Théorie des langages et compilation
TD 3

Exercice 1

Soit la grammaire suivante pour définir les expressions bien parenthésées :

$$\begin{array}{lcl} S & \rightarrow & B \ \$ \\ B & \rightarrow & \epsilon \quad | \quad B \ (\ B \ ) \end{array}$$

1. Déterminer les non terminaux qui peuvent se dériver en ϵ.
2. Calculer *premier* et *suivant* pour chaque non terminal.
3. Calculer *directeur* de chaque règle.
4. Cette grammaire est-elle LL(1) ? Justifier.

Soit maintenant la grammaire :

$$\begin{array}{lcl} S & \rightarrow & B \ \$ \\ B & \rightarrow & \epsilon \quad | \quad (\ B \ ) \ B \end{array}$$

5. Déterminer les non terminaux qui peuvent se dériver en ϵ.
6. Calculer *premier* et *suivant* pour chaque non terminal.
7. Calculer *directeur* de chaque règle.
8. Cette grammaire est-elle LL(1) ? Justifier.
9. Dérouler l'algorithme de parsing LL(1) et donner les dérivations correspondantes sur les chaines suivantes :
   (a) (( ) ( )) $
   (b) ( )) ($

Eléments de solution de l'exercice 1

1. Déterminer les non terminaux qui peuvent se dériver en ϵ. :
   B.
2. Calculer *premier* et *suivant* pour chaque non terminal.
   $premier(S) = premier(B) \cup premier(\$)$ car B se dérive en ϵ. $premier(B) = \{\epsilon\} \cup premier(B) \cup \{(\}$ donc $premier(B) = \{\epsilon, (\}$
   $suivant(S) = \emptyset$
   $suivant(B) = \{\$, (,)\}$
3. Calculer *directeur* de chaque règle. $dir(S \rightarrow B\$) = (premier(B) - \epsilon) \cup premier(\$) = \{(, \$\}$
   $dir(B \rightarrow \epsilon) = suivant(B) = \{\$, (,)\}$
   $dir(B \rightarrow B(B)) = premier(B) - \epsilon \cup \{(\} = \{(\}$
4. Cette grammaire est-elle LL(1) ? Justifier.
   Non elle n'est pas LL(1) car $dir(B \rightarrow \epsilon) \cap dir(B \rightarrow B(B)) = \{(\}$
5. Déterminer les non terminaux qui peuvent se dériver en ϵ.
   B

6. Calculer *premier* et *suivant* pour chaque non terminal.
   $premier(S) = premier(B) \cup premier(\$)$ car B se dérive en ϵ. $premier(B) = \{\epsilon\} \cup \{(\}$ donc $premier(B) = \{\epsilon, (\}$
   $suivant(S) = \emptyset$
   $suivant(B) = \{\$,)\}$
7. Calculer *directeur* de chaque règle.
   $dir(S \to B\$) = (premier(B) - \epsilon) \cup premier(\$) = \{(, \$\}$
   $dir(B \to \epsilon) = suivant(B) = \{\$,)\}$
   $dir(B \to (B)B) = premier(() = \{(\}$

8. Cette grammaire est-elle LL(1) ? Justifier.
   Oui La grammaire est LL(1) car $dir(B \to \epsilon) \cap dir(B \to (B)B) = \emptyset$
9. Dérouler l'algorithme de parsing LL(1) et donner les dérivations correspondantes sur les chaines suivantes :

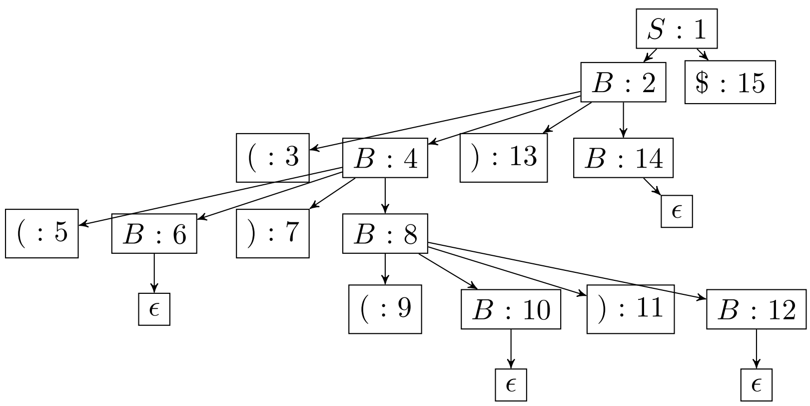
```
Procedure S()
    switch carCour do
        case ∈ directeur(S → B) do
            B() ;
            consommer($) ;
        otherwise do
            ERREUR;
    return ;
```

```
Procedure B()
    switch carCour do
        case ∈ directeur(B → ε) do
            ;
        case ∈ directeur(B → (B)B) do
            consommer(() ;
            B() ;
            consommer()) ;
            B() ;
        otherwise do
            ERREUR ;
    return ;
```

```
Procedure consommer(car)
    if carCour ≠ car then
        ERREUR ;
    carCour = suivant () ;
    return ;

Procedure Analyser(w)
    carCour = premierCar(w) ;
    S() ;
    return ;
```

