ESIAL 1 — Module Mathématiques appliquées discrètes

Analyse syntaxique descendante : suppression de la récursivité à gauche, factorisation à gauche, construction de tables LL(1), analyseur prédictif non récursif

Exercice 1 *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). Ici on traite l'élimination de *la récursivité à gauche immédiate*. On pourra consulter le cours pour la suppression de la récursivité à gauche en général.

On dit qu'une grammaire est *immédiatement récursive à gauche* s'il existe un non-terminal A et une règle de la forme $A \to A\alpha$.

Soit une grammaire comportant les règles $A \to A\alpha_1 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \ldots \mid \beta_n$.

1. Que peut-on dire des règles précédentes ? Donner le langage engendré par A.
2. En introduisant un nouveau non-terminal A', déterminer des règles engendrant le même langage et ne comportant plus de récursivité à gauche.
3. Soit la grammaire : $G = (\{E,\ T,\ F,\ P\},\ \{a,\ b,\ c,\ +,\ *,\ -,\ /,\ (,\ )\},\ \to,\ E)$
   $E \to E + T \mid E - T \mid T$
   $T \to T * F \mid T/F \mid F$
   $F \to P \mid -P$
   $P \to a \mid b \mid c \mid (E)$.
   Pour quelles raisons G est-elle *immédiatement récursive à gauche* ?
   Donner une grammaire G' non récursive à gauche et équivalente à G.

Exercice 2 *Factorisation à gauche des grammaires* La factorisation des grammaires est une opération préalable à l'analyse LL(1). Étant donné le non-terminal A à réécrire et le caractère a lu, un analyseur syntaxique LL(1) doit pouvoir déterminer quelle règle utiliser. Si l'on dispose par exemple des règles :
$A \to abcdB$
$A \to abceC$,
on ne peut décider en lisant un caractère à l'avance quelle règle choisir pour réécrire le non-terminal A. Dans ce cas, la factorisation consiste à remplacer ces deux règles par
$A \to abcA'$
$A' \to dB \mid eC$.

Algorithme 1 Algorithme de factorisation

```
pour tout non-terminal A faire
    répéter
        Trouver le plus long préfixe commun α ∈ (N ∪ T)^+ à deux ou plus de deux membres
        droits de règles dont le membre gauche est A ;
        remplacer les règles A → αβ_1 |...| αβ_n | γ_1 |...| γ_p par les règles
        A → αA' | γ_1 |...| γ_p  et A' → β_1 |...| β_n ;
    jusqu'à épuisement des factorisation à effectuer
fin pour
```

Soit la grammaire :
$G = (\{X,\ Y,\ Z\},\ \{a,\ b,\ c,\ d\},\ \to,\ X)$
$X \to aYbX \mid aYbXdZ \mid a$
$Y \to bcZ \mid bca$
$Z \to cd$.

Factoriser à gauche la grammaire G.

Exercice 3 *Construction d'une table LL(1) et analyseur prédictif non récursif*

Soit $G = (N, T, \rightarrow, S)$ une grammaire. La table M d'analyse syntaxique de G est une table à deux entrées : $M : N \times (T \cup \{\$\}) \rightarrow \mathcal{R}$ où $\mathcal{R}$ est l'ensemble des règles de G.

Construction de la table M. On suppose que les symboles directeurs SD ont été calculés.

Algorithme 2 Algorithme de construction de la table

```
pour tout règle r : A → α de G faire
  pour tout symbole directeur a ∈ SD(r) faire
    si M[A, a] vide alors
      M[A, a] := r
    sinon
      "erreur : la grammaire G n'est pas LL(1)"
    finsi
  fin pour
fin pour
```

L'analyseur prédictif comporte un tampon d'entrée, une pile, une table d'analyse et un flot de sortie. Le tampon d'entrée contient le mot à analyser suivi du caractère \$ (marqueur de fin) qui est aussi utilisé pour marquer le fond de la pile. X dénote le symbole en sommet de pile et a le symbole d'entrée courant.

Algorithme 3 Algorithme de l'analyseur prédictif

Entrées: la pile est initialisée à $\$S$, où S est l'axiome de la grammaire ;
a est la première lettre du mot à utiliser

Sorties: *succès* ou *erreur* est vrai

```
succès := faux ; erreur := faux
répéter
  si X == a == $ alors
    succès := vrai
  sinon si X == a et X ≠ $ alors
    dépiler(X) ;
    lecture(a) ;                    on dépile X et on avance sur le caractère suivant
  sinon si X ∈ N et M[X, a] ≠ ∅ alors
    // M[X, a] contient X → α1 ... αn
    afficher(X → α1 ... αn) ;
    dépiler(X) ;
    empiler(αn) ;
    ... ;
    empiler(α1) ;
  sinon
    erreur := vrai ;
  finsi
jusqu'à succès ou erreur
```

Soit la grammaire :
$G = (\{S,\ T,\ U\},\ \{a,\ b,\ (,\ ),\ ;\},\ \rightarrow,\ S)$
$S \rightarrow a \mid b \mid (T)$
$T \rightarrow SU$
$U \rightarrow ; SU \mid \varepsilon.$

1. Faire une analyse LL(1) de G.
2. Construire la table LL(1) de G.
3. Analyser les mots suivant :
   $(b; a; b)$, $((a); b)$, $(a); b$, $(a; ba$

TD 10
MaD
1

Avant de faire une analyse LL(1)
- grammaire non ambiguë (?)
- grammaire non récursive gauche (ex 1)
- grammaire factorisée (ex 2).

