Rappels de Cours sur les formalismes de description de langages

1 Définition formelle d'une grammaire

Une grammaire G est un quadruplet (V_t, V_{nt}, P, S) où :

— V_t est un ensemble fini de symboles terminaux (généralement notés en minuscules)
— V_{nt} est un ensemble fini de symboles non terminaux (généralement notés en majuscules)
— P est un ensemble de règles dites règles de production de la forme :
$\alpha \rightarrow \beta$,
α et β étant des mots définis sur $V_t \cup V_{nt}$,
c'est-à-dire α *et* $\beta \in (V_t \cup V_{nt})^\star$

2 Dérivation

Soit G une grammaire (V_t, V_{nt}, P, S), soit $\alpha \in (V_t \cup V_{nt})^\star$, soit $\beta \in (V_t \cup V_{nt})^\star$

La grammaire G permet de **dériver en une étape** le mot α en un mot β ce qui se note : $\alpha \Rightarrow_G \beta$ si et seulement si

— $\alpha = \delta_1\ \alpha'\ \delta_2, \quad \delta_i \in (V_t \cup V_{nt})^\star$
— $\beta = \delta_1\ \beta'\ \delta_2, \quad \delta_i \in (V_t \cup V_{nt})^\star$
— et s'il existe dans G une règle de production de la forme $\alpha' \rightarrow \beta'$

3 Extended Backus Normal Form : EBNF

— les terminaux se notent avec des minuscules : a, b, ...
— les non-terminaux se notent $<$ X , Y $>$, ...
— définition d'un non terminal : $<$ A $>$: := $<$ B $>$
— définition de la séquence : $<$ A $>$ $<$ B $>$
— définition du choix : $<$ A $>$ | $<$ B $>$
— définition de la répétition : $\{\alpha\}$
— définition de l'optionnel : $[\alpha\ ]$

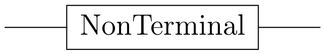
Remarque :
Si { , [... appartiennent au langage à décrire, on les fait précéder de \ pour éviter de les confondre avec l' { et l' [du métalangage EBNF.

4 Diagramme de Conway

C'est une manière graphique de décrire les langages

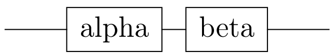
— les terminaux se notent :

— les non-terminaux se notent :

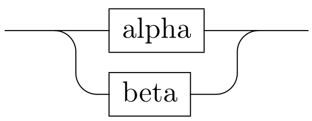


— définition d'un non terminal :
X

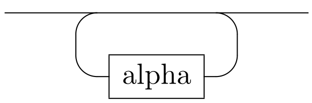
— définition de la séquence :



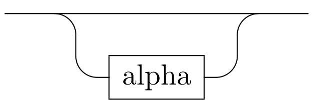
— définition du choix :



— définition de la répétition :



— définition de l'optionnel :



5 Transformation d'une grammaire en grammaire régulière droite

5.1 Substitution

1. Une règle avec un non-terminal A en membre gauche s'appelle une A-production
2. Soit la règle $Y \to \beta$ où β est une séquence de symboles contenant X. Remplacer le non-terminal X dans la règle $Y \to \beta$ consiste à :
   — retrouver toutes les X-productions c'est-à-dire les règles de la forme $X \to \alpha_1$; $X \to \alpha_2$; … ;$X \to \alpha_p$:
   — remplacer la règle $Y \to \beta$ par les régles $Y \to \ [\alpha_i \ /X \ ]\beta$ pour tout $i \in 1..p$

5.2 Factorisation

Soit une grammaire définie par un ensemble de règles de la forme suivante :

$$\begin{cases} X & \to \alpha A \\ X & \to \alpha B \end{cases}$$

Cet ensemble de règles peut être remplacé en factorisant par l'ensemble de règles suivant équivalent :

$$\begin{cases} X & \to \alpha Y \\ Y & \to A \\ Y & \to B \end{cases}$$

5.3 Récursivité gauche et élimination de la récursivité gauche

5.3.1 Définition de la récursivité gauche

Une A-production est récursive à gauche si et seulement si elle est de la forme $A \to A\,\alpha$ avec A terminal et α une suite de symboles de $(V_t \cup V_{nt})^{\star}$. Une grammaire peut aussi avoir des règles récursives gauches "non évidentes". Par exemple, les règles de production suivantes

$$\begin{cases} A & \to B \\ B & \to A\alpha \end{cases}$$

cachent une récursivité gauche "non évidente" : $A \to A\alpha$

5.3.2 Elimination de la récursivité gauche

Algorithme :

1. Repérer les règles récursives gauches "évidentes"
2. Trouver les règles récursives gauches non évidentes et les transformer (par substitution) en règles récursives gauches évidentes
3. Eliminer les non-terminaux inaccessibles
4. Regrouper les A-productions récursives gauches : $A \to A\alpha_1\ |A\alpha_2 ...\ |A\alpha_p$
5. Regrouper les A-productions NON récursives gauches (c'est-à-dire celles dont le membre droit ne commence pas par A) :$A \to \beta_1\ |\beta_2 ...\ |\beta_p$
6. Factoriser, ce qui permet d'obtenir des règles de la forme : $A \to A(\alpha_1\ |\alpha_2 ...\ |\alpha_p)\ |\beta_1\ |\beta_2 ...\ |\beta_p$ ce qui peut se réécrire sous la forme $A \to A\alpha\ |\beta$
7. Remplacer $A \to A\alpha\ |\beta$ par :

$$\begin{cases} A & \to \beta A' \\ A' & \to \alpha A' \\ A' & \to \Lambda \end{cases}$$

A' étant un nouveau non-terminal

5.4 Transformation d'une grammaire quelconque en une grammaire régulière droite factorisée

Algorithme :

1. Eliminer les récursivités gauches évidentes et "non évidentes" en appliquant l'algorithme précédent
2. Partir de l'axiome, et faire évoluer les règles de production en :
   — faisant des substitutions des NON-TERMINAUX EN TÊTE des membres droits des règles de production
   — en factorisant DES QUE POSSIBLE le terminal obtenu en tête des membres droits des règles
3. Refaire les mêmes actions de substitution et de factorisation pour les autres règles de production