Chapitre 11

Analyse grammaticale

Dans ce chapitre nous donnons quelques éléments d'introduction de l'analyse grammaticale de certaines sous-classes de langages algébriques. L'analyse syntaxique est un passage obligé dans le processus de compilation et d'interprétation d'un programme.

Nous avons vérifié dans le chapitre précédent que pour tout langage algébrique, il existe un algorithme qui à chaque mot du langage associe un arbre de dérivation. De manière générale c'est possible en temps de l'ordre de $O(n^3)$. Dans la suite de ce chapitre nous nous intéresserons à des langages algébriques particuliers pour lesquels il est possible de donner un arbre en temps linéaire (c'est-à-dire en $O(n)$).

Dans tout ce chapitre on ne considère que des grammaires algébriques dont toutes les variables sont utiles, c'est-à-dire servent à écrire des mots terminaux.

11.1. Analyse descendante et les grammaires LL(k)

Introduction. Considérons une grammaire algébrique $\mathcal{G}$ non ambiguë. Soit w un mot engendré par $\mathcal{G}$. Alors il existe une unique dérivation gauche qui produit w à partir de l'axiome S de la forme

$$S \to a_1...a_{i_1}V_1\beta_1 \to a_1...a_{i_2}V_2\beta_2 \to ... \to a_1...a_{i_n}V_n\beta_n \to a_1.....a_m = w.$$

L'idée de l'analyse descendante (descendante dans l'arbre de dérivation) consiste à essayer de retrouver cette dérivation gauche en lisant linéairement de gauche à droite le mot w. Par exemple, il y a des grammaires telles que la production d'une lettre à gauche à partir d'une variable donnée n'est possible que d'une seule manière. Dans ce cas on peut reconstruire facilement la suite de productions.

11.1.1. Exemple. Soit $\mathcal{G}_1$ la grammaire ayant pour alphabet $\{a, b, c\}$, pour variables $\{S, T, U\}$, S pour axiome et pour productions :

$$\begin{aligned} S &\to aS|bcT|cTT \\ T &\to cU|ab \\ U &\to aTU|cS \end{aligned}$$

Considérons le mot $w = acabcaabcbcab$. Alors pour déterminer si ce mot est engendré ou non par $\mathcal{G}_1$ et en même temps une dérivation, on lit le mot de gauche à droite et on constate :

1. Ce mot commence par a donc seule la production $S \to aS$ a pu être utilisée au départ.
2. Ensuite on lit la lettre c donc on a nécessairement utilisée la production $S \to cTT$.
3. Puis la lettre a correspond à $T \to ab$. On en est alors à la dérivation

$$S \to aS \to acTT \to acabT.$$

4. Puis on lit donc c qui correspont à $T \to cU$.
5. Ainsi de suite. Et on obtient

$$\to aS \to acTT \to acabT \to acabcU \to acabcaTU \to acabcaabU$$
$$\to acabcaabcS \to acabcaabcbcT \to acabcaabcbcab.$$

11.1.2. Exemple. Soit $\mathcal{G}_2$ la grammaire ayant pour alphabet $\{a, b\}$, pour variables $\{S, T, U\}$, S pour axiome et pour productions :

$$S \to ab|bT|TT$$
$$T \to bU$$
$$U \to aS$$

Cette grammaire n'a pas la propriété de la précédente grammaire. En particulier la lettre b peut être produite à partir de S de deux façons : par exemple $S \to bT$ et $S \to TT \to bUT$. Mais par contre, si l'on regarde la lettre suivante, on peut lever la détermination : on a en fait $S \to bT \to bbU$ et $S \to TT \to bUT \to baST$.

Généralisons maintenant cette méthode qui consiste à trouver une dérivation gauche par lecture linéaire du mot, éventuellement en anticipant en lisant systématiquement k lettres.

