CH.4 Grammaires non contextuelles

- 4.1 Les dérivations
- 4.2 La simplification
- 4.3 La forme normale de Chomsky
- 4.4 La forme normale de Greibach
- 4.5 Les grammaires linéaires droites

4.1 Les dérivations

Modèle pour la génération d'expressions satisfaisant une syntaxe plus évoluée que ce que permettent les expressions régulières. Par exemple, les expressions parenthésées.
Contrepartie : analyse plus difficile, ne s'appliquant que pour certains langages.

Grammaire non contextuelle G = (V, T, P, S).
V alphabet fini de variables (A, B, ...) ;
T alphabet terminal (a, b, 1, **id**, ...) ;
S axiome = variable distinguée ;
P ensemble de productions $A \rightarrow \alpha$,
où α est un mot sur $V \cup T$. Les productions de A sont, pour abréger, séparées par | .
On notera aussi (X,Y, ...) des symboles dans $V \cup T$,
et (u, v, x, ...) des mots sur T.

On utilise les productions pour récrire les variables.
Si $A \to \beta$ est une production de G, et si $\alpha A \gamma$ est un mot sur $V \cup T$, on dit que $\alpha A \gamma$ dérive immédiatement en $\alpha\beta\gamma$ et on note : $\alpha A \gamma \Rightarrow \alpha\beta\gamma$. La fermeture réflexive et transitive de $\Rightarrow$ est notée $\overset{*}{\Rightarrow}$.
Le langage engendré par G est l'ensemble L(G) des mots w sur T tels que $S \overset{*}{\Rightarrow} w$.
Les mots sur $V \cup T$ qui dérivent de S, les métamots, ne sont intéressants que comme étapes intermédiaires.

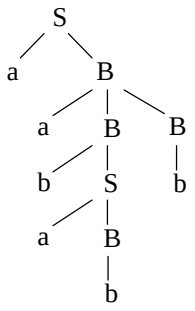
Exemples :
G = (V, T, P, S), avec V = {S}, T = {0, 1}, et P =
$S \to 0S1 \mid 01$.
Seule dérivation possible donnant un mot :
$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0^3S1^3 \Rightarrow ... \Rightarrow 0^{n-1}S1^{n-1} \Rightarrow 0^n1^n$.
Donc $L(G) = \{0^n1^n, n \geq 0\}$.

En général très difficile de caractériser le langage engendré par une grammaire.

Exemple :
G = (V, T, P, S), avec V = {S, A, B}, T = {a, b}, et P =
$S \to aB \mid bA$
$A \to a \mid aS \mid bAA$ $\quad$ $B \to b \mid bS \mid aBB$
Montrer que L(G) = mots ayant autant de a que de b.
Pour cela, montrer par récurrence sur la longueur de w que
$S \overset{*}{\Rightarrow} w \Leftrightarrow$ w a autant de a que de b,
$A \overset{*}{\Rightarrow} w \Leftrightarrow$ w a un a de plus que de b,
$B \overset{*}{\Rightarrow} w \Leftrightarrow$ w a un b de plus que de a.

Arbres de dérivation.
On représente une dérivation par un arbre : il conserve l'historique de la dérivation et est plus canonique.

Exemple :
Grammaire précédente et w = aababb.



Dérivations :
- à gauche
  $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabaBB$
  $\Rightarrow aababB \Rightarrow aababb$;
- à droite
  $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aaBb \Rightarrow aabSb$
  $\Rightarrow aabaBb \Rightarrow aababb$.

On montre par récurrence sur la longueur de la dérivation le lemme suivant.

Lemme : Un métamot α dérive d'une variable A si et seulement s'il existe un arbre de dérivation de racine A dont les feuilles lues de gauche à droite donnent α.

Une grammaire est ambiguë si un même mot possède plus d'un arbre de dérivation.

L'analyse syntaxique consiste, étant donné un mot, à dire s'il est engendré par une grammaire donnée. Si oui, à produire un arbre de dérivation. Les techniques classiques d'analyse descendante et ascendante ne s'appliquent qu'à des grammaires non ambiguës.
La grammaire de l'exemple est ambiguë : par exemple, le mot bbbaababaa possède deux arbres de dérivation.

