910 - Langages algébriques. Exemples et applications

On suppose connus les résultats sur les automates finis.

I) Représentations

1) Grammaires algébriques

a) Définition et premières propriétés

Déf 1: Une grammaire algébrique G est un triplet (Σ, Γ, P) où Σ et Γ sont des alphabets finis et P est une partie finie de $\Gamma \times (\Sigma \cup \Gamma)^*$. Les symboles de Σ sont appelés terminaux et ceux de Γ non terminaux ou variables. Les éléments de P sont appelés règles. On note $X \to u_1 + \dots + u_n$ pour $(X, u_1), \dots, (X, u_n) \in P$.

Ex 2: $G_1 = (\Sigma_1, \Gamma_1, P_1)$, $\Sigma_1 = \{a, b\}$, $V_1 = \{S\}$, $P_1 = \{S \to aSb, S \to \varepsilon\}$

Rq 3: Par convention, on notera toujours les non-terminaux en majuscules et les terminaux en minuscules. Ainsi, on pourra se contenter de donner P sans introduire d'ambiguïté.

Ex 4: $G_1 = \{S \to aSb + \varepsilon\}$, $G_2 = \left\{\begin{array}{l} P \to aI + \varepsilon \\ I \to aP \end{array}\right\}$,

$G_{3,n} = \left\{\begin{array}{l} S \to ST + \varepsilon \\ T \to a_1 S b_1 + \dots + a_n S b_n \end{array}\right\}$

Déf 5: On étend $\to$ à $(\Sigma \cup \Gamma)^*$ par $\alpha X \beta \to \alpha u \beta$ si $X \to u$. On note $\to^*$ la clôture réflexive transitive de $\to$. Si $u \to^* v$, on dit que u se dérive en v.

Ex 6: $S \to aSb \to aaSbb \to aabb$ dans G_1.

Déf 7: On note $\hat{L}_G(u) = \{v \in (\Sigma \cup \Gamma)^* \mid u \to^* v\}$ et $L_G(u) = \hat{L}_G(u) \cap \Sigma^*$. On appelle $L_G(u)$ le langage engendré par u dans G.

Ex 8: $L_{G_1}(S) = \{a^n b^n \mid n \in \mathbb{N}\}$, $L_{G_2}(P) = \{a^{2n} \mid n \in \mathbb{N}\}$, $L_{G_2}(I) = \{a^{2n+1} \mid n \in \mathbb{N}\}$

$D_n^* := L_{G_{3,n}}(S)$ est appelé langage de Dyck. C'est le langage des mots bien parenthésés (si l'on considère a_i comme une parenthèse ouvrante et b_i comme la parenthèse fermante correspondante).

Déf 9: Un langage est algébrique ssi il est engendré par un non terminal dans une grammaire algébrique.

Lemme 10 (Fondamental): $(u_1 u_2 \to^k v) \iff \left(\begin{array}{l} \exists v_1, v_2 \\ \exists k_1, k_2 \end{array} \text{ tq } \left\{\begin{array}{l} v = v_1 v_2 \\ u_1 \to^{k_1} v_1 \\ u_2 \to^{k_2} v_2 \\ k = k_1 + k_2 \end{array}\right.\right)$

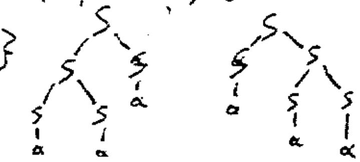
b) Arbre de dérivation

Déf 11: Un arbre de dérivation partiel est un arbre fini dont les noeuds sont étiquetés par $\Sigma \cup \Gamma \cup \{\varepsilon\}$ tel que si X étiquette un noeud et $a_1, \dots, a_n$ étiquettent ses fils, alors $X \to a_1 \dots a_n$. Un arbre de dérivation est un arbre de dérivation partiel dont les feuilles sont étiquetées par $\Sigma \cup \{\varepsilon\}$.

Déf 12: La frontière d'un arbre de dérivation partielle est la concaténation de gauche à droite de ses feuilles.

Prop 13: $L_G(S)$ (resp. $\hat{L}_G(S)$) est l'ensemble des frontières d'arbres de dérivation (resp. partielle) de racine étiquetée par S.

Exemple 14: $G_4 = \{S \to SS + a\}$



Déf 15: Une grammaire est dite ambiguë s'il existe un mot qui est frontière de deux arbres de dérivation distincts dont la racine est étiquetée par un même non terminal.

Ex 16: G_4 est ambiguë.

Rq 17: $G_5 = \{S \to aS + a\}$ n'est pas ambiguë et $L_{G_5}(S) = L_{G_4}(S)$

c) Simplification des grammaires

Déf 18: Une grammaire est dite réduite pour $S \in V$ ssi : $\forall T \in \Gamma, L_G(T) \neq \emptyset$
- $\forall T \in \Gamma, \exists u, v \in (\Sigma \cup \Gamma)^*$ tq $S \to^* uTv$

Déf 19: Une grammaire est dite propre ssi elle ne contient aucune règle de la forme $S \to \varepsilon$ ou $S \to T$.

