définit sa longueur $|u| = n$.
3. Le mot vide est noté ϵ ou 1_Σ, $|1_\Sigma| = 0$.
4. Σ^* désigne l'ensemble de tous les mots sur Σ.
5. La loi de composition interne sur Σ^* notée $\cdot$ est la concaténation, qui est associative et admet 1_Σ comme élément neutre.
6. Et ça, c'est un monoïde.

def 2 Un monoïde est dit libre lorsque la décomposition d'un élément quelconque en “éléments de base" suivant sa loi de composition, est unique. Σ^* est alors le monoïde libre engendré par Σ.
Remarque — On voit immédiatement que deux mots sont égaux si et seulement si ils sont de même longueur, et ont leur lettres égales deux à deux. Cette propriété caractérise les monoïdes libres.

def 3 v est un facteur de $u \Leftrightarrow \exists \alpha, \beta \in \Sigma^*, u = \alpha v \beta$; c'est un facteur droit (resp. gauche) si $\beta = 1_\Sigma$ (resp. si $\alpha = 1_\Sigma$). C'est un facteur propre si $v \neq u$ et $v \neq 1_\Sigma$.

def 4 Pour $x \in \Sigma$, $|\cdot|_x$ est le nombre d'occurrences de x dans un mot ; on définit de même le nombre d'occurrences d'un mot dans un autre.

1.2 Langages

Un langage est un ensemble quelconque de mots ($L \subseteq \Sigma^*$, $L \in \mathcal{P}(\Sigma^*)$). L'union, l'intersection et le complémentaire sont définis intuitivement sur les langages.

def 5 Les autres opérations usuelles sont le produit $L \cdot M = \{uv | u \in L \text{ et } v \in M\}$, l'étoile de Kleene $L^* = \cup_{n \geqslant 0} L^n$ et l'étoile propre $L^+ = L^* \setminus L^0$.

2 Grammaires algébriques

def 6 Une grammaire est un quadruplet $G = (\Sigma, V, S, P)$ où Σ est l'alphabet terminal, V disjoint de Σ l'alphabet des non-terminaux, $S \in V$ l'axiome de G et $P \subsetneq V \times (X \cup V)^*$ l'ensemble des règles de production.

Exemple

$G_1 = (\Sigma, V, S, P)$ avec $\Sigma = \{a, b\}$, $V = \{S, T\}$ et les règles de production $P \to aSb + aT$, $T \to b$.

def 7 Une grammaire est dite linéaire (resp. à droite et à gauche) si $P \subset V \times (\Sigma^* \times V \times \Sigma^* \cup \Sigma^*)$ (resp. $\Sigma^* \times V$ and so on).

Exemple

G_1 est linéaire.

La dérivation consiste à engendrer un mot à partir d'un autre en suivant une règle de production. Elle est notée $\to$, sa fermeture réflexive $\to^*$ et une dérivation à l'ordre n, $\to^n$.

def 8 Le langage engendré par une grammaire est $L(G) = \{f \in \Sigma^* | S \to^* f\}$ (le langage élargi accepte aussi V^*). Réciproquement, un langage est dit algébrique s'il existe une grammaire G telle que $L = L(G)$.
Remarque — Pour prouver qu'une grammaire engendre un langage, on doit donc vérifier deux inclusions.

Exemple

$L(G_1) = \{a^n b^n | n \geqslant 1\}$

La famille des langages algébriques sur un alphabet Σ est notée $Alg(\Sigma^*)$.

lem 1 Lemme fondamental : soit G une grammaire, $f \in (\Sigma \cup V)^*$. Si f se factorise en $f_0 S_1 f_1 ... S_k f_k$ où $f_i \in \Sigma^*$ et $S_i \in V$, alors pour tout $g \in (\Sigma \cup V)^*$,

$$f \to^* g \Leftrightarrow g = f_0 h_1 f_1 ... h_k f_k \text{ et } \forall i S_i \to^* h_i$$

Plus précisément, $f \to^n g$ si idem et $\forall i S_i \to^{n_i} h_i$ avec $\sum n_i = n$.

prop 1 Principe de récurrence : soit $S \subset \mathbb{N}$ telle que $0 \in S$ et $\forall n, n \in S \Rightarrow n+1 \in S$. Alors $S = \mathbb{N}$.
Remarque — En appliquant ce principe à une propriété $\mathcal{P}(n)$ pour n entier, on peut démontrer des trucs. Il existe aussi la version dite forte de la récurrence.

def 9 A est un arbre de dérivation pour une grammaire G si les étiquettes de A sont dans $\Sigma \cup V \cup \{\epsilon\}$, les ϵ n'ont pas de frères et les nœuds E de fils $e_1, ..., e_n$ sont tels que $E \to e_1, ..., e_n$ est une règle de production de G.
Remarque — Les nœuds internes sont donc nécessairement étiquetés dans V.

def 10 Une grammaire est ambiguë si elle génère des mots qui possèdent plusieurs arbres de dérivation distincts.

On peut choisir de dériver à gauche ou à droite pour gérer l'ambiguïté.

prop 2 Si L et M sont algébriques, alors $L \cup M$ est algébrique. Étant données deux grammaires engendrant L et M, on en construit une qui engendre $L \cup M$ par union des règles de production.

Clôture des algébriques : Alg(X^*) est clos par union, produit et étoile mais pas par intersection ni complémentaire.

3 Automates finis

def 11 Un automate fini est un quintuplet $A = (\Sigma, Q, q_0, F, \delta)$ où Σ est l'alphabet d'entrée, Q l'ensemble des état, $q_0 \in Q$ l'état initial, $F \subseteq Q$ l'ensemble des états finals et $\delta \subseteq Q \times \Sigma \times Q$ l'ensemble des transitions.

def 12 Quelques définitions :
- Un chemin est une suite de transitions cohérentes ; sa trace est la suite de ses étiquettes $(x_1, ..., x_n) \in \Sigma^n$.
- Un mot est reconnu par un automate s'il est la trace d'un chemin menant de l'état initial vers un état final.
- Un langage est reconnaissable s'il existe un automate fini qui reconnaît tous ses mots.

Remarque — Les langages vide et Σ^* sont trivialement reconnaissables.

prop 3 La famille $\mathrm{Rec}(\Sigma^*)$ contient les parties finies de Σ^* et est close par union (trivial suivant les définitions).