Deux grammaires engendrant le même langage sont équivalentes.
L'ambiguïté est une propriété des grammaires et non des langages.
Pour un même langage, certaines grammaires peuvent être ambiguës, d'autres non. Parfois, toutes sont ambiguës. Un tel langage est dit intrinsèquement ambigu.

Exemple : Les deux grammaires suivantes engendrent les expressions algébriques parenthésées.

$E \rightarrow E + E \mid E * E \mid (E) \mid \mathbf{id}$ est ambiguë ;

$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid \mathbf{id}$ n'est pas ambiguë.

4.2 La simplification

On va donner des algorithmes permettant de transformer une grammaire non contextuelle en une grammaire équivalente plus simple.

1. Élimination des symboles inutiles.
En deux étapes :
D'abord éliminer les variables d'où ne dérive aucun mot en symboles terminaux.
Pour cela, algorithme récursif n°1 :
- Les variables dont une production au moins ne contient que des terminaux sont utiles ;
- Les variables dont une production au moins ne contient que des terminaux et des symboles utiles sont utiles.

Cet algorithme se termine en fournissant les symboles à retenir.

Les autre sont inutiles, on les enlève. La grammaire ainsi obtenue est équivalente à la grammaire de départ.

Puis éliminer tous les symboles (terminaux ou non) n'appartenant à aucun métamot dérivé de l'axiome.
Pour cela, algorithme récursif n°2:
- L'axiome est utile ;
- Les symboles apparaissant dans les productions de symboles utiles sont utiles.

Les symboles non retenus sont inutiles, on les enlève et on a encore une grammaire équivalente.

L'enlèvement des symboles inutiles se fait en appliquant l'algorithme n°1 puis l'algorithme n°2. L'ordre a de l'importance.

Exemple : Partons de la grammaire G suivante :
$S \rightarrow AB \mid CA$ $A \rightarrow a$ $B \rightarrow AB \mid EA$
$C \rightarrow aB \mid b$ $D \rightarrow aC$ $E \rightarrow BA$

L'algorithme n°1 conserve A et C, puis S et D.

Il reste donc :
$S \rightarrow CA$ $A \rightarrow a$ $C \rightarrow b$ $D \rightarrow aC$

L'algorithme n°2 élimine D.

Il reste finalement :
$S \rightarrow CA$ $A \rightarrow a$ $C \rightarrow b$

2. Suppression des ε-productions

Les ε-productions ont comme inconvénient que, dans une dérivation, la longueur des métamots peut décroître. Pour des besoins d'analyse, il est préférable d'éviter cette situation. Si ε appartient à L(G), il n'est bien sûr pas possible d'éviter toutes les ε-productions.

On traite donc uniquement les langages non contextuels dont on a éventuellement enlevé le mot vide.

On cherche récursivement les variables annulables, celles d'où dérive le mot vide ; on part d'une grmmaire sans symbole inutile :

- Les variables qui se récrivent ε sont annulables ;
- Les variables dont une production au moins ne contient que des variables annulables sont annulables.

L'ensemble ANNUL(G) des variables annulables de G étant déterminé, on modifie les productions contenant des variables annulables.

On remplace dans une production $A \rightarrow \alpha$ les symboles annulables de α par le mot vide de toutes les manières possibles.

Enfin, on supprime les productions vides.

La grammaire ainsi obtenue est équivalente à la grammaire de départ (au mot vide près éventuellement).

Exemple : La grammaire suivante engendre les mots bien parenthésés :

$S \rightarrow S(S) \mid \varepsilon$

La grammaire

$S \rightarrow S(S) \mid (S) \mid S() \mid ()$

engendre les mêmes mot, sauf le mot vide.

