910 — Langages algébriques. Exemples et applications.

Introduction : les langages algébriques, appelés aussi langages hors-contexte, sont utiles pour l'analyse syntaxique des langages de programmation (ex : ocamlyacc).

I Grammaires algébriques.

1) Définitions

DEF : une grammaire algébrique est un triplet (A,V,P) où A,V alphabets finis et disjoints, et P partie finie de $V \times (A \cup V)^*$.
Les symboles de A sont les terminaux, ceux de V les variables. P contient des règles $X \to \omega$ (notation).

EX : $A=\{a,b\}$, $V=\{S\}$, $P=\{S \to aS + \varepsilon\}$, $G_{()}=(A,V,P)$.

REM : Par convention les lettres minuscules sont des terminaux, les majuscules des variables.

DEF : Soient $u,v \in (A+V)^*$. u se dérive en v, et on note $u \to v$, s'il existe $\alpha,\beta \in (A+V)^*$ et $X \in V$ tels que $v=\alpha X \beta$, $v=\alpha\omega\beta$ et $(X \to \omega) \in P$.

EX : $G_{()}$: $S \to aSb \to aaSbb \to aabb$.

REM : On note $\to^*$ la clôture réflexive et transitive de $\to$.

DEF : Si $u \in (A+V)^*$, $\widehat{L_G}(u) := \{v \in (A+V)^* \mid u \to^* v\}$
et $L_G(u) := \widehat{L_G}(u) \cap A^*$.
$L_G(S)$ est appelé langage engendré par G.

DEF : Un langage est dit algébrique s'il existe G grammaire algébrique et $S \in V$ tel que $L = L_G(S)$.

EX : $G_{()}$ est algébrique. $L_{G_{()}}(S)=\{a^n b^n \mid n \geq 0\}$.

EX : Langage de Dyck pour n parenthèses :

$$\begin{cases} A_n = \{a_1,\dots,a_n,\bar{a}_1,\dots,\bar{a}_n\} \\ S \to ST + \varepsilon \\ T \to a_1 S \bar{a}_1 + \dots + a_n S \bar{a}_n \end{cases}$$

Lemme fondamental : Soit $G=(A,V,P)$ une grammaire algébrique, et u,v deux mots de $(A+V)^*$. On suppose $u=u_1u_2$. Alors il existe $u \xrightarrow{k} v$ ssi $v=v_1v_2$ et il existe $u_1 \xrightarrow{k_1} v_1$, $u_2 \xrightarrow{k_2} v_2$ avec $k=k_1+k_2$.

2) Simplifications de grammaires

DEF : Une grammaire algébrique $G=(A,V,P)$ est réduite pour $S_0 \in V$ si :
- $\forall S \in V$, $L_G(S) \neq \emptyset$
- $\forall S \in V$, $\exists u,v \in (A+V)^*$, $S_0 \to^* uSv$.

PROP : Pour toute grammaire G, et $S_0 \in V$, il existe une grammaire G' réduite pour $S_0' \in V'$ telle que $L_G(S_0) = L_{G'}(S_0')$.

Complexité : $O(n^2)$ où n est la taille de la grammaire.

DEF : G est propre si elle ne contient aucune règle $S \to \varepsilon$ ou $S \to S'$ pour $S,S' \in V$.

EX : $S \to aSb + ab$

PROP : Pour toute grammaire G et $S \in V$, il existe une grammaire G' et $S' \in V'$ propre telle que
$L_{G'}(S') = L_G(S) \setminus \{\varepsilon\}$.

DEF : (Forme normale de Chomsky) C'est une grammaire telle que toutes les règles sont de la forme

$$\begin{cases} S \to S_1 S_2 \text{ où } S_1,S_2 \in V \\ \text{ou} \\ S \to a \text{ où } a \in A. \end{cases}$$

PROP: Pour toute grammaire G et $S \in V$, il existe une grammaire sous forme normale de Chomsky G', et $S' \in V'$ telle que $L_{G'}(S') = L_G(S) \setminus \{\varepsilon\}$.

Complexité : $O(n^2)$ où n est la taille de la grammaire.

Conséquence : algorithme pour savoir si un mot donné est engendré par une certaine grammaire G.

3) Systèmes d'équations

DEF : À $G = (A, V, P)$ grammaire où $V = \{X_1, \dots, X_m\}$ on associe le système $S(G)$: $L_i = \sum_{X_i \to w} w(L)$ $1 \leq i \leq m$ où $w(L)$ morphisme de substitution de $(A+V)^* \to P((A+V)^*)$ défini naturellement par induction.

DEF : on note $L_G = (L_G(X_1), \dots, L_G(X_m))$.

PROP : $\forall w \in (A+V)^*$, $L_G(w) = w(L_G)$.

PROP : L_G est la solution minimale de $S(G)$ (pour l'inclusion).

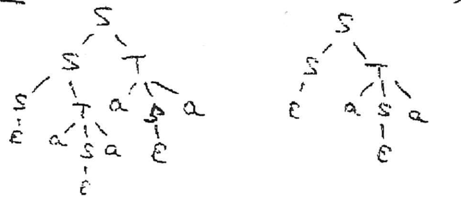
REM : On n'a pas unicité par contre : $X \to XX + \varepsilon$ a pour solutions tous les K^* où $K \subseteq A^*$. (on a unicité du minimal)

REM : systèmes d'équation : origine de la terminologie langage « algébrique ».

