10/02 2015

310

Langages algébriques. Exemples.

1

Insuffisance des langages rationnels : $\{a^n b^n\}$ n'est pas reconnaissable par automate !

I Grammaires hors contextes et langages algébriques.

WOL p48
CAR p75

Déf 1 : Une grammaire hors contexte (ou algébrique) est un quadruplet (V, T, R, S) vérifiant :

- V, l'ensemble des variables, est fini (on notera ses éléments en majuscule)
- T, l'ensemble des terminaux, est fini (éléments notés en minuscule), avec $V \cap T = \emptyset$.
- $S \in V$ est le symbole de départ.
- R est un ensemble de règles : $R \subset V \times (V \cup T)^*$. On notera, pour $(X, u) \in R$, $X \to u$. X est le membre de gauche et le mot u sur l'alphabet $(V \cup T)$ est le membre de droite.

WOL p78

Exemple 2 : $T = \{a, b\}$, $V = \{S\}$, $R = \{S \to aSb,\ S \to \varepsilon\}$.

Exemple 3 : $T = \{a, b\}$, $V = \{S\}$, $R = \{S \to aSa,\ S \to bSb,\ S \to \varepsilon\}$.

CAR p76

Déf 4 : Soit $G = (V, T, R, S)$ une grammaire algébrique, u et v deux mots sur $V + T$. On dit que u se dérive en v, noté $u \to v$, lorsque : $\exists \alpha, \beta \in (A+V)^*$, $X \in V$ tels que :

$u = \alpha X \beta$, $v = \alpha w \beta$, et $X \to w \in R$.

Remarque : l'appelation grammaire hors contexte est licite !

WOL p48

Déf 5 : Un langage est dit algébrique s'il est généré par une grammaire algébrique, id est : $L = \{v \in T^* \ ; \ S \xrightarrow{*} v\}$, où $G = (V, T, R, S)$ est une grammaire algébrique.

Exemple 6 : Le langage algébrique généré par la grammaire de l'exemple 2 est $\{a^n b^n, n \geq 0\}$, et celui de l'exemple 3 est $\{w \bar{w}^R, w \in \{a,b\}^*\}$.

CAR p79

Lemme 7 (fondamental) : Soit $G = (V, T, R, S)$ une grammaire algébrique, u et v deux mots sur $(T+V)^*$. On suppose que u se factorise en $u = u_1 u_2$. Alors il existe une dérivation de longueur h $u \xrightarrow{h} v$ ssi il existe une factorisation $v = v_1 v_2$ telle que : $u_1 \xrightarrow{h_1} v_1$, $u_2 \xrightarrow{h_2} v_2$ et $h_1 + h_2 = h$.

CAR p90

Exemple 8 : G définie par $T = \{a, b\}$, S, $V = \{S, A\}$, $R = \{S \to SA + \varepsilon,\ A \to aSb\}$ (langage de Dyck).

AA se dérive $abaabb$ avec $A \to ab$ et $A \to aabb$.

CAR p90

Déf 9 : G une grammaire algébrique. Un arbre de dérivation est un arbre fini étiqueté par $V \cup T \cup \{\varepsilon\}$ vérifiant : si A est l'étiquette d'un noeud et $a_1, \dots, a_n$ les étiquettes de tous ses fils, alors $A \to a_1 \dots a_n \in R$. La frontière de l'arbre est la concaténation de toutes les feuilles, de gauche à droite.

Exemple 10 : cf annexe 1.

Déf 10 : Une grammaire G est dite ambiguë s'il existe au moins un mot ayant ayant au moins deux arbres de dérivation.

Exemple 11 : $V = \{S\}$ $T = \{a\}$ $R = \{S \to SS + a\}$. cf annexe 2.

CAR p80

Déf 12 : Une grammaire G est dite propre s'il n'existe aucune règle de la forme $A \to \varepsilon$ ou $A \to B$ avec $A, B \in V$.

Exemple 13 : la grammaire de l'exemple 2 n'est pas propre, mais la suivante l'est : $R = \{S \to aSb + ab\}$. Les langages reconnus sont les mêmes à ε près.

HU p92

Déf 14 Th : Tout langage algébrique ne générant pas ε peut être engendré par une grammaire sous forme normale de Chomsky, c'est-à-dire que toutes les productions sont de la forme $A \to BC$ ou $A \to a$.

