(9/0)

Intro Les langages rationnels ont de bonnes propriétés de clôture et de calculabilité, mais sont peu expressifs. Les machines de Turing sont très puissants mais peu pratique.

Langage algébrique : Bon compromis utile pour définir un langage informatique et le compiler.

II Grammaires algébriques

Chomsky : 1ère moitié : définir les types de grammaires (nouvelles / ancienne / moyenne)

1) Définitions

Def : Une grammaire algébrique est un triplet (A, V, P) avec A et V alphabets finis et disjoints et P une partie finie de $V \times (A \cup V)^*$. Les symboles de A sont dit terminaux, ceux de V sont appelés variable. P est un ensemble de règles $X \to \omega$.

Ex $A = \{a, b\}$ $V = \{S\}$ et $P = \{S \to aSb \ ; \ S \to \varepsilon\}$: $\{a^n b^n\}$

Rq Par convention, les terminaux sont en minuscule et les variable en majuscule

Def Soient $(u, v) \in (A \cup V)^*$, u se dérive en v ($u \to v$) si $\exists \alpha, \beta \in (A \cup V)^*$ et $X \in V$ tel que
$u = \alpha X \beta$ et $v = \alpha \omega \beta$ et $(X \to \omega) \in P$

Def On note $\to^*$ la fermeture transitive réflexive de $\to$

Def Soient $u \in (A \cup V)^*$ on note $\hat{L}_G(u) = \{v \ / \ u \to^* v\}$ et $L_G(u) = \hat{L}_G(u) \cap A^*$

Rq Pour une grammaire G on distingue souvent une variable particulière S_0 nommé axiome. Dans ce cas on appelle langage généré par G l'ensemble $L_G(S_0)$.

Def Un langage est dit algébrique ssi il est généré par une grammaire.

Ex Langage de Dyck pour n parenthèse :

$$\begin{cases} S \to ST & S \to \varepsilon \\ \forall k \leq n & T \to a_k S \overline{a_k} \end{cases}$$

Lemme fondamental : Soit $G = (A, V, P)$ une g.a. et u, v deux mots de $(A \cup V)^*$ avec $u = u_1 u_2$. Alors $u \to^k v$ (en k étapes) ssi $\exists (v_1, v_2)$ et (k_1, k_2) tq $\begin{cases} u_1 \to^{k_1} v_1 \\ u_2 \to^{k_2} v_2 \\ u = u_1 u_2 \ \ k = k_1 + k_2 \end{cases}$

2) Formes simplifiés.

Def Une grammaire algébrique est réduite par $S_0 \in V$ si :
- $\forall S \in V$, $L_G(S) \neq \emptyset$
- $\forall S \in V$, $\exists u, v \in (A \cup V)^*$, $S_0 \to^* u S v$

Prop Pour toute grammaire G et $S_0 \in V$ il existe une grammaire G' réduite par $S_0' \in V'$ telle que :
$L_G(S_0) = L_{G'}(S_0')$: $\mathcal{O}(n^2)$ avec n taille de la grammaire

Def G est propre si elle ne contient aucune règles de la forme : $S \to \varepsilon$ ou $S \to S'$ pour $S, S' \in V$

Def (Forme normale de Chomsky) : G' est une grammaire où les règles sont de la forme
$\begin{cases} S \to S_1 S_2 & \text{avec } S_1, S_2 \in V \\ S \to a & \text{avec } a \in A \end{cases}$ $\forall S \in V$

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

Bien.

Logique : enfin parenthèse → déc. (⇐)

Langages algébriques : CARTON (Aubert) : plan + dév

Remarque

9/10

910

3) Système d'équation

À une grammaire on peut associer un système d'équation sur les langages.

$$\begin{cases} X_1 \to X_1 X_2 + \varepsilon \\ X_2 \to a X_2 + b \end{cases} \rightsquigarrow \begin{cases} L_1 = L_1 L_2 + \varepsilon \\ L_2 = a L_2 + b \end{cases}$$

def on pose $\varepsilon(L) = \{\varepsilon\}$ $\quad a(L) = \{a\}$ pour $a \in A$ $\quad X_i(L) = L_i$

$xy(L) = x(L)\,y(L) \quad x, y \in (A \cup V)^*$

$K(L) = \bigcup_{w \in K} w(L)$ avec $K \subseteq (A+V)^*$

À une grammaire G on associe le système $S(G)$ formé par les équations $L_i = \sum_{X_i \to w} w(L)$ pour chaque variable L_i

Def On note $L_G = (L_G(X_1) \dots L_G(X_n))$

Prop $\forall w \in (A+w)^* \quad L_G(w) = w(L_G)$

Prop L_G est la solution minimal de $S(G)$

à déf (en quel sens inclusion ms pr un vect de lg ?)

