TELECOM Nancy (1A) — Mathématiques Appliquées pour l'Informatique

Analyse syntaxique descendante, prétraitement des grammaires :
réduction, suppression de la récursivité à gauche, factorisation à gauche

Exercice 1

Réductions inférieure et supérieure des grammaires.

Présentation du problème

La *réduction* est une des étapes préalables à l'analyse des grammaires algébriques. Cette étape consiste à éliminer les non-terminaux et les règles inutiles de la grammaire donnée. La réduction est une transformation qui préserve le langage engendré par la grammaire. Dans la suite on considère une grammaire $G = (N,\ T,\ \rightarrow,\ S)$.

— La réduite inférieure d'une grammaire consiste à éliminer les non-terminaux X tels qu'il n'existe pas de dérivation de la forme $X \overset{*}{\rightarrowtail} w$ où $w \in T^*$, c'est-à-dire les non-terminaux ne produisant pas de mots de T^*.

On considère la suite $(E_i)_{i\geq 0}$ définie par récurrence par :

$$\begin{array}{lcl} E_0 & = & \emptyset \\ E_{i+1} & = & E_i \cup \{X,\ X \in N \textit{ et } (\exists \alpha \in (E_i \cup T)^*)\ (X \rightarrow \alpha \textit{ est une règle de } G)\} \end{array}$$

La suite E_i est croissante et les ensembles E_i sont tous finis (sous-ensembles de l'ensemble fini N), la suite E_i est donc stationnaire à partir d'un certain rang. Soit E la limite de cette suite, la réduction inférieure de la grammaire consiste à éliminer les règles comportant les éléments de $N \setminus E$.

— La réduite supérieure d'une grammaire consiste à éliminer les non-terminaux qui ne sont pas accessibles à partir de l'axiome de la grammaire. On considère la suite $(F_i)_{i\geq 0}$ définie par récurrence par :

$$\begin{array}{lcl} F_0 & = & \{S\} \\ F_{i+1} & = & F_i \cup \{X,\ X \in N \textit{ et } (\exists A \in F_i)\ (\exists \alpha, \beta \in (N \cup T)^*)\ \ (A \rightarrow \alpha X \beta \textit{ est une règle de } G)\} \end{array}$$

La suite F_i est stationnaire à partir d'un certain rang (pour les mêmes raisons que la suite E_i). Soit F la limite de la suite F_i, la réduction supérieure consiste à éliminer les règles de la grammaire comportant les éléments de $N \setminus F$.

La réduction d'une grammaire consiste à effectuer dans l'ordre une réduction inférieure suivie d'une réduction supérieure.

Questions

Soit la grammaire G définie par les règles suivantes :

$A \rightarrow AT \mid T \mid U$
$T \rightarrow bT \mid aCX \mid \varepsilon$
$U \rightarrow bU \mid abW \mid \varepsilon$
$B \rightarrow abU \mid bW \mid a$
$C \rightarrow bT \mid UC$
$W \rightarrow bUX \mid TUC$
$X \rightarrow aXU \mid bTYZ$
$Y \rightarrow aYb \mid aW$
$Z \rightarrow XU \mid aZY$

1. Déterminer les ensembles respectifs N et T des non-terminaux et des terminaux, sachant que les non-terminaux sont des majuscules et les terminaux des minuscules, on suppose que l'axiome est A.
2. Effectuer une réduction inférieure de la grammaire G.
3. Effectuer une réduction supérieure de la grammaire obtenue précédemment et en déduire la grammaire réduite de G.

Exercice 2

Élimination de la récursivité à gauche immédiate.

La récursivité à gauche est une caractéristique des grammaires qui empêche la construction d'un analyseur LL(1). Dans cet exercice on traite l'élimination de *la récursivité à gauche immédiate*, on pourra consulter le cours pour la suppression de la récursivité à gauche dans des cas plus généraux.

On dit qu'une grammaire est *immédiatement récursive à gauche* s'il existe un non-terminal A et une règle de la forme $A \rightarrow A\alpha$. Soit une grammaire comportant les règles $A \rightarrow A\alpha_1 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \ldots \mid \beta_n$.

1. Que peut-on dire des règles précédentes ? Donner en fonction de $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_n$, le langage engendré par A.

2. En introduisant un nouveau non-terminal A', déterminer des règles engendrant le même langage que dans la question précédente et ne comportant plus de récursivité à gauche.
3. Application : soit la grammaire G définissant des expressions arithmétiques,
   $G = (\{E,\ T,\ F,\ P\},\ \{a,\ b,\ c,\ +,\ *,\ -,\ /,\ (,\ )\},\ \rightarrow,\ E)$ où les règles sont :
   $E \rightarrow E + T \mid E - T \mid T \qquad T \rightarrow T * F \mid T/F \mid F \qquad F \rightarrow P \mid -P \qquad P \rightarrow a \mid b \mid c \mid (E).$
   Pour quelles raisons G est-elle *immédiatement récursive à gauche* ? En utilisant la transformation de la question précédente, donner une grammaire G' non récursive à gauche et équivalente à G.

Exercice 3

Factorisation à gauche des grammaires.

La factorisation des grammaires est une opération préalable à l'analyse LL(1) proprement dite. Étant donné le non-terminal A à réécrire et le caractère courant a lu, un analyseur syntaxique LL(1) doit pouvoir déterminer quelle règle utiliser, si l'on dispose par exemple des règles $A \rightarrow abcdB$ et $A \rightarrow abceC$, on ne peut décider en lisant un caractère à l'avance quelle règle choisir pour réécrire le non-terminal A. La factorisation permet de résoudre ce problème, dans ce cas on remplace les deux règles précédentes par les règles $A \rightarrow abcA' \qquad A' \rightarrow dB \mid eC$.

Un algorithme de factorisation est le suivant :

Algorithme de factorisation à gauche

Pour chaque non terminal A

(a) Trouver le plus long préfixe commun α à deux ou plus de deux membres droits de règles dont le membre gauche est A ;
si $\alpha \neq \varepsilon$, remplacer les règles $A \rightarrow \alpha\beta_1 \mid \ldots \mid \alpha\beta_n \mid \gamma_1 \mid \ldots \mid \gamma_p$ (où α n'est pas préfixe de γ_i) par les règles $A \rightarrow \alpha A' \mid \gamma_1 \mid \ldots \mid \gamma_p$ et $A' \rightarrow \beta_1 \mid \ldots \mid \beta_n$;
(b) Recommencer (a) jusqu'à ce qu'il n'y ait plus de factorisation à effectuer ;

finPour

Soit la grammaire $G = (\{X,\ Y,\ Z\},\ \{a,\ b,\ c,\ d\},\ \rightarrow,\ X)$ telle que
$X \rightarrow aYbX \mid aYbXdZ \mid a$
$Y \rightarrow bcZ \mid bca$
$Z \rightarrow cd.$
Factoriser à gauche la grammaire G.