Exemple 15 : $G = (\{S, A, B\}, \{a, b\}, A, S)$ avec les règles :

$S \to bA + aB$
$A \to bAA + aS + a$
$B \to aBB + bS + b$

forme normale de Chomsky :

$S \to B'A + AB'$
$A \to B'A'' + A'S + a$
$B \to A'B'' + B'S + b$
$B' \to b$, $A' \to a$,
$B'' \to BB$, $A'' \to AA$.

HU p95

Déf Th 16 : Tout langage algébrique ne générant pas ε peut être engendré par une grammaire sous forme normale de Greibach, id est toutes les productions sont de la forme : $A \to a\alpha$, où $\alpha \in V^*$.

CAR p77

Exemple 17 (langage de Dyck généralisé) :

$G = (\{S, A\}, \{a_1, \bar{a}_1, \dots, a_n, \bar{a}_n\}, R, S)$, avec R : $\begin{cases} S \to SA + \varepsilon \\ A \to a_1 S \bar{a}_1 + \dots + a_n S \bar{a}_n \end{cases}$

forme normale de Greibach : $\begin{cases} S \to SA + \varepsilon \\ A \to a_1 S X_1 + \dots + a_n S X_n \\ X_i \to \bar{a}_i \end{cases}$

II Propriétés générales des langages algébriques.

CAR p84 **Déf 18** : Système d'équation associé à une grammaire algébrique G :
Soit $n := |V|$, alors le système est constitué des n équations :
$$l_i = \sum_{X_i \to w} w(l) \text{, où } X_i \in V \text{, en les variables } l_1, \dots, l_n. \quad (\text{noté } S_G)$$
Propo 19 : Soit $G = (\{X_1, \dots, X_n\}, T, R, S)$. Alors le langage généré par G est la solution minimale du système défini précédemment. (*)
Exemple 20 : $\begin{cases} X_1 \to X_1 X_2 + \varepsilon \\ X_2 \to a X_2 + b \end{cases}$ donne le système $\begin{cases} l_1 = l_1 l_2 + \varepsilon \\ l_2 = a l_2 + b \end{cases}$

La solution minimale est : $\{\varepsilon\} \cup \{(a+b)^* b\}$
Caractérisation des langages algébriques (conditions nécessaires) :
WOL p88 **Lemme 21 (Ogden)** : Soit $G = (V, T, R, S)$ et L le langage généré par V.
Il existe $K > 0$, telle que pour tout mot $f \in L$ avec $|f| > K$, il existe une factorisation $f = u v x y z$ avec v ou $y \neq \varepsilon$, $|vxy| \leq K$ et $u v^n x y^n z \in L$ pour tout $n \geq 0$.
HU p130 - p135 **Exemple 22** : $L = \{a^i b^j c^k \mid i \neq j, j \neq k \text{ et } i \neq k\}$ n'est pas algébrique.
Th 23 (Bar-Hillel, Perl, Shamir) : Les langages algébriques sont clos par substitution, union, concaténation (morphisme).
Th 24 : Si L est algébrique et K est rationnel, alors $L \cap K$ est algébrique.
Th 25 : La classe des langages algébriques n'est pas close par intersection !
Exemple 26 : $L_1 = \{a^i b^i c^i, i \geq 1\}$ $\quad L_2 = \{a^i b^i c^j, i \geq 1, j \geq 1\}$ algébrique
$L_3 = \{a^i b^j c^j, i \geq 1, j \geq 1\}$ algébrique.

Le lemme d'Ogden montre que L_1 n'est pas algébrique, mais $L_1 = L_2 \cap L_3$!
CAR p88 **Déf 27** : Une grammaire étendue G est définie par (V, T, R, S) avec V et T deux alphabets finis disjoints, $R \subset V \times (T+V)^*$ telle que pour tout $X \in V$, $\{w \mid (X, w) \in R\}$ est rationnel.
Propo 28 : L'ensemble des mots engendrés par une grammaire étendue est algébrique.

Déf 29 : On note $\bar{L}$ l'image commutative d'un langage : $\bar{L} := \{\bar{w} \mid w \in L\}$, où $\bar{w}$ est la classe des anagrammes de w (on peut permuter les lettres) CAR p86
Th 30 (Parikh) : Tout langage algébrique L est commutativement équivalent à un langage rationnel R (ie $\bar{L} = \bar{R}$). (*) $\boxed{\text{DVT}}$

III Automates à pile.

