MPI* Info

Forme normale de Chomsky pour une grammaire non contextuelle

TD16

PICQUET Augustine

1 Forme normale de Chomsky

Le résultat suivant permet de restreindre la forme des règles de dérivation utilisées dans une grammaire. D'un point de vue algorithmique c'est un point central car il permet de considérer les règles sous une forme uniquement et non pas aussi générales que la définition d'une grammaire ne l'autorise.

Théorème

Toute grammaire qui ne génère pas le mot vide est faiblement équivalente à une grammaire dont toutes les règles sont de la forme $X \to AB$ ou $X \to a$ où A et B sont des symboles non terminaux distincts du symbole initial et a est un symbole terminal. On appelle la grammaire dont les règles sont de cette forme, forme normale de Chomsky.

Voici le principe de la transformation (en 5 étapes)

- 1. Si nécessaire (si S apparaît dans un des termes de droite), on introduit un nouveau symbole non terminal S_0 qui sera le nouveau symbole initial et la règle $S_0 \to S$ où S est l'ancien symbole initial
- 2. Élimination des règles de la forme $A \to \varepsilon$; pour toute règle de cette forme, on va considérer toutes les règles où A apparaît dans le terme de droite. Chacune des ces règles sera de la forme $X \to \alpha A\beta$ et on ajoutera les règles $X \to \alpha\beta$. Attention, si on a plusieurs occurrences de A alors il faut ajouter des règles qui suppriment chaque combinaison possible de A. Par exemple, si on a $X \to aAbAcA$, on ajoutera $A \to abc, X \to aAbAc, X \to aAbc, X \to abAcA, X \to abAc, X \to aAbcA, X \to abAc$. En procédant ainsi, on peur créer de nouvelles règles de la forme que l'on est en train d'éliminer, dans ce cas on recommence en évitant de faire réapparaître des formes déjà éliminées jusqu'à ne plus avoir de règles du tout.
- 3. Élimination des règles de la forme $A \to B$: pour chacune des ces règles, on ajoute pour chaque règle de la forme $B \to \alpha$ la règle $A \to \alpha$. On répète sans recréer de règles déjà éliminées jusqu'à ne plus en avoir.
- 4. Remplacement des longues productions : pour chaque production dont le terme de droite est de longueur au moins 3 de la forme $X \to a_1..a_n$, on crée de nouveaux symboles non terminaux et des règles $X \to a_1N_1, N_1 \to a_2N_2, .., N_{n-2} \to a_{n-1}N_{n-1}, N_{n-1} \to a_n$.
- 5. Pour chaque non terminal a qui reste à droite d'une règle sans être seul, on introduit un nouveau terminal X_a qui le remplace et une règle $X_a \to a$

Appliquer cette méthode aux grammaires suivantes :

1. $S \to AbA; A \to Aa|\varepsilon$

- **Etape 1.** OK
- **Etape 2.** On a $A \to \varepsilon$, donc pour $S \to AbA$, ajoute $S \to b, S \to Ab, S \to bA$.
  Pour $A \to Aa$, on ajoute $A \to a$
  On a donc :

$$S \to AbA|b|Ab|bA$$

$$A \to a|Aa$$

- **Etape 3.** OK
- **Etape 4.** Pour $S \to AbA$, on ajoute $S \to AN_1$ et $N_1 \to bA$ On a donc.

$$S \to AN_1|b|Ab|bA|Aa$$

$$A \to a|Aa$$

$$N_1 \to bA$$

- **Etape 5.** On ajoute $X_b \to b$ et $X_a \to a$ Finalement, on a :

$$S \to AN_1|b|AX_b|X_bA$$

$$N_1 \to X_bX_a$$

$$X_b \to b$$

$$X_a \to a$$

$$A \to AX_a|a$$

2. $S \to aXbX; X \to aY|bY|\varepsilon; Y \to X|c$

— **Etape 1.** OK

— **Etape 2.** On a $X \to \varepsilon$, et $Y \to X$ et $S \to aXbX$. On ajoute donc $S \to abX|ab|aXb$ et $Y \to \varepsilon$ Maintenant il faut régler le "problème" $Y \to \varepsilon$, on a $X \to aY|bY$, on ajoute donc $X \to a|b$. On obtient donc :

$$S \to ab|aXb|abX|aXbX$$

$$X \to aY|bY|a|b$$

$$Y \to X|c$$

— **Etape 3.** Ajout de $Y \to a|b$.

$$S \to ab|aXb|abX|aXbX$$

$$X \to aY|bY|a|b$$

$$Y \to X|c|a|b$$

— **Etape 4.**

$$S \to ab|aN_1|aN_2|aN_3$$

$$X \to aY|bY|a|b$$

$$Y \to a|b|c$$

$$N_1 \to bX, N_2 \to Xb, N_3 \to XN_4, N_4 \to bX$$

— **Etape 5.** On ajoute $X_b \to b$ et $X_a \to a$ Finalement, on a :

$$S \to X_aX_b|X_BN_1|X_aN_2|X_aN_3$$

$$X \to X_aY|X_bY|a|b$$

$$Y \to a|b|c$$

$$N_1 \to X_bX, N_2 \to Xb, N_3 \to XN_4, N_4 \to X_bX$$

$$X_b \to b, X_a \to a$$

2 Quelques exercices supplémentaires.

1. On considère la grammaire suivante :

$$E \to E+T|T$$

$$T \to T*F|F$$

$$F \to a|(E)$$

Donner un arbre d'analyse pour ((a)) et pour (a+a)*a + a

1. CF photo!!

2. Soit $L = \{a^i b^j c^k | i = j \ ou \ j = k, (i,j,k) \in \mathbb{N}^3\}$. Donner une grammaire reconnaissant L. Est-elle ambiguë?

2.

$$S \to E|F$$

$$F \to aGbX|\varepsilon$$

$$G \to aGb|\varepsilon$$

$$X \to Xc|\varepsilon$$

$$E \to ZbHc|\varepsilon$$

$$H \to bHc|\varepsilon$$

$$Z \to Za|\varepsilon$$

Cette grammaire est ambiguë, en effet, "abc" peut être généré en commençant par $S \to E$ ou en commençant par $S \to F$, ce qui nous donne deux arbres de dérivation différents.

3. Donner une grammaire qui engendre l'ensemble des mots qui contiennent au moins 3 occurrences de 1 sur l'alphabet $\{0;1\}$.

3.

$$S \to A1A1A1A$$

$A \to 0|1|\varepsilon|0A|1A$

4. Sur le même alphabet, donner une grammaire qui génère les mots de longueur impaire ayant un 0 au milieu.

4. $S \to 0S0|0S1|1S0|1S1|101|100|001|000$

5. Encore sur le même alphabet, donner une grammaire qui engendre les mots ayant plus de 0 que de 1.

5. Si l'inégalité est large :

$$S \to SS|0S|S0|01S|10S|S10|S01|0S1|1S0|\varepsilon$$

Si l'inégalité est stricte :

$$S \to SS|0S|S0|01S|10S|S10|S01|0S1|1S0|0$$