11.1.3. Définition. Soit Σ un alphabet et k un entier strictement positif. Pour tout mot $w \in \Sigma^*$ on définit le *préfixe de longueur au plus* k de w par

$$\text{Préfixe}_k(w) = \begin{cases} w \text{ si } |w| \leq k \\ u \text{ tel que } w = uv \text{ et } |u| = k \text{ sinon.} \end{cases}$$

Une grammaire algébrique $\mathcal{G}$ est dite $LL(k)$ si pour tout variable T admettant deux dérivations gauches $T \xrightarrow{*} w_1$ et $T \xrightarrow{*} w_2$ où $w_1, w_2 \in \Sigma^*$ telles que $\text{Préfixe}_k(w_1) = \text{Préfixe}_k(w_2)$ alors ces deux dérivations commencent par la même production $T \to u$ (i.e. $T \to u \xrightarrow{*} w_1$ et $T \to u \xrightarrow{*} w_2$).

11.1.4 Proposition. *Tout langage engendré par une grammaire* $LL(k)$ *est reconnaissable par un automate à pile déterministe. Plus exactement, si l'on considère* L *une langage engendré par une grammaire* $LL(k)$ *et* \$ *un nouveau symbole (pour fin de mot) alors le langage* $L\$$ *est reconnaissable par un automate à pile déterministe par pile vide.*

Démonstration. Soit Σ l'aplphabet de L et V l'ensemble des variables de $\mathcal{G}$. On considère l'automate à pile suivant :

- ensemble d'états : $\{q_v, q'_v : v \in \Sigma^*, |v| \leq k\}$,
- alphabet de ruban : $\Sigma \cup \{\$\}$,
- alphabet de pile : $\Sigma \cup V$,
- état initial : q_ε le mot vide,
- pile initiale : S, l'axiome de $\mathcal{G}$,

Les transitions sont les suivantes :

- $(q_v, x) \xrightarrow{a} (q_{va}, x)$ pour tout $|v| < k$, toute lettre $a \in \Sigma \cup \{\$\}$ et tout symbole de haut pile x (lecture du préfixe) ;
- $(q_v, T) \xrightarrow{\varepsilon} (q_v, u)$ pour tout $|v| = k$ et toute production $T \to u$ telle qu'il existe une dérivation gauche $T \to u \xrightarrow{*} w$ avec $\text{Préfixe}_k(w) = v$ (empilement correspondant à une production) ;

– $(q'_v, T) \xrightarrow{\varepsilon} (q'_v, u)$ pour toute production $T \to u$ telle qu'il existe une dérivation gauche $T \to u \xrightarrow{*} w$ avec $\text{Préfixe}_k(w) = v$ (empilement correspondant à une production, en fin de mot) ;
– $(q_v, a) \xrightarrow{\varepsilon} (q_{v'}, \varepsilon)$ pour tout $|v| = k$ et toute lettre $a \in \Sigma$ tels que $v = av'$ (dépilement de la première lettre du préfixe) ;
– $(q'_v, a) \xrightarrow{\varepsilon} (q'_{v'}, \varepsilon)$ pour toute lettre $a \in \Sigma$ telle que $v = av'$ (dépilement de la première lettre du préfixe, en fin de mot) ;

On vérifie alors par récurrence que les dérivations gauches de S vers un mot terminal w correspondent aux chemins dans l'automate de l'état initial avec pile initiale S vers la pile vide et l'état q'_ε étiqueté par w.

□

Remarque. Rappelons que nous pouvons supprimer les boucles infinies d'un automate à pile déterministe et ainsi borner le nombre de transitions instantannées successives. Il suit que tout langage reconnaissable par un automate à pile déterministe est reconnaissable par un algorithme en temps linéaire. Si $\mathcal{G}$ est une grammaire $LL(k)$, il existe un algorithme linéaire qui décide si un mot est engendré par $\mathcal{G}$ et si c'est le cas donne alors l'unique dérivation gauche associée à ce mot (une telle grammaire est donc non-ambiguë).