Exercice 1 :

récursivité gauche $\exists A \in N \quad A \xrightarrow{+} A\alpha \quad (A \rightarrow \dots \rightarrow A\alpha)$.
récursivité gauche immédiate $\quad A \rightarrow A\alpha$

1) Elles produisent des récursivités gauches immédiates.

$A \rightarrow \beta_1$
$\vdots$
$A \rightarrow \beta_m$
$A \rightarrow A\alpha_1 \rightarrow \beta_1\alpha_1$
$\vdots$
$A \rightarrow A\alpha_4 \rightarrow A\alpha_2\alpha_4 \rightarrow \beta_2\alpha_4\alpha_4$.
$\vdots$
$A \xrightarrow{*} \{(\beta_1, \dots \beta_m)\}\{\alpha_1, \dots, \alpha_m\}^*$.

2) $A \rightarrow \beta_1 A' \mid \dots \mid \beta_m A'$
$A' \rightarrow \varepsilon \mid \alpha_1 A' \mid \dots \mid \alpha_m A'$
On a supprimé la récursivité à gauche immédiate mais A' est non-terminal.

3) Les règles suivantes produisent de la récursivité immédiate à gauche :
$E \rightarrow E + T \mid E - T \mid T$
$T \rightarrow T \times F \mid T / F \mid F$

Transformation : $E \rightarrow E + T \mid E - T \mid T \qquad \{T\}\{+T, -T\}^*$
$E \rightarrow TE'$ et $E' \rightarrow \varepsilon \mid +TE' \mid -TE'$

$T \rightarrow T \times F \mid T/F \mid F \qquad \{F\}\{\times F, /F\}^*$
$T \rightarrow FT'$ et $T' \rightarrow \varepsilon \mid \times FT' \mid /FT'$.

Exercice 2 :

factorisat° :

$A \to abxd\,B$

$A \to abxc\,C$

On factorise A, car on ne sait lequel choisir au dép

$A \to abx\,A'$

$A' \to dB \mid cC$

On factorise :

$X \to aYbXX' \mid a$

$X' \to \varepsilon \mid dZ$

$Y \to bcY'$

$Y' \to a \mid Z$

$Z \to CD$

$X \to aX''$

$X'' \to YbXX' \mid \varepsilon$

$X' \to \varepsilon \mid dZ$

$Y \to bcY'$

$Y' \to a \mid Z$

$Z \to CD.$

Exercice 3 :

1) $S \to a \mid b \mid (T)$

$T \to SU$

$U \to ;SU \mid \varepsilon$

suivant(T) = }) }

ici, suivant(U) donne suivant(T)

si $U = \varepsilon$ alors suivant(S) donne suivant(T)

sinon suivant(S) donne premier(U)

TD 10

MaD 2

① Calcul des symboles directeurs des règles de G (voir TD9).

ⓐ Vide = $\{\}$ $\left(\begin{array}{l} \text{S/T ne produisent pas le vide} \\ \text{car } S \to a,\ S \to b,\ S \to (T) \\ \text{et } T \to SU \text{ et } S \to \varepsilon. \end{array}\right)$

ⓑ

Premier (S)	Premier (T)	Premier (U)	
$\emptyset$	$\emptyset$	$\emptyset$	← initialisation.
$\{a, b, (\}$	$\{a, b, (\}$	$\{;\}$	

ⓒ

Suivant (S)	Suivant (T)	Suivant (U)	
$\{\$\}$	$\emptyset$	$\emptyset$	← initialisat°
$\{\$, ;,)\}$	$\{)\}$	$\{)\}$	

$\$$: caractère de fin.

SD $(S \to a) = \{a\}$ SD $(T \to SU) = \{a, b, (\}$

SD $(S \to b) = \{b\}$ SD $(U \to ;SU) = \{;\}$

SD $(S \to (T)) = \{(\}$ SD $(U \to \varepsilon) = \{)\}$

rappel : SD $(A \to \alpha) = \begin{cases} \text{Premier}(\alpha) & \text{si non } (\alpha \xrightarrow{*} \varepsilon) \\ \text{Premier}(\alpha) \cup \text{Suivant}(A) & \text{sinon.} \end{cases}$

② Construction de la table.

M \ resultat SDI	a	b	(	)	;	$
S	S→a	S→b	S→(T)			
T	T→SU	T→SU	T→SU			
U				U→ε	U→;SU	

il ne doit pas y avoir de règles différentes dans une même case, sinon elle ne serait pas LL(1).

③ Analyseur prédictif non récursif.

PILE	Entrée	Sortie
$ S	(b; a; b) $	S→(T)
$) T (← ce S est l'axiome de la grammaire	(b; a; b) $	
	b; a; b) $	T→SU
$) T		
$) U S	b; a; b) $	S→b
$) U b	b; a; b) $	
$) U	; a; b) $	U→; SU
$) U S ;	; a; b) $	
$) U S	a; b) $	S→a
$) U a	a; b) $	
$) U	; b) $	U→; SU
$) U S ;	; b) $	
$) U S	b) $	S→b
$) U b	b) $	
$) U	) $	U→ε
$)	) $	
$	$	Succès