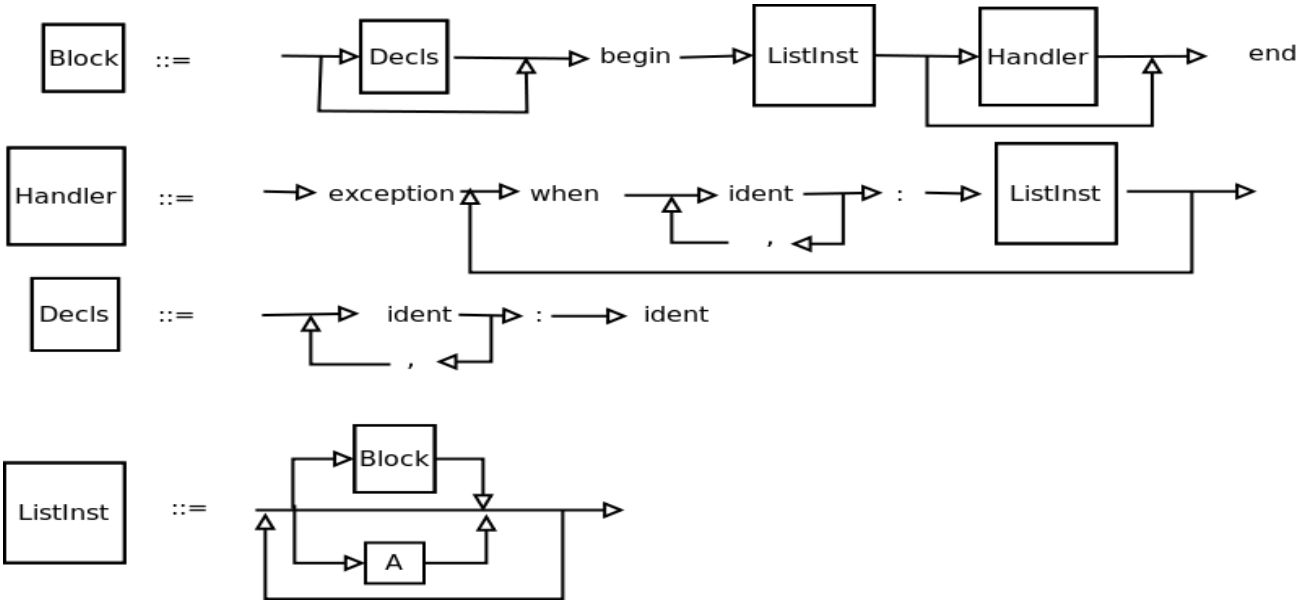
Polytech Paris-Sud ET4 Informatique Compilation 2020-2021

Grammaires et Analyses Lexicale et Syntaxique

1. Donnez une grammaire algébrique **non-ambiguë** pour le langage spécifié par les chartes syntaxiques ci-dessous :



Les non-terminaux sont dans des boites, les symboles terminaux apparaissent directement. Le non-terminal `A` représente une instruction différente d'un bloc ; on ne le détaille pas ici. L'axiome est `Block`.

2. Donnez une grammaire non-ambiguë pour une version restreinte des expressions du langage C. On considérera les opérateurs suivants, donnés par **priorités croissantes** :

= (binaire, associatif à droite)
+, - (binaires, associatifs à gauche)
*, / (binaires, associatifs à gauche)
* (unaire, déréférençage de pointeurs)

La grammaire doit permettre de manipuler des identificateurs, des constantes et des accès à des tableaux à une dimension (exemple : `*(p+1) = a[x*(2+y)][z+3] = **z*2`). On rappelle que tous les aspects d'un langage ne peuvent pas être exprimés syntaxiquement mais que si une vérification peut être faite au niveau syntaxique, il vaut mieux le faire à ce moment.

3. Complétez l'exercice précédent de manière à prendre en :

les opérateurs d'égalité et de non-égalité (`==` et `!=`)
les opérateurs de comparaison (`<`, `<=`, `>`, `>=`)
les connecteurs logiques (`&&`, `||` et `!`)
les opérateurs `+` et – unaires

en respectant les priorités et les associativités du langage C.

Question subsidiaire : ajoutez les appels de fonctions, selon la syntaxe du langage C. Ajouter les opérateurs préfixes et suffixes ++.

4. Une grammaire algébrique est dite « ε-libre » si

1. elle est sans symbole inaccessible ou improductif.
2. elle n'a pas de production de la forme X ::= ε sauf éventuellement pour l'axiome et dans ce cas l'axiome n'apparaît pas en partie droite de production,

Suivant le principe des algorithmes du polycopié pour détecter les symboles inaccessibles ou improductifs, donnez un algorithme qui calcule l'ensemble des terminaux qui peuvent dériver en une ou plusieurs étapes.

Donnez un algorithme pour calculer une grammaire « ε-libre » équivalente à une grammaire donnée.

Donnez une grammaire« ε-libre » équivalente à la grammaire suivante (les terminaux sont en minuscules, les non-terminaux en majuscules, l'axiome est S). On ne se préoccupera pas du langage engendré, ni du fait que la grammaire soit ambiguë.

```
S ::= a X Y | X X
X ::= Y Z | S | ε
Z ::= b Z T | ε
Y ::= a Y | X Z
T ::= b T | b
U ::= b V | ε
V ::= a X | ε
```

5. On considère le langage des formules logiques propositionnelles construites à l'aide de propositions atomiques, symbolisées dans la suite par l'unité lexicale p, les connecteurs binaires $\wedge$ (et logique), $\vee$ (ou logique), $\Rightarrow$ (implication), $\Leftrightarrow$ (équivalence) et le connecteur unaire $\neg$ (négation). Un tel langage peut être décrit par la grammaire suivante:

$$F ::= p \mid (F) \mid F \wedge F \mid F \vee F \mid F \Rightarrow F \mid F \Leftrightarrow F \mid \neg F$$

1. Donnez l'arbre syntaxique de la formule $(p \wedge q) \Rightarrow \neg r$, où p, q et r sont des instances de P.
2. Montrez à l'aide d'un exemple que la grammaire est ambiguë. Donnez une grammaire non ambiguë qui décrive le même langage, mais en imposant les précédences suivantes aux opérateurs (un "<" indique que le symbole gauche est moins prioritaire que celui de droite, un "=" qu'ils ont même priorité). On supposera les opérateurs binaires associatifs à gauche :
$$\Leftrightarrow < \Rightarrow < \wedge = \vee < \neg$$
3. Quel est l'arbre syntaxique associé au mot $p \wedge q \vee g \Rightarrow r \wedge p$?
4. Donnez une grammaire non ambiguë qui respecte les précédences ci-dessus mais qui interdisent les mélanges non parenthésés de $\wedge$ et de $\vee$, ainsi que toute associativité à $\Rightarrow$ et $\Leftrightarrow$. Les opérateurs $\wedge$ et $\vee$ restent associatifs à gauche. Ainsi les mots suivants devront être illégaux :
$$p \wedge q \vee r \wedge g,\ p \Rightarrow q \Rightarrow r,\ p \Leftrightarrow q \Leftrightarrow r$$

Par contre les mots suivants sont légaux:

$$p \wedge (q \vee r) \wedge g,\ p \vee q \vee r,\ p \Rightarrow (q \Rightarrow r),\ (p \Leftrightarrow q) \Leftrightarrow r$$