Prop Soit G propre, L_G est l'unique solution propre de $S(G)$

Def Soit w un mot, on note $\overline{w}$ l'ensemble des anagrammes.
Soit L un langage on note $\overline{L} = \bigcup_{w \in L} \overline{w}$

Th (Parikh) Soit L algébrique, il existe un langage rationnel R tel que $\overline{L} = \overline{R}$

Cor Sur un alphabet unaire tout langage algébrique est rationnel.

II] Arbre de dérivation

1) Définition

Def Soit G une grammaire, un arbre de dérivation est un arbre fini étiqueté par $A \cup V \cup \{\varepsilon\}$ vérifiant :
Si S est l'étiquette d'un noeud interne avec comme fils $a_1 \dots a_n$ alors $S \to a_1 \dots a_n$ est une règle de G.

Def La frontière d'un arbre est obtenue en concaténant les feuilles de gauche à droite

Ex $S \to SS + a$



Def une grammaire est ambigüe si il existe un mot ayant deux arbres de dérivation ayant la même racine.

Def un langage est ambigüe si toutes ces grammaires le sont

2) Lemme d'itération (généralisation du lemme de l'étoile)

Lemme (Ogden) Pour toute grammaire G et toute variable S
Il existe un entier K tel que pour tout mot f de $\overline{L_G}(S)$ ayant au moins K lettres distinguées, f se factorise en $\alpha u \beta v \gamma$ avec

1) $S \xrightarrow{*} \alpha T \gamma$ et $T \xrightarrow{*} uTv + \beta$
2) Soit α, u, β soit β, v, γ contiennent des lettres distinguées
3) $u \beta v$ contient moins de K lettres distinguées

3) Corollaires

récente

(a) $a^n b^n c^n$ n'est pas algébrique

(b) $a^n b^p c^n d^p$ non plus (parenthèses imbriquées)

(c) $a^n b^n c^p \cup a^n b^p c^p$ est ambigüe

(d) Les langages algébriques ne sont clos ni par intersection ni par complémentaire.

III] Propriétés des langages algébriques

1) propriété de clôture

Prop les langages algébriques sont clos par union, concaténation et étoile

Prop les langages ~~algébriques~~ rationnelles sont algébriques

(910)

Def Soit A et B deux alphabets et $\sigma : A^* \to \mathcal{P}(B^*)$ avec $\sigma(A)$ algébrique $\forall a \in A$. Soit L algébrique sur A alors

$$\sigma(L) = \bigcup_{w \in L} \sigma(w) \text{ est algébrique sur } B$$

Prop Si L est algébrique et K rationnel alors $L \cap K$ est algébrique

2) Indécidabilité

Prop Les problèmes suivants sont indécidables. Soit G et G' deux grammaires :

1) $L_G(S) \cap L_{G'}(S') = \emptyset$?
2) $L_G(S) = L_{G'}(S')$?
3) $L_G(S) = A^*$?
4) G ambigüe ?

III Automates à piles

Def Un automate à pile est constitué :
- un alphabet d'entrée A
- " " de pile Z et d'un symbole initial
- un ensemble fini d'état Q avec $q_0 \in Q$ initial
- des transitions : $(q, z) \xrightarrow{y} (q', h)$ $\quad q, q' \in Q$, $y \in A \cup \{\varepsilon\}$, $z \in Z$, $h \in Z^*$

On appelle configuration C un élément de $Q \times Z^*$

Def Si $C = (p, z\,w)$ $C' = (q, h\,w)$ si $p z \xrightarrow{y} q h$ alors $C \xrightarrow{y} C'$

Def Plusieurs mode de terminaison : Pile vide / état final / sommet de pile final

Tentes équivalents

Th Un langage est algébrique ssi il est reconnu par un automate à pile

D2 3/3 SIPSER ?

Def Un automate est dit déterministe si à partir d'une configuration, il n'y a qu'une transition franchissable

Def Un langage déterministe est un langage reconnu par un automate déterministe.

Prop Le complémentaire d'un langage déterministe est déterministe.

Prop Reconnaissance par pile vide $\Rightarrow$ reconnaissance par état final.

Prop Un langage déterministe est non ambigüe.

V Algorithmes

Algo de Cocke Kasami et Younger : Soit $w = a_1 \ldots a_n$ et $w_{ij} = a_i \ldots a_j$. Soit $E_{ij} = \{S / w_{ij} \in L_G(S)\}$.
On calcule les E_{ij} par programmation dynamique $(O(n^3))$

Analyse descendante : idée $(S, w) \xrightarrow{*} (\varepsilon, \varepsilon)$

$S \to S_1 S_2 \qquad (S, w) \to (S_1 S_2, w)$
$\qquad\qquad\qquad (Sa, aw) \to (S, w)$

Si on sait quelle règle appliquer à partir des k premières lettres de w alors : grammaire LL(k)

Analyse ascendante : $(\varepsilon, w) \xrightarrow{*} (S, \varepsilon)$

$S \to S_1 S_2 \qquad (S_1 S_2, w) \to (S, w)$
$\qquad\qquad\qquad (S, aw) \to (Sa, w)$

Si on sait quand réduire et quand lire = partir des k premières lettres : LR(k)