11.2. Analyse ascendante et les grammaires LR(0)

De nombreux langages algébriques ne possèdent pas de grammaire $LL(k)$. En particulier les langages utilisant des systèmes de parenthèses. Considérons l'exemple suivant :

11.2.1. Exemple. Considérons la grammaire $\mathcal{G}_3$ qui a pour variables S et E (S l'axiome), pour alphabet $\Sigma = \{(,), *, +, a, b\}$ et pour productions :

$$S \to S * E | E$$
$$E \to (S + S) | a | b$$

Par exemple l'expression arithmétique $((a+b)*a*(b+a)+b)$ est engendrée par $\mathcal{G}_3$. Cette grammaire n'est $LL(k)$ pour aucun k. En effet deux expressions peuvent être engendrées par $\mathcal{G}_3$ et commencées par k parenthèses ouvrantes (avec respectivement dans leurs dérivations gauches pour première production $S \to S * E$ et $S \to E$.

Nous allons voir que pour une telle grammaire il est quand même possible de reconstruire les arbres de dérivations en temps linéaire. Pour celà, il faudra reconstruire les arbres de bas en haut (analyse ascendante) ; c'est-à-dire une dérivation à partir de la droite. La lecture du mot se fera toujours à partir de la gauche. Les grammaires permettant de reconstruire uen dérivation de cette façon s'appelle grammaire LR pour "Left-Right". Le but de la suite de cette partie est de donner précisément la définition de cette notion et de montrer qu'elle est en correspondance avec las automates à pile déterministe.

11.2.2. Contextes gauches. Nous rappelons qu'une dérivation est droite si et seulement à chaque étape la réécriture porte sur la variable la plus à droite. Dans toute la suite de cette partie nous ne considérerons que des dérivations droites. On notra une telle dérivation $u \xrightarrow{*}_d v$.

Considérons maintenant une réécriture droite

$$uTw \to_d uvw$$

où u est un mot constitué de variables et terminaux, T est une variable, w est un mot terminal et $T \to v$ est une production. Le principe de l'analse ascendante est de pouvoir déterminer la production $T \to v$ à partir de la connaissance de uv. Pour cela, nous définissons la notion de *contexte gauche* d'une variable.

11.2.3 Définition. Soit $\mathcal{G}$ une grammaire algébrique. On appelle *contexte gauche* d'une variable T l'ensemble des facteurs à sa gauche dans toute dérivation droite :

$$CG(T) = \{u \mid S \xrightarrow{*}_d uTv,\ u, v \in (\Sigma \cup V)^*\}.$$

Par exemple pour la grammaire d'axiomes S et de productions $S \to aT$ et $T \to aTcTba|a|b$ les contextes gauches sont :

$$CG(S) = \{\varepsilon\},$$
$$CG(T) = a(a \cup aTc)^*$$

11.2.4 Proposition. *Les contextes gauches des variables (utiles) d'une grammaire algébrique sont des langages rationnels.*

Démonstration. On remarque que les contextes gauches vérifient le système d'équations

$$CG(U) = CG(T_1)u_1 \cup CG(T_2)u_2 \cup ... \cup \varepsilon(U)$$

où pour tout i, u_iU apparait dans une production sous la forme $T_i \to u_iUv_i$ et où $\varepsilon(U)$ est égal à $\{\varepsilon\}$ si $U = S$ l'axiome, est vide sinon. (La notation $CG(T)u$ signifie $CG(T) \cdot \{u\}$.) Ce système est un système d'équations linéaires droites, qui peut être associé à un automate fini dont les états peuvent être représentés par les variables, l'état initial par S et les transitions sont de la forme $T_i \xrightarrow{u_i} U$. Alors chaque contexte U correspond à l'ensemble des mots qui étiquettent un chemin de S vers U. □