Déf 20: Une grammaire est en forme normale quadratique / de Chomsky ssi toutes ses règles sont de la forme $S \to UV$ ou $S \to a$.

Déf 21: Une grammaire est en forme normale de Greibach (resp. Greibach quadratique) ssi toutes ses règles sont de la forme $S \to w$ avec $w \in \Sigma\Gamma^*$ (resp. $\Sigma \cup \Sigma\Gamma \cup \Sigma\Gamma^2$).

Prop 22: Pour toute grammaire algébrique de taille n engendrant L pro[...]

- L est engendré par une grammaire réduite de taille $O(n)$ calculable en temps $O(n)$.
- $L\setminus\{\varepsilon\}$ est engendré par une grammaire propre de taille $O(n^2)$ calculable en temps $O(n^2)$.
- $L\setminus\{\varepsilon\}$ est engendré par une grammaire en forme normale quadratique de taille $O(n^2)$ calculable en temps $O(n^2)$.
- $L\setminus\{\varepsilon\}$ est engendré par une grammaire en forme normale de Greibach.

2) Automates à pile

Déf 23: Un automate à pile est constitué : - d'un alphabet d'entrée Σ
- d'un alphabet de pile Z dont un symbole initial z_0
- d'un ensemble fini d'états Q dont un état initial q_0
- de transitions de la forme $q, z \xrightarrow{y} q', h$ avec $q, q' \in Q$, $y \in \Sigma \cup \{\varepsilon\}$, $z \in Z$, $h \in Z^*$.

Déf 24: Une étape de calcul d'un automate à pile (AP) est une paire de configurations (C, C') notée $C \xrightarrow{y} C'$ telles que $C = (p, zw)$, $C' = (q, hw)$ et $p, z \xrightarrow{y} q, h$.
Un calcul est une succession d'étapes de calcul : $C_0 \xrightarrow{y_1} C_1 \xrightarrow{y_2} \dots \xrightarrow{y_n} C_n$.
Le mot $y_1 \dots y_n$ est l'étiquette du calcul.

Déf 25: Un AP accepte un mot u **par pile vide (resp. par état final)** ssi $(q_0, z_0) \xrightarrow{u}{}^* (q, \varepsilon)$ (resp. (q, w) où $q \in F$, F étant un sous-ensemble distingué de Q, et w est quelconque).

Prop 26: **L'ensemble des langages acceptés** par pile vide est égal à l'ensemble des langages acceptés par état final.

Thm 27: Un langage L est algébrique ssi il existe un AP qui accepte L.

Déf 28: Un AP est dit déterministe ssi pour tout C, **toutes** les étapes de calculs partant de C sont étiquetées par des symboles de Σ deux à deux distincts, ou il en existe une unique étiquetée par ε.

Rq 29: Il n'y a pas équivalence des modes d'acceptation pour les AP déterministes.

II) Propriétés

1) Lemme d'itération

Lemme 30: (Ogden) Pour toute grammaire et toute variable $S \in V$, il existe un entier $K \in \mathbb{N}$ tel que pour tout mot $f \in \hat{L}_G(S)$ ayant au moins K lettres distinguées se factorise en $f = \alpha u \beta v \gamma$ tq :
- $S \xrightarrow{*} \alpha T \gamma$ et $T \xrightarrow{*} u T v + \beta$
- Soit α, u et β, soit β, v et γ contiennent des lettres distinguées
- $u \beta v$ contient moins de K lettres distinguées

Corollaire 31: (Théorème de Bar-Hillel, Perles et Shamir) Pour tout langage algébrique L, il existe $N \geq 0$ tq pour tout mot $f \in L$, si $|f| \geq N$ alors f se factorise en $f = \alpha u \beta v \gamma$ tq $|uv| > 0$, $|u\beta v| < N$ et $\alpha u^n \beta v^n \gamma \in L$ pour tout n.

Appli 32: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ n'est pas algébrique. Il peut représenter de la mise en forme en mode texte : $\underline{\text{titre}}$

Appli 33: $\{a^m b^n c^m d^n \mid m, n \in \mathbb{N}\}$ n'est pas algébrique. Il peut représenter la déclaration de deux procédures (à arité m et n) ~~et leurs utilisations~~.

c'est une abstract° de ...

c pr... au ... séparer analyse syntaxique et lexicale

2) Propriétés de clôture

Déf 34: Un morphisme de X^* dans Y^* est une fonction φ de X^* dans Y^* tq $\varphi(\varepsilon) = \varepsilon$ et $\varphi(uv) = \varphi(u)\varphi(v)$. Un morphisme est uniquement déterminé par son action sur X.
L'image de $L \subseteq X^*$ par φ est $\{\varphi(u) \mid u \in L\}$ et l'image inverse de $L \subseteq Y^*$ par φ est $\{u \mid \varphi(u) \in L\}$.