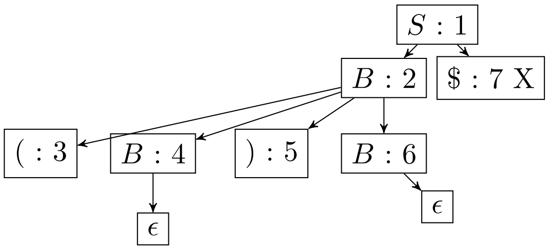
(a) (( ) ( )) $



1 carCour='$(_1$' ∈ directeur($S \to B\$$)
2 carCour='$(_1$' ∈ directeur($B \to (B)B$)
3 consommation de $(_1$: carCour=$(_2$
4 carCour='$(_2$' ∈ directeur($B \to (B)B$)
5 consommation de $(_2$: carCour=$)_3$
6 carCour='$)_3$' ∈ directeur($B \to \epsilon$)
7 consommation de $)_3$: carCour=$(_4$
8 carCour='$(_4$' ∈ directeur($B \to (B)B$)
9 consommation de $(_4$: carCour=$)_5$
10 carCour='$)_5$' ∈ directeur($B \to \epsilon$)
11 consommation de $)_5$: carCour=$)_6$

12 carCour=')$_6$' $\in$ directeur($B \to \epsilon$)
13 consommation de)$_6$: carCour=\$
14 carCour='\$' $\in$ directeur($B \to \epsilon$)
15 consommation de \$ et fin de l'analyse.

(b) ( )) (\$



1 carCour='($_1$' $\in$ directeur($S \to B\$$)
2 carCour='($_1$' $\in$ directeur($B \to (B)B$)
3 consommation de ($_1$: carCour=)$_2$
4 carCour=')$_2$' $\in$ directeur($B \to \epsilon$)
5 consommation de)$_2$: carCour=)$_3$
6 carCour=')$_3$' $\in$ directeur($B \to \epsilon$)
7 consommation de \$: échec car carCour=)$_3$ et fin de l'analyse avec un échec X.

Exercice 2

1. Proposer une grammaire LL(1) pour des expressions qui consistent en : des variables, l'addition binaire infixée, la multiplication binaire infixée (avec les règles de priorité usuelles) et les parenthèses.
2. Justifier pourquoi votre grammaire est LL(1).
3. Dérouler l'algorithme de parsing LL(1) et donner les dérivations correspondantes sur les chaines suivantes :
   (a) $id \; + \; id \; * \; (\; id \; + \; id \;) \; \$$
   (b) $id \; * \; id \; * \; id \; \$$
   (c) $(\; (\; id \;) \;) \; \$$

Exercice 3

Soit la grammaire suivante pour définir les chiffres romains (les unités) :

$$\begin{array}{lcllllll} S & \to & N\,\$ & & & & & \\ N & \to & i\,A & | & I_3 & | & v\,I_3 & | \; \epsilon \\ A & \to & v & | & x & & & \\ I_3 & \to & I_2\,i & | & \epsilon & & & \\ I_2 & \to & I_1\,i & | & \epsilon & & & \\ I_1 & \to & i & | & \epsilon & & & \end{array}$$

1. Cette grammaire est-elle ambigüe ? Proposer une grammaire non ambigüe G' équivalente ?
2. G' est-elle LL(1) ?
3. Proposer une grammaire LL(1) équivalente.

Eléments de solution de l'exercice 3

1. Cette grammaire est-elle ambigüe ? Proposer une grammaire non ambigüe G' équivalente ?
   Cette grammaire est ambigüe, en effet, le mot vide ϵ peut être dérivé par $N \to \epsilon$ ou bien par $N \to I_3 \to \epsilon$ Sinon, pour le mot i, la seule dérivation possible est : $N \to I_3 \to I_2 i \to \epsilon i = i$ et il n'y a pas d'autres dérivations, pour ii, la seule dérivation est $N \to I_3 \to I_2 i \to I_1 ii \to \epsilon ii = ii$, pour iii, la seule dérivation est $N \to I_3 \to I_2 i \to I_1 ii \to iii$, pour iv, la seule dérivation est $N \to iA \to iv$. Pour v, la seule dérivation est $N \to vI_3 \to v\epsilon = v$, pour vi, la seule dérivation possible est $N \to vI_3 \to vI_2 i \to v\epsilon i = vi$ et il n'y a pas d'autres dérivations, pour vii, la seule dérivation est $N \to vI_3 \to vI_2 i \to vI_1 ii \to v\epsilon ii = vii$, pour $viii$, la seule dérivation est $N \to vI_3 \to vI_2 i \to vI_1 ii \to viii$, pour ix, la seule dérivation est $N \to iA \to ix$. Pour x, la seule dérivation est $N \to vI_3 \to v\epsilon = v$.
   Pour enlever l'ambiguité, il suffit d'enlever ou bien la règle $N \to \epsilon$ ou bien $I_3 \to \epsilon$.
2. G' est-elle LL(1) ? Non la grammaire n'est pas LL(1), en effet $directeur(N \to iA) = \{i\}$ et $directeur(N \to I_3) = \{i, \$, \}$.
3. Proposer une grammaire LL(1) équivalente. Soit la grammaire suivante modifiée pour définir les chiffres romains (les unités) :

$$
\begin{array}{lcl}
S & \to & N\,\$ \\
N & \to & i\,T \quad | \quad v\,I_3 \\
T & \to & A \quad | \quad I_2 \\
A & \to & v \quad | \quad x \\
I_3 & \to & i\,I_2 \quad | \quad \epsilon \\
I_2 & \to & i\,I_1 \quad | \quad \epsilon \\
I_1 & \to & i \quad | \quad \epsilon
\end{array}
$$