On appelle *contexte gauche d'une production* $T \to u$ le langage

$$CG(T \to u) = CG(T)u$$

11.2.5. Exemple. Reprenons la grammaire $\mathcal{G}_3$ de l'exemple 11.2.1. L'automate correspondant aux contextes gauches est alors :

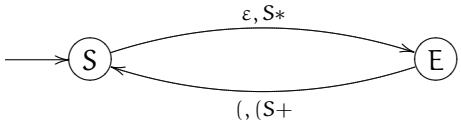

FIG. 11.1 – Automate des contextes gauches de $\mathcal{G}_3$

Les contextes gauches valent donc :

$$CG(S) = \{(, (S+, S * (, S * (S+\}^*,$$
$$CG(E) = CG(S) \cup CG(S)S*,$$
$$CG(S \to S * E) = CG(S)S * E,$$
$$CG(S \to E) = CG(S)E,$$
$$CG(E \to (S + S)) = CG(E)(S + S),$$
$$CG(E \to a) = CG(E)a,$$
$$CG(E \to b) = CG(E)b$$

11.2.6. Définition. L'ensemble des préfixes des mots d'un langage L est :

$$\text{Préfixe}(L) = \{u \mid uv \in L \text{ pour un mot } v\}.$$

Une grammaire algébrique est dite $LR(0)$ si pour toutes productions $T \to t$ et $U \to u$ on a la propriété

$$CG(T \to t) \cap \text{Préfixe}(CG(U \to u)) = \emptyset \text{ et } S \notin CG(T \to t).$$

Remarque la deuxième condition est utile pour savoir si la lecture de S correspond au début d'une dérivation ou non. Elle est vérifiée dès que l'on n'a pas de production de vide ni de production de la forme $T \to S$ (ou si S n'apparaît dans aucune production, il suffit pour cela de remplacer l'axiome S par un axome S_0 avec la production $S \to S_0$).

11.2.7. Exemple. On vérifie facilement que les contextes gauches des productions de la grammaire $\mathcal{G}_3$ ci-dessus vérifient cette propriété et donc que $\mathcal{G}_3$ est une grammaire $LR(0)$.

11.2.8 Proposition. *Tout langage engendré par une grammaire* $LR(0)$ *est reconnaissable par un automate à pile déterministe. Plus exactement, si l'on considère* L *un langage engendré par une grammaire* $LR(0)$ *et* \$ *un nouveau symbole (pour fin de mot) alors le langage* $L\$$ *est reconnaissable par un automate à pile déterministe par pile vide.*

Réciproquement tout langage reconnu par un automate à pile déterministe par pile vide admet une grammaire $LR(0)$.

Démonstration. Nous ne présentons ici que la construction de l'automate à pile correspondant à une grammaire $LR(0)$. Soient $P_1, ..., P_p$ les productions de la grammaire $\mathcal{G}$. Pour chaque i, notons $\mathcal{A}_i$ un automate fini reconnaissant le $CG(P_i)$. Ces automates seront considérés avec un état rebut. L'idée est de faire marcher ces automates en parallèles dans une pile. Par la propriété $LR(0)$ au plus un de ces automates peut se trouver en même temps dans un état final. La pile représentera la suite des états de ces automates d'une part et le facteur à gauche de la tête de lecture à la hauteur où l'on a remonté dans la dérivation à l'instant donné. Formellement, on considère l'automate à pile $\mathcal{A}_\mathcal{G}$ défini de la manière suivante :

- L'alphabet de ruban est $\Sigma \cup \{\$\}$;
- L'alphabet de pile est $X \times Q_1 \times Q_2 \times \cdot \times Q_p$ où $X = \Sigma \cup V \cup \{F\}$ et chaque Q_i est l'ensemble des états de $\mathcal{A}_i$ (où F est un nouveau symbole pour le fond de la pile).
- La pile initiale est $(F, q_1^0, .., q_p^0)$ où les q_i^0 sont les états initiaux des $\mathcal{A}_i$.