Déf 35: Une substitution algébrique est une fonction $\sigma : X^* \to \mathcal{P}(Y^*)$ telle que $\sigma(\varepsilon) = \{\varepsilon\}$ et $\sigma(uv) = \sigma(u)\sigma(v)$ et pour tout $a \in X$, $\sigma(a)$ est algébrique.

Prop 36: L'ensemble des langages algébriques est stable par union, concaténation, passage à l'étoile, intersection avec un rationnel, substitution algébrique, morphisme et morphisme inverse mais ni par complémentation ni par intersection.

p bien comprendre, il faut commencer par se demander ce qu'il se passe avec D_1

3) Théorème de Chomsky et Schützenberger

Thm 37 : (Chomsky et Schützenberger) Un langage est algébrique ssi $L = \varphi(D_n^* \cap K)$ pour un $n \in \mathbb{N}$, un langage rationnel K et un morphisme alphabétique φ $(\forall u, |\varphi(u)| \leq 1)$.

Lemme 38 : Il existe un morphisme $\psi : \Sigma_{3,n}^* \to \Sigma_{3,2}^*$ tq $D_n^* = \psi^{-1}(D_2^*)$.

Corollaire 39 : Un langage est algébrique ssi $L = \varphi(\psi^{-1}(D_2^*) \cap K)$ pour ~~des~~ morphismes et K langage rationnel.

Rq Les applications de la forme $X \mapsto \varphi(\psi^{-1}(X) \cap K)$ sont les transductions rationnelles et apparaissent naturellement lors de l'étude des langages rationnels.

IV) Sous-classes remarquables

L'annexe A ~~illustre~~ les inclusions entre ces classes.

1) Langages rationnels

Prop 40 : Tout langage rationnel est algébrique et l'inclusion est stricte.

2) Langages déterministes

Déf 41 : Un langage algébrique est dit déterministe (resp. déterministe préfixe) ssi il est accepté par un AP déterministe par état final (resp. par pile vide).

Prop 42 : Tout langage rationnel est déterministe et l'inclusion est stricte.

Prop 43 : L'ensemble des langages algébriques déterministes est stable par complémentation et intersection avec un rationnel mais ni par union, ni par intersection.

Prop 44 : Un langage algébrique est déterministe-préfixe ssi il est déterministe et préfixe ($\forall u, v \in L$, u n'est pas préfixe strict de v)

Rq 45 : Soit L un langage tq $L \cap \$^* = \emptyset$. Alors $L\$$ est préfixe.

3) Langages ambigus

Déf 46 : Un langage algébrique est dit ambigu si toutes les grammaires algébriques qui l'engendrent le sont et non ambigu sinon.

Prop 47 : Tout langage algébrique déterministe est non ambigu et l'inclusion est stricte.

Prop 48 : L'inclusion des langages non ambigus dans les langages algébriques est stricte.

Prop 49 : L'ensemble des langages algébriques non ambigus est stable par union disjointe, ~~union avec un rationnel~~ et intersection avec un rationnel mais pas par union.

III) Problèmes de décision

1) Problèmes décidables

Th 50 : La vacuité d'un langage algébrique (représenté par une grammaire algébrique G) est décidable en temps $O(|G|)$.

Th 51 : L'appartenance d'un mot w à un langage algébrique (représenté par une grammaire algébrique G en forme normale quadratique) est décidable en temps $O(|G||w|^3)$.

2) Problèmes indécidables

Th 52 : Les problèmes suivants sont indécidables : [DEV] (PCP adapté)

- Pour deux grammaires, l'intersection des langages engendrés est-elle vide ?
- Pour deux grammaires, engendrent-elles le même langage ?
- Pour une grammaire, engendre-t-elle Σ^* ?
- Pour une grammaire, est-elle ambiguë ?

V) Applications

Déf 53: Soit G une grammaire. À chaque $A \to \alpha\beta \in P$, on associe $[A \to \alpha \cdot \beta]$ qu'on appelle item. On note It_G l'ensemble des items. Soit $S \in \Gamma$ une variable distinguée. On appelle automate des items l'automate dont l'alphabet d'entrée est Σ, dont l'alphabet de pile est It_G, dont les transitions sont

(E) $[X \to \beta . Y\gamma] \xrightarrow{\varepsilon} [X \to \beta . Y\gamma][Y \to . \alpha]$ pour $Y \to \alpha \in P$

(L) $[X \to \beta . a\gamma] \xrightarrow{a} [X \to \beta a . \gamma]$

(R) $[X \to \beta . Y\gamma][Y \to \alpha .] \xrightarrow{\varepsilon} [X \to \beta Y . \gamma]$

dont l'état initial est $[S' \to . S]$ et dont l'état final est $[S' \to S .]$ où S' est une variable que l'on ajoute à Γ.