Pour obtenir le mot vide, on tolère une seule production vide, sur l'axiome, et sans autoriser celui-ci à apparaître en partie droite de production. On rajoute donc un nouvel axiome, ici T :

$T \rightarrow \varepsilon \mid S \quad S \rightarrow S(S) \mid (S) \mid S() \mid ()$

3. Élimination des productions unitaires

Il s'agit des productions de la forme $A \to B$, où B est une variable.

On suppose que G n'a aucune variable inutile, ni production vide.

On cherche toutes les dérivations de la forme $A \overset{*}{\Rightarrow} B$. Cela se fait récursivement à partir des productions unitaires. Chaque fois qu'une telle dérivation est obtenue, on ajoute aux productions de A toutes les productions non unitaires de B. Enfin, on efface les productions unitaires.

La grammaire ainsi obtenue a peut-être des symboles inutiles, qu'on supprime. La grammaire finale est équivalent à la grammaire de départ, n'a pas de symboles inutiles, de productions vides ni de productions unitaires.

Exemple : Pour la grammaire précédente, on obtiendrait :

$T \to \varepsilon \mid S(S) \mid (S) \mid S() \mid ()$ $\quad$ $S \to S(S) \mid (S) \mid S() \mid ()$

4.3 La forme normale de Chomsky

On limite la forme des productions :

seules sont admises les formes $A \to BC$ et $A \to a$.

Une telle grammaire est une grammaire normale de Chomsky.

Théorème : Tout langage non contextuel sans le mot vide peut être engendré par une grammaire normale de Chomsky.

Démonstration :

1. Partir de G sans variable inutile ni ε-production ni production unitaire ;
2. Pour chaque terminal a, ajouter une variable C_a avec l'unique production $C_a \to a$;
3. Soit $A \to X_1X_2...X_m$ une production de G avec $m > 1$. Remplacer toutes les occurrences des terminaux par la variable associée au 2.

4. Les productions sont de la forme désirée ou bien de la forme $A \rightarrow B_1B_2...B_m$ où $m > 2$ et où les B_i sont des variables. Remplacer une telle production par $A \rightarrow B_1D_1$, $D_1 \rightarrow B_2D_2$, ...

Exemple : $S \rightarrow aB \mid bA \quad A \rightarrow a \mid aS \mid bAA \quad B \rightarrow b \mid bS \mid aBB$
devient :
$S \rightarrow C_aB \mid C_bA \quad C_a \rightarrow a \quad C_b \rightarrow b \quad A \rightarrow a \mid C_aS \mid C_bD \quad D \rightarrow AA$
$B \rightarrow b \mid C_bS \mid C_aE \quad E \rightarrow BB$

La forme normale de Chomsky a un intérêt essentiellement théorique.

4.4 La forme normale de Greibach

On limite cette fois les productions à la forme $A \rightarrow aB_1B_2...B_m$, où $m \geq 0$. Toutes les productions débutent par un terminal suivi de variables. La grammaire initiale de l'exemple précédent est déjà sous forme normale de Greibach.

Lemme : Si les productions de B sont $B \rightarrow \beta_1 \mid \beta_2 \mid ... \mid \beta_m$, on ne change pas le langage engendré en remplaçant une production $A \rightarrow \alpha_1B\alpha_2$ par $A \rightarrow \alpha_1\beta_1\alpha_2 \mid \alpha_1\beta_2\alpha_2 \mid ... \mid \alpha_1\beta_m\alpha_2$.

Démonstration : évident.

On indique maintenant comment éliminer la récursivité directe à gauche. Utile pour fabriquer des analyseurs syntaxiques par descente récursive.
Supposons aucune production vide ni unitaire.

Si A est une variable récursive à gauche, ses productions sont :
$A \to A\alpha_1 \mid A\alpha_2 \mid ... \mid A\alpha_r$ (productions récursives à gauche)
$A \to \beta_1 \mid \beta_2 \mid ... \mid \beta_s$, (productions non récursives)

On introduit une nouvelle variable B et on remplace les productions de A par :
$A \to \beta_1 \mid \beta_2 \mid ... \mid \beta_s$ $B \to \alpha_1 \mid \alpha_2 \mid ... \mid \alpha_r$
$A \to \beta_1 B \mid \beta_2 B \mid ... \mid \beta_s B$ $B \to \alpha_1 B \mid \alpha_2 B \mid ... \mid \alpha_r B$

Cela ne change pas le langage engendré.
Puisque les B ne sont produits qu'à partir des productions de A, il suffit de voir que les productions de A dans l'une comme dans l'autre des grammaires produisent in fine les mêmes mots. Peut-être des productions unitaires sont-elles apparues pour B. On peut vérifier que leur suppression n'introduit aucune récursivité directe à gauche.
Ceci démontre le théorème suivant.