Déf 31 : Un automate à pile est défini par un septuplet $M = (Q, \Sigma, \Gamma, \Delta, Z, s, F)$: WOL p73
Q est un ensemble fini d'états, Σ est l'alphabet d'entrée,
$Z \in \Gamma$ est le symbole initial de pile, Γ est l'alphabet de pile,
$s \in Q$ est l'état initial, $F \subseteq Q$ est l'ensemble des états accepteurs,
$\Delta \subset (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$ est la relation de transition.
Déf 32 : La configuration (q', w', α') est dérivable en une étape de la configuration (q, w, α) (noté $(q, w, \alpha) \vdash (q', w', \alpha')$) si :
$w = u w'$, $\alpha = \beta \delta$ et $\alpha' = \gamma \delta$ (sur la pile),
$((q, u, \beta), (q', \gamma)) \in \Delta$
Déf 33 : Une exécution d'un automate à pile sur un mot w est une suite de configurations (finie ou infinie) :
$$(s, w, Z) \vdash (q_1, w_1, \alpha_1) \vdash \dots \vdash (q_n, w_n, \alpha_n) \vdash \dots$$
Déf 34 : Un automate à pile acceptant sur état final, accepte le mot w si :
$$(s, w, Z) \vdash^* (p, \varepsilon, \gamma), \text{ avec } p \in F.$$
Déf 35 : Un automate à pile acceptant sur pile vide, accepte le mot w si : WOL p76
$$(s, w, Z) \vdash^* (q, \varepsilon, \varepsilon).$$
Déf 36 : Un langage L est reconnu par un automate à pile M si et seulement si l'ensemble des mots acceptés par M est L.
Propo 37 : Un langage est reconnu par un automate acceptant sur état final si et seulement il est reconnu par automate acceptant par pile vide. (cf annexe 3)
Exemple 38 : $Q = \{s, p, q\}$ $\quad \Sigma = \{a, b\}$ $\quad \Gamma = \{A\}$ $\quad F = \{q\}$ et
$$\Delta = \begin{cases} (s, a, \varepsilon) \to (s, A) \; ; \; (s, \varepsilon, Z) \to (q, \varepsilon) \; ; \; (s, b, A) \to (p, \varepsilon) \; ; \\ (p, b, A) \to (p, \varepsilon) \; ; \; (p, \varepsilon, Z) \to (q, \varepsilon) \end{cases} \text{ reconnaît } \{a^n b^n\}.$$

+ mots de Lukasiewicz

CAR p108

Th 39 : Un langage L est algébrique si et seulement s'il est reconnu par un automate à pile.

Déf 40 : Un automate à pile M est déterministe si pour toute paire $(q, \alpha) \in Q \times \Gamma$; soit il existe une unique transition $q, \alpha, \varepsilon \to q', \beta$ et il n'existe pas d'autre transition $q, \alpha, a \to q', \beta$, où $a \in \Sigma$; soit il n'existe pas de transition de la forme $q, \alpha, \varepsilon \to q', \beta$, et il existe au maximum un $a \in \Sigma$ tel que $q, \alpha, a \to q', \beta$.

AUT p127

Déf 41 : Un langage algébrique est déterministe s'il est reconnu par un automate à pile déterministe acceptant sur état final.

Propo 42 : Le complémentaire d'un langage algébrique déterministe est un langage algébrique déterministe.

Propo 43 : Si L est reconnu par un automate à pile déterministe acceptant sur pile vide, alors il est reconnu par un automate à pile déterministe acceptant sur état final.

AUT

⚠ La réciproque n'est pas vraie !

CAR p 115

Propo 44 : Tout langage algébrique déterministe est non ambigu.

WOL p 97

Exemple 45 (ADMIS) : $L_1 = \{w c \overline{w}^R \mid w \in \{a;b\}^*\}$ est déterministe, mais $L_2 = \{w \overline{w}^R \mid w \in \{a;b\}^*\}$ ne l'est pas.

IV Analyse syntaxique.

AUT p 133

Une partie du travail effectuée par le compilateur : vérifier que la syntaxe est correcte que le mot appartient au langage.

Analyse descendante : on part d'un mot et de S, et on modifie S (ou la partie gauche dans l'exemple) avec les règles pour supprimer au fur et à mesure les lettres du mot.

Analyse ascendante : on essaie de réduire le mot de droite sur l'exemple, en utilisant les membres droits d'une dérivation.