L'automate est ensuite constitué principalement d'un état *contrôle* qui est l'état initial. Cet état contrôle si l'un des automates $\mathcal{A}_i$ est dans un état final ou non, et fait alors les opérations suivantes :

- Si aucun des automates $\mathcal{A}_i$ n'est dans un état final (i.e. le haute de la pile est de la forme $(\pi, q_1, ..., q_p)$ avec aucun état q_i final) et si la tête de lecture est sur une lettre $a \in \Sigma$ alors on empile $(a, p_1, ..., p_p)$ tels que $q_i \xrightarrow[\mathcal{A}_i]{a} p_i$ et on déplace la tête de lecture. On obtient alors la pile suivante :

a	p_1	p_2	$\cdots$	p_p
x	q_1	q_2	$\cdots$	q_p
F	q_1^0	q_2^0	$\cdots$	q_p^0

– Si l'un des automates $\mathcal{A}_i$ est dans un état final (il y en a au plus un) alors $\mathcal{A}_i$ a reconnu un contexte gauche de la production $T_i \to u_i$ et la pile est de la forme suivante avec $u_i = x_1 ... x_n$:

x_n	p_1^n	p_2^n	$\cdots$	p_p^n
x_1	p_1^1	p_2^1	$\cdots$	p_p^1
x	q_1	q_2	$\cdots$	q_p
F	q_1^0	q_2^0	$\cdots$	q_p^0

On dépile u_i et on empile T_i ce qui donne la pile suivante tel que $q_i \xrightarrow[\mathcal{A}_i]{T} p_i$:

T	p_1	p_2	$\cdots$	p_p
x	q_1	q_2	$\cdots$	q_p
F	q_1^0	q_2^0	$\cdots$	q_p^0

– Si le haut de la pile est de la forme $(S, q_1, .., q_p)$ où aucun des q_i n'est dans un état final et si la tête de lecture est sur la lettre \$ alors on dépile et on se place dans état auxiliaire et on dépile à nouveau une fois. On se retrouvera alors sur la pile vide si la pile était en fait dans la configuration suivante (avec $q_i^0 \xrightarrow[\mathcal{A}_i]{S} p_i$) :

S	p_1	p_2	$\cdots$	p_p
F	q_1^0	q_2^0	$\cdots$	q_p^0

Remarque : la deuxième hypothèse sur les grammaires $LR(0)$ (i.e. S n'appartient à aucun contexte gauche d'aucune production, permet de s'assurer que si la pile est dans la configuration ci-dessus alors aucun des $\mathcal{A}_i$ n'est dans un état final).
Notons que l'on peut arrêter l'analyse dès que tous les automates sont en même temps dans l'état rebut.

□

11.2.9. Remarque. Le principe d'analyse ascendante s'étant naturellement de la façon suivante : il peut être nécessaire pour déterminer une production de connaître le facteur gauche de celle-ci mais également le mot de longueur au plus k qui suit. Il s'agit des grammaires $LR(k)$ pour lesquelles on détermine la réécriture droite

$$S \xrightarrow{*}_d uTw \to_d uvw$$

par la connaissance de uv et du préfixe $\text{Préfixe}_k(w)$.

EXERCICES

11.1 Exercice. Déterminer si les mots suivants sont engendrés par la grammaire de l'exemple 11.1.1 et le cas échéant la dérivation gauche correspondante : bccccab, ccaaaacbcabcab et bcccbcccab.

11.2 Exercice. Décrire un automate à pile déterministe reconnaissant le langage engendré par la grammaire de l'exemple 11.1.2.

11.3 Exercice. Calculer les contextes gauches des grammaires suivantes. Déterminer celles qui sont $LR(0)$.

1. $S \to aSa|bSb|c$;
2. $S \to aSb|bSa|c$;
3. $S \to aSa|cbSb|ac$

11.4 Exercice. Simuler l'automate à pile associé à la grammaire $\mathcal{G}_3$ sur le mot $((a+b)*a*(b+a)+b)$.