Théorème (Élimination de la récursivité directe à gauche) :
Tout langage non contextuel sans le mot vide peut être engendré par une grammaire sans symbole inutile ni production vide ni production unitaire ni récursivité directe à gauche.

On peut maintenant énoncer et démontrer le théorème de la mise sous forme normale de Greibach.

Théorème : Tout langage non contextuel sans le mot vide peut être engendré par une grammaire sans symbole inutile dont toutes les productions sont de la forme $A \to aB_1B_2...B_m$, où $m \geq 0$.

Démonstration : On part d'une grammaire G sous forme normale de Chomsky.
On ordonne (arbitrairement) les variables : $A_1, A_2, ..., A_m$; c'est une bonne idée de supposer que la première est l'axiome.

Les productions considérées seront de trois types possibles :
- type 1 : $A_i \rightarrow a\alpha$, où $\alpha \in \{A_i + B_j\}^*$;
- type 2 : $A_i \rightarrow A_j\alpha$, où $\alpha \in \{A_i\}\{A_i + B_j\}^*$ et $j > i$;
- type 3 : $A_i \rightarrow A_j\alpha$, où $\alpha \in \{A_i\}\{A_i + B_j\}^*$ et $j \leq i$.

Au départ, les trois types sont seuls présents (Chomsky).
Dans un premier temps, on va éliminer les productions de type 3, quitte à introduire de nouvelles variables B_i et des productions de la forme $B_i \rightarrow A_j\beta$, où $\beta \in \{A_i + B_j\}^*$.
Supposons qu'on a pu éliminer les productions de type 3 pour les variables jusqu'à A_{i-1} en introduisant les B_j correspondantes.
On considère les productions de type 3 dans l'ordre croissant des indices j : $A_i \rightarrow A_1\alpha$ et on remplace A_1 par ses productions. On fait ainsi apparaître des productions des trois types, mais plus aucune de type 3 commençant par A_1. Puis on fait de même pour A_2, ... , A_{i-1}. Les seules productions de type 3 restantes sont directement récursives à gauche.

Les productions de A_i sont des types :
- type 1 : $A_i \rightarrow a\alpha$, où $\alpha \in \{A_i + B_j\}^*$;
- type 2 : $A_i \rightarrow A_j\alpha$, où $\alpha \in \{A_i\}\{A_i + B_j\}^*$ et $j > i$;
- type 3 : $A_i \rightarrow A_i\gamma$, où $\gamma \in \{A_i\}\{A_i + B_j\}^*$.

On élimine le type 3 par élimination de la récursivité directe à gauche : on introduit donc la variable B_i et on obtient des productions :
$A_i \rightarrow a\alpha \mid a\alpha B_i$ $\quad A_i \rightarrow A_j\alpha \mid A_j\alpha B_i$ $\quad B_i \rightarrow \gamma \mid \gamma B_i$.
Les productions restantes sont bien de types 1 ou 2, et les productions de B_i commencent bien par une variable A_j.

On considère maintenant les productions de type 2 par ordre décroissant des indices.
La dernière variable A_m ne peut en avoir. Dans les productions de A_{m-1} de type 2, seule apparaît A_m, qu'on remplace par ses productions et ainsi de suite jusqu'à A_1.

Maintenant, toutes les productions des A_i sont de type 1. Dans les productions des B_i, on remplace le premier symbole, qui est un A_i, par les productions de celui-ci.
La grammaire obtenue est sous forme normale de Greibach.