Analyse descendante LL

$$\begin{cases} S \to cT & (1) \\ T \to aTbc & (2) \\ T \to b & (3) \end{cases}$$

$(S, cabbc)$
$(Tc, cabbc)$ (1)
$(T, abbc)$
$(cbTa, abbc)$ (2)
(cbT, bbc)
(cbb, bbc) (3)
(cb, bc)
(c, c)
$(1, 1)$

Analyse ascendante LR

$$\begin{cases} E \to E+T & (1) \\ T \to (E) & (2) \\ E \to T & (3) \\ T \to a & (4) \\ T \to b & (5) \end{cases}$$

$[1, b+(a+a)]$
$[b, +(a+a)]$
$[T, +(a+a)]$ (5)
$[E, +(a+a)]$ (3)
$[E+, (a+a)]$
$[E+(, a+a)]$
$[E+(a, +a)]$
etc $[E+(T, +a)]$ (4)
$[E+(E+T,)]$
$[E+(E,)]$ (1)
$[E+(E), 1]$
$[E+T, 1]$ (2)
$[E, 1]$ (1)

Voici un algorithme qui teste l'appartenance d'un mot à un langage algébrique : l'algorithme Cocke-Younger-Kasami.

FB p320

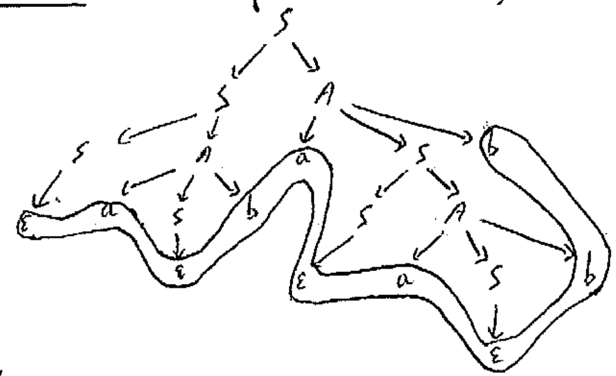
Th 46 : Il existe un algorithme qui teste l'appartenance à un langage algébrique en temps $O(|w|^3)$. $\boxed{DVT}$

(*) Propo 19 bis : Soit G une grammaire propre ; alors L_G est l'unique solution propre de S_G.

(**) Propo 29 bis : Soit G une grammaire ; alors toute solution L propre du système S_G vérifie $\overline{L} = \overline{L}_G$.

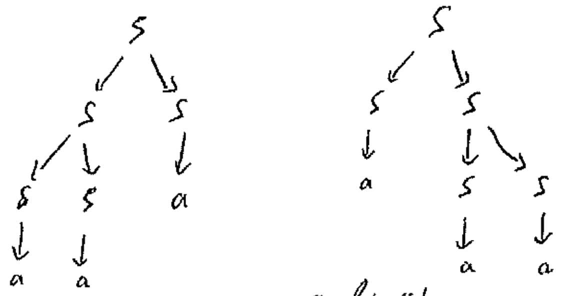
H-U : Hopcroft - Ullman : Introduction to automata theory...
CAR : Carton, Langages formels.
WOL : Wolper, introduction à la calculabilité.
AUT : Autebert ; Théorie des langages et des automates.
F.B : Floyd-Biegel, Le langage des machines.

Annexe 1 : avec la grammaire de l'exemple 8.



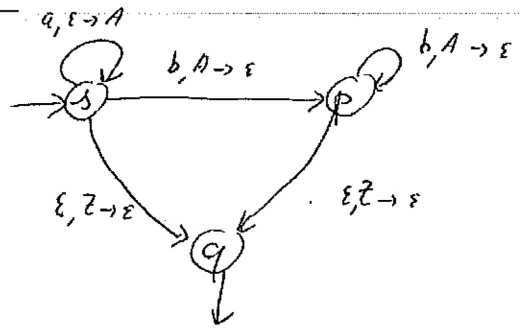
a b a a b b
est reconnu.

Annexe 2 :



ambiguë !

Annexe 3 :



Pour reconnaître aabba, on a sur la pile :

puis plus de transition possible.

Théorème de Parikh

Soit L un langage algébrique sur un alphabet A, alors il existe R rationnel sur A tel que $\overline{L}=\overline{R}$ (où $\overline{\cdot}$ désigne l'image commutative)

Preuve : par abus, on se permettra d'omettre le symbole de départ des grammaires manipulées

I) Lemme : soit G une grammaire propre, alors toute solution L de $\overline{S_G}$ vérifie $\overline{L}=\overline{L_G}$

II) Proposition : soit G une grammaire généralisée, alors le système $\overline{S_G}$ admet une solution rationnelle, i.e. si $G=(A,V,\bigcup_{i=0}^{n}\{X_i\to S_i(X)\})$, où $V=\{X_0,\dots,X_n\}$ et $X=(X_0,\dots,X_n)$, il existe un $n+1$-uplet $R=(R_0,\dots,R_n)$ de langages rationnels sur A tel que $\forall i\in[\![0,n]\!]$ $\overline{R_i}=\overline{S_i(R)}$

III) Conclusion

I) Il s'agit de la proposition 29 bis du plan. La preuve est essentiellement la même que celle de la proposition 19 bis (voir Carton p. 95)

II) On raisonne par récurrence sur le nombre de non-terminaux de G

Soit $\mathcal{P}(n)$: "Pour toute grammaire $G=(A,V,P)$ généralisée telle que $|V|=n$, $\overline{S_G}$ admet une solution rationnelle".