4) Arbres de dérivation

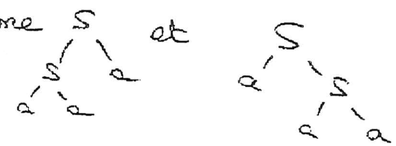
DEF : Soit $G = (A, V, P)$ une grammaire algébrique. Un arbre de dérivation est un arbre fini étiqueté par $A \cup V \cup \{\varepsilon\}$ tel que si S est l'étiquette d'un nœud interne et si $a_1, \dots, a_n$ sont les étiquettes de ses fils, alors $S \to a_1 \dots a_n$ est une règle de G. La frontière de l'arbre est le mot obtenu en concaténant les étiquettes des feuilles de gauche à droite.

EX : Pour $\{S \to ST + \varepsilon, T \to aS\}$



DEF : Une grammaire est ambiguë s'il existe un mot ayant deux arbres de dérivation distincts avec même racine.

EX : $S \to Sa + aS + aa$ donne



TH : (lemme d'itération, Bar-Hillel, Perles et Shamir).
Pour tout langage algébrique L, il existe $N \geq 0$ tel que pour tout mot $f \in L$, si $|f| \geq N$, alors on peut trouver une factorisation $f = \alpha u \beta v \gamma$ tel que $|uv| > 0$, $|u\beta v| < N$ et $\alpha u^n \beta v^n \gamma \in L$ pour tout $n \geq 0$. (DEV)

Application : $L = \{a^n b^n c^n \mid n \geq 0\}$ n'est pas algébrique.

COR : La classe des langages algébriques n'est close ni par intersection ni par complémentation.

II Propriétés des langages algébriques

1) Propriétés de clôture

PROP : Les langages algébriques sont clos par union, concaténation et étoile.

COR : Les langages rationnels sont algébriques.

DEF: une substitution $\sigma : A^* \to \mathcal{P}(B^*)$ où A, B sont deux alphabets, est *algébrique* si $\sigma(a)$ est algébrique pour tout $a \in A$.

PROP: Si $L \subseteq A^*$ est algébrique, et σ est une substitution algébrique $A^* \to \mathcal{P}(B^*)$, alors $\sigma(L) = \bigcup_{w \in L} \sigma(w)$ est algébrique.

PROP: Si L est algébrique, et K rationnel, alors $K \cap L$ est algébrique.

2) Problèmes de décidabilité

décision

PROP: Ces problèmes sont indécidables : (DEV)

a) pour deux grammaires G et G', est-ce que $L_G(S) \cap L_{G'}(S') = \emptyset$?

b) pour deux grammaires G et G', est-ce que $L_G(S) = L_{G'}(S')$?

c) pour une grammaire G, est-ce que $L_G(S) = A^*$

d) pour une grammaire G, est-ce que G est ambigüe?

rajouter des décidables ($L=\emptyset$)

III Automates à pile

1) Définitions

DEF: Un automate à pile est constitué :
- d'un alphabet d'entrée A
- d'un alphabet de pile Z et un symbole initial $z_0 \in Z$.
- d'un ensemble fini d'états Q dont un état initial $q_0 \in Q$
- de transitions $q, z \xrightarrow{y} q', h$ avec $q, q' \in Q$, $y \in A \cup \{\varepsilon\}$, $z \in Z$ et $h \in Z^*$

y est l'*étiquette* de la transition.

DEF: une *étape de calcul* est une paire de *configurations* $(C, C') \in (Q \times Z^*)^2$, notée $C \xrightarrow{y} C'$ telle que $C = (p, zw)$, $C' = (q, hw)$ et $p, z \xrightarrow{y} q, h$ est une transition de l'automate.

Un *calcul* est une suite d'étapes de calcul consécutives.

2) Lien avec les grammaires

TH: Un langage $L \subseteq A^*$ est algébrique ssi il existe un automate à pile qui accepte L.

REM: Il y a plus modes d'acceptation équivalents (par état final, pile vide, etc...).

3) Automates à pile déterministes

DEF: Un automate à pile (Q, A, Z, E, q_0, z_0) est *déterministe* si pour toute paire $(p, z) \in Q \times Z$:
- soit il existe une unique $p, z \xrightarrow{\varepsilon} q, h$ et pas de $p, z \xrightarrow{a} q, h$ pour $a \in A$
- soit il n'existe pas de $p, z \xrightarrow{\varepsilon} q, h$, et pour chaque $a \in A$, il existe au plus un $p, z \xrightarrow{a} q, h$.

EX: $Q = \{q_0, q_1, q_2\}$ avec
$q_0, z \xrightarrow{a} q_1, zz$
$q_1, z \xrightarrow{a} q_1, zzz$ est déterministe
$q_1, z \xrightarrow{b} q_2, \varepsilon$
$q_2, z \xrightarrow{b} q_2, \varepsilon$

REM: Les différents mode d'acceptation ne sont plus équivalents.

PROP: Le complémentaire d'un langage algébrique déterministe en est un aussi.

PROP: Tout langage algébrique déterministe est non ambigu.

→ certaines prop. deviennent décidables ($L = A^*$, $L = L'$)

Références:

Carton Langages formels
Dehornoy Mathématiques de l'informatique
Hopcroft Ullman (pour les complexités)