Remarques :

Cela accroît le nombre de productions de façon exponentielle.

Dans la pratique, il n'est pas indispensable que la grammaire soit sous forme normale de Chomsky, mais le déroulement de l'algorithme peut en être perturbé.

Exemple : $A_1 \to A_2A_3 \quad A_2 \to A_3A_1 \mid b \quad A_3 \to A_1A_2 \mid a$
Première étape : A_1 et A_2 sans changements ; pour A_3, garder $A_3 \to a$; puis $A_3 \to A_2A_3A_2$, puis $A_3 \to A_3A_1A_3A_2 \mid bA_3A_2$; les productions de A_3 sont donc : $A_3 \to A_3A_1A_3A_2$ et $A_3 \to bA_3A_2 \mid a$.
Élimination de la récursivité directe à gauche :
$A_3 \to bA_3A_2 \mid a \mid bA_3A_2B_3 \mid aB_3$ et $B_3 \to A_1A_3A_2 \mid A_1A_3A_2B_3$.
Comme prévu, les productions de A_3 sont de type 2. Utilisons-les pour récrire les productions de A_2 :
$A_2 \to bA_3A_2A_1 \mid aA_1 \mid bA_3A_2B_3A_1 \mid aB_3A_1 \mid b$.
Récrivons maintenant les productions de A_1 :
$A_1 \to bA_3A_2A_1A_3 \mid aA_1A_3 \mid bA_3A_2B_3A_1A_3 \mid aB_3A_1A_3 \mid bA_3$,
puis les productions de B_3 :
$B_3 \to bA_3A_2A_1A_3A_3A_2 \mid aA_1A_3A_3A_2 \mid bA_3A_2B_3A_1A_3A_3A_2 \mid$
$aB_3A_1A_3A_3A_2 \mid bA_3A_3A_2 \mid$
$bA_3A_2A_1A_3A_3A_2B_3 \mid aA_1A_3A_3A_2B_3 \mid bA_3A_2B_3A_1A_3A_3A_2B_3 \mid$
$aB_3A_1A_3A_3A_2B_3 \mid bA_3A_3A_2B_3$.

4.5 Les grammaires linéaires droites

C'est une grammaire dont les seules productions sont $A \to aB$ ou $A \to a$. Elles sont sous forme normale de Greibach.

Théorème : Un langage est engendré par une grammaire linéaire droite si et seulement s'il est régulier sans le mot vide.

Démonstration :

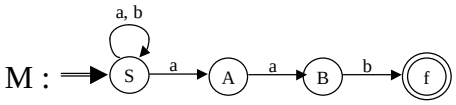
- Si $G = (V, T, P, S)$ est linéaire droite, soit $M = (V \cup \{f\}, T, \delta, S, \{f\})$ un automate (non déterministe) où :
  $\delta(A, a) = \{B : A \to aB\}$ et $\{B : A \to aB\} \cup \{f\}$ si $A \to a$.
  On montre par récurrence sur la longueur de w (écrire $w = ax$) que :
  $(f \in \delta(A, w)) \Leftrightarrow (A \overset{*}{\Rightarrow} w)$.
  Il s'ensuit que $L(G) = L(M)$.

- Réciproquement, soit $M = (V, T, \delta, q_0, F)$ un AFD. Soit $G = (V, T, P, q_0)$ la grammaire d'axiome q_0 dont les productions P sont :
  $q_i \to aq_j$ si $q_j = \delta(q_i, a) \notin F$ et $q_i \to aq_j \mid a$ si $q_j = \delta(q_i, a) \in F$.
  On montre le même résultat que précédemment et donc $L(M) = L(G)$.

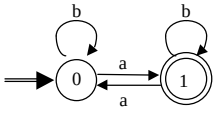
On peut remarquer que, dans ce cas, G est non ambiguë.

Exemples : G : $S \to aS \mid bS \mid aA$ $A \to aB$ $B \to b$

M :



Pour la réciproque : M :



G : $q_0 \to aq_1 \mid bq_0$ et $q_1 \to aq_0 \mid bq_1 \mid b$