Montrons $\mathcal{P}(1)$: soit $G=(A,\{X\},\{X\to S(X)\})$ une grammaire généralisée.

Soit $T=S(X)\cap A^*$ et $P(X)=S(X)\setminus T=S(X)\cap((A+X)^*X(A+X)^*)$

On a donc $P(X)+T=S(X)$; $T\in Rat(A)$ et $P(X)\in Rat(A+X)$

Il existe $Q(X)\in Rat(A+X)$ tel que $\overline{P(X)}=\overline{Q(X)X}$ (par exemple, on peut choisir Q formé des mots de P privés de leur première occurrence de X)

Le système $\overline{S_G}$ se réécrit alors, pour l'inconnue L :

$\overline{L}=\overline{Q(L)L+T}$

Soit donc $R=Q(T)^*T$, $R\in Rat(A)$ (car $Q(T)$ est l'image de $Q(X)$ par un morphisme rationnel)

et $\overline{Q(R)R+T}=\overline{Q(Q(T)^*T)Q^*T+T}=\overline{(Q(T)Q(T)^*+\varepsilon)T}=\overline{Q(T)^*T}=\overline{R}$

donc R est solution de $\overline{S_G}$, d'où $\mathcal{P}(1)$

Soit $n \geqslant 1$, on suppose $\mathcal{P}(k)$ pour $k \leqslant n$

Soit $G = (A, \{X_0, \dots, X_n\} = V, S = \hat{\bigcup}_{i=0}^{n} \{X_i \to S_i(X)\})$ une grammaire généralisée à $n+1$ non-terminaux, où X désigne $(X_0, \dots, X_n)$.

On notera $V' = \{X_1, \dots, X_n\}$ et $X' = (X_1, \dots, X_n)$.

On définit $G_0 = (A+V', \{X_0\}, \{X_0 \to S_0(X_0, X')\})$.

Par hypothèse de récurrence, $\overline{S_{G_0}}$ admet une solution $R_0(X') \in Rat(A+V')$,

i.e. $\overline{S_0(R_0(X'), X')} = R_0(X')$ (*)

Soit maintenant $G' = (A, V', \hat{\bigcup}_{i=1}^{n} \{X_i \to S_i(R_0(X'), X')\})$

Par hypothèse de récurrence, $\overline{S_{G'}}$ admet une solution rationnelle $R' = (R_1, \dots, R_n)$

i.e. $\forall i \in [\![1, n]\!]\ R_i \in Rat(A)$ et $\overline{R_i} = \overline{S_i(R_0(R'), R')}$

De plus, (*) entraîne en particulier $\overline{S_0(R_0(R'), R')} = \overline{R_0(R')}$

et $R_0(R') \in Rat(A)$ en tant qu'image de $R_0(X') \in Rat(A+V')$ par un morphisme de $A+V'$ dans $Rat(A)$

donc $R = (R_0(R'), R')$ est solution rationnelle de $\overline{S_G}$, d'où $\mathcal{P}(n+1)$.

$\mathcal{P}(n)$ est vrai pour tout n, ce qui conclut la preuve de II)

III) ; Soit L un langage algébrique sur un alphabet A.

Soit G une grammaire propre reconnaissant $L \setminus \{\varepsilon\}$, $G = (A, \{X_0, \dots, X_n\}, P, X_0)$

Alors si $L_G = (L_0, \dots, L_n)$, on a $L \setminus \{\varepsilon\} = L_0$

Soit $R = (R_0, \dots, R_n)$ une solution rationnelle de $\overline{S_G}$. On a $\overline{R} = \overline{L_G}$.

donc en particulier $\overline{R_0} = \overline{L_0} = \overline{L \setminus \{\varepsilon\}}$

d'où $\overline{L} = \overline{R_0 + \varepsilon(L)}$, avec $R_0 + \varepsilon(L)$ rationnel sur A.