Thm 54: L'automate des items accepte $L_G(S)$. [DEV]

1) Analyse descendante (LL(k))

On part de S et on essaye d'appliquer des règles pour arriver au mot en faisant des dérivations gauches.

"Déf 55": Une grammaire est dite $LL(k)$ si il est possible de savoir quelle règle utiliser pour dériver le non terminal le plus à gauche en connaissant les k prochains symboles du mot.

2) Analyse ascendante (LR(k))

On part du mot et on essaye d'appliquer les règles $X \to u$ pour réduire le mot en S (en appliquant l'inverse de dérivations droites).

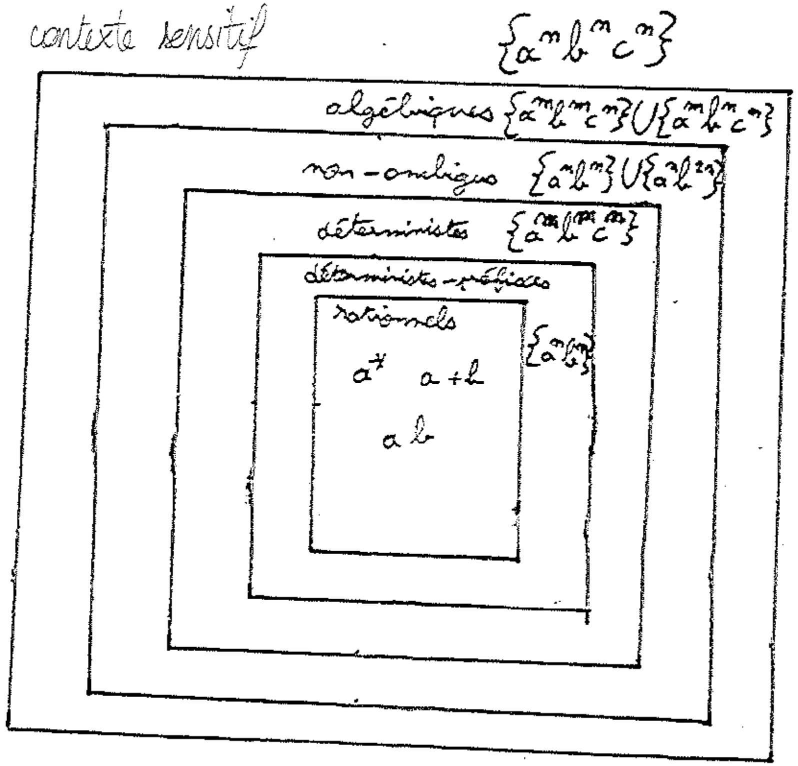
"Déf 56": $LR(k)$ si on sait quand effectuer une lecture ou une réduction avec les k premières lettres après la partie du mot à potentiellement réduire.

Thm 57: Un langage algébrique est engendré par une grammaire $LR(0)$ ssi il est déterministe-préfixe.

Thm 58: Un langage algébrique est engendré par une grammaire $LR(k)$, $k \geq 1$ ssi il est engendré par une grammaire $LR(1)$ ssi il est déterministe.

pas récu énum.

récursivem^t énumérables



Annexe A

Rq: • Il manque au départ, une motivation, un contexte général.

• Il faut être enthousiaste dans la présentat° du plan; articuler le discours...

• Quand on étudie la complexité, il faut qu'il y ait qqch derrière, un but, un bilan.. Qu'il faut mettre en avant.

• Contre-exemple; Lemme d'Ogden pas CS.

(Rq: Est-ce qu'un langage est algébrique // Lemme d'Ogden: pas décidables ?)

• Ptés de clôture: on peut s'en servir pour des raisonnements par l'absurde, ...

On ne sait pas dire vite $L_1 \subseteq L_2$ pour les algébriques (Pb de décision III)

↳ Du coup on peut commencer à distinguer des sous-classes. Si L_1 rationnel, L_2 déterministe,

↳ mieux ?

$L \subset R$? → facile : on décide $L \cap R^c = \emptyset$?

↳ Sous classe remarquables: on veut améliorer les ppté de clôture et/ou les problèmes de décision

Question : Décidables ?

— Algébrique? si on a une grammaire gale

— Régulier? si on a un algébrique?

— Pour L algébrique. Est-ce que L est déterministe? décidable

[Carton]: Les mots de la pile: langage régulier

Donc si on peut majorer le langage de pile est $< \infty$, (nbre fini de configurat° ⇒ automate fini) alors le langage alg. est régulier.

Compléments : Langages récursivement énumérables; thm de Reisz