Université Bordeaux I **Master Informatique 2014-2015**
J1IN7M13EX Compilation, Examen
Lionel Clément **Jeudi 24 juin 2015 – de 14h00 à 15h30**

Ce sujet comporte 3 pages – Documents autorisés

Exercice 1 : 5 points

Une expression de type est définie inductivement de la sorte :
— Un type de base : *integer*, *real*, *character*, *boolean*, etc. est une expression de type ;
— Une variable de type $'x$ est une expression de type ;
— Un symbole α est une expression de type ;
— Si T_1 et T_2 sont deux expressions de type, $(T_1 \times T_2)$ est une expression de type ;
— Si T_1 et T_2 sont deux expressions de type, $(T_1 \to T_2)$ est une expression de type ;
— Si $T_1, T_2, \ldots T_k$, sont des expressions de type, $C_k(T_1, T_2, \ldots, T_k)$ est une expression de type, où C_k est un constructeur pour un type d'arité k.

Dans le code[1] suivant où `^T` représente un pointeur sur un type `T` et `x^` la valeur pointée par `x`.

```
1  struct Node {
2     info: 'x;
3     left: ^Node;
4     right: ^Node;
5  }
6
7  function binarySearch(node: ^Node, info: 'y): ^Node;
8  {
9     if (Node^.info < info)
10       return binarySearch(node^.left);
11    else if (Node^.info > info)
12       return binarySearch(node^.right);
13    else
14       return node;
15 }
```

1. Ce programme ne prévoit pas le cas de l'arbre vide, ou lorsque l'élément cherché dans l'arbre n'est pas présent.

Questions

1. Écrire l'expression de type de la structure `Node` en utilisant la définition.
2. L'opérateur $<$ est-il polymorphe ? Pourquoi ?
3. Est-ce qu'on peut connaître a priori la taille de la structure `Node` ? Pourquoi ?
4. Quelle est la condition pour que le typage statique soit garanti avec ce langage de programmation ?

Exercice 2 : 5 points

Soit G une grammaire algébrique $(\{E\}, \{P, M, L, R, N\}, E, R)$ dont les règles de production R sont les suivantes :

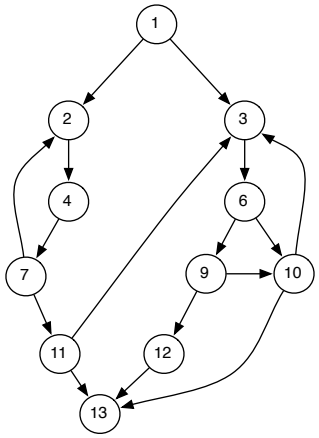
E $\rightarrow$ E P E
E $\rightarrow$ E M E
E $\rightarrow$ L E R
E $\rightarrow$ N

Questions

1. Décrire l'analyse Earley de la séquence $LNPNRMN$
2. Écrire une séquence qui produit deux analyses distinctes. Montrer comment l'analyseur Earley permet de produire ces deux analyses.
3. Est-il souhaitable que certaines séquences produisent plusieurs analyses différentes pour un compilateur ? Pourquoi ? Proposez une solution décrite dans le cours pour éviter d'avoir 2 analyses pour la même séquence avec cette grammaire.

Exercice 3 : 5 points

Soit le graphe de flot de contrôle suivant, où 1 représente le bloc initial :



1. Calculer le graphe des dominants
2. Est-ce que ce graphe de flot de contrôle contient une ou plusieurs boucles telles que définies dans le cours ? Si oui, lesquelles ?

Exercice 4 : 5 points

On considère les lignes de code suivantes :

```
1  L1:
2  if (n <= 3) jump L8
3  L2:
4  m := n mod 2
5  if (m == 0) jump L7
6  L3:
7  d := 3
8  jump L4
9  L4:
10 q := d * d
11 if (q > n) jump L8
12 L5:
13 m := n mod d
14 if (m == 0) jump L7
15 L6:
16 d := d + 2
17 jump L4
18 L7:
19 p := 0
20 jump L9
21 L8:
22 p := 1
23 jump L9
24 L9:
25 f(p)
```

Questions

1. Dessiner le graphe de flot de contrôle correspondant.
2. Pour chaque bloc de contrôle, indiquer les variables vivantes en entrée et en sortie.
3. Savez-vous ce que calcule ce programme ?