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(Σ^*) 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), produit et étoile.

prop 4 Toute grammaire linéaire droite engendre un langage reconnaissable par un automate fini, et réciproquement.

def 13 Un automate A est dit déterministe si $\forall q \in Q, \forall x \in \Sigma, \#\{q' \in Q | (q, x, q') \in \delta\} \leqslant 1$ et complet si $\geqslant 1$.
Remarque — Pour un automate déterministe complet, δ est une fonction totale de $Q \times \Sigma \to Q$ qui s'étend récursivement et intuitivement en $\hat{\delta} : Q \times \Sigma^* \to Q$.
Remarque — Complétion d'automate : tout automate fini est équivalent à un automate complet qui peut se construire en ajoutant un état "puits" vers lequel on redirige toutes les transitions manquantes.

Clôture des reconnaissables : $\mathrm{Rec}(\Sigma^*)$ est clos par étoile (pour un automate donné, on renvoie vers l'état initial les transitions pointant vers un état final) et par intersection (pour deux automates, on construit l'automate des couples d'états, la fonction de transition étant inférée).

3.1 Lemme de l'étoile

lem 2 Soit L un langage reconnaissable ; $\exists N$ tel que, $\forall w \in L, |w| \geqslant N$, il existe une factorisation de $w = xyz$ telle que $|x| > 0$ et $\forall n, xy^n z \in L$.
Remarque — La preuve se construit à partir d'un automate, en posant $N = \#Q$.

prop 5 Le langage $\{a^n b^n | n \geqslant 0\}$ n'est pas reconnaissable.

3.2 Théorème de Kleene

def 14 $\mathrm{Rat}(\Sigma^*)$ est la plus petite famille de langages contenant les parties finies de Σ^*, close par union, intersection et étoile.

prop 6 Tout langage rationnel ($\in\mathrm{Rat}(\Sigma^*)$) possède une expression sous la forme d'un nombre fini d'unions, de produits et d'étoiles de langages finis. (On définit par suite les expressions rationnelles.)

prop 7 Théorème de Kleene — $\mathrm{Rat}(\Sigma^*) = \mathrm{Rec}(\Sigma^*)$
Remarque — Le sens réciproque s'obtient par exemple, par résolution de systèmes avec le lemme d'Arden ($X = AX \cup B \Rightarrow X = A^* B$).

3.3 Résiduels

Diapo 6, pp. 11-24.

prop 8 L'automate des résiduels de L est minimal pour L.

3.4 Nérode

def 15 Un mot de Σ^* sépare deux états q et q' d'un automate s'il mène à un état final par un, et non final par l'autre. Deux états sont Nérode-équivalents si aucun mot ne les sépare.

def 16 L'automate quotient est l'automate des classes de Nérode (en particulier, $\bar{\delta} = \{(\bar{q}, x, \bar{q'}) | \exists q \in \bar{q}, q' \in \bar{q'}$ tels que $(q, x, q') \in \delta\}$). Il est équivalent à l'automate initial et minimal (puisque $\#Res(L) = \#\{L_q | q \in Q\} = \#\{\bar{q} | q \in Q\}$).

On le construit par induction pour trouver l'équivalence de Nérode : $\sim_0$ sépare les états finals et non-finals, $\sim_i$ se déduit de $\sim_{i-1}$ en séparant ses classes selon les lettres de Σ.

4 Automates à pile

def 17 Un automate à pile simple est un quadruplet $(\Sigma, \Gamma, \gamma \in \Gamma, \lambda : \Sigma \times \Gamma \to \mathcal{P}(\Gamma^*))$. γ est le symbole de fond de pile.

def 18 Un mot est dit reconnu par un automate à pile simple, si la pile est vide après lecture de ce mot.

def 19 Une grammaire algébrique est sous forme normale de Greibach si $P \subset V \times XV^*$.

prop 9 Tout langage algébrique propre (ne contenant pas le mot vide) possède une grammaire sous forme normale de Greibach.

prop 10 Un langage propre est algébrique si, et seulement si il existe un automate à pile simple qui le reconnaît.
Remarque — On construit l'automate directement à partir de la grammaire, et réciproquement.

4.1 Automates à pile généraux

def 20 Même chose avec un ensemble fini d'états, dont un initial et éventuellement des finals.

def 21 Ces automates peuvent reconnaître un langage par pile vide ou par état final (i.e. un calcul valide mène à...).

prop 11 Un langage est algébrique si, et seulement si il existe un automate à pile qui le reconnaît par pile vide (ou par état final, c'est équivalent).
Remarque — Les automates à pile simple ne sont qu'un cas particulier (un seul état et pas d'ϵ-transition).

def 22 Un automate à pile est déterministe si tout calcul est entièrement déterminé par l'état, le sommet de pile et la lettre lue.
Remarque — Un langage est déterministe quand un automate déterministe le reconnaît par état final.

prop 12 Inclusions : Rec(Σ^*) $\subset$Det(Σ^*) $\subset$Non ambigus$\subset$Alg(Σ^*).

prop 13 L'ensemble des langages algébriques est clos par intersection rationnelle.

prop 14 Théorème de Bar-Hillel, Perles et Shamir : le lemme de l'étoile en mieux. Pour L algébrique, il existe N tel que tout mot $f \in L, |f| > N$ se décompose en $\alpha u \beta v \gamma$ avec $uv \neq \epsilon$, $|u\beta v| \leqslant N$ et $\forall n, \alpha u^n \beta v^n \gamma \in L$.