

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, $(T_1 \rightarrow T_2)$ est une expression de type ;
- Si T_1, T_2, \dots, T_k , sont des expressions de type, $C_k(T_1, T_2, \dots, T_k)$ est une expression de type, où C_k est un constructeur pour un type d'arité k .

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

```
1 struct Link {
2     info: 'x;
3     next: ^Link;
4 }
5
6 function findLink(link: ^Link, info: 'y): ^Link;
7 {
8     while ((link != null) && (info != link^.info))
9         link := link^.next;
10    if (link != null)
11        return link;
12    else
13        return null;
14 }
15
16 str_link: ^Link;
17 actual_link: ^Link;
18 str_link := new Link;
19 str_link^.info := 32;
20 str_link^.next := new Link;
21 str_link^.next^.info := "John";
22 str_link^.next^.next := new Link;
23 str_link^.next^.next^.info := true;
24 str_link^.next^.next^.next := null;
25 actual_link := findLink(str_link, "John")
```

Questions

1. Écrire l'expression de type de la structure `Link`
2. La fonction `findLink` est-elle polymorphe? Pourquoi?
3. On remarquera que dans le code précédent, la taille de la structure `Link` est inconnue a priori. En examinant l'ensemble du code, est-ce que le compilateur peut en inférer automatiquement le type complet pour chaque pointeur et en déduire la taille à allouer lors de chaque instruction `new Link`?
4. Est-ce que le typage statique est garanti avec ce langage de programmation?

Exercice 2 : 5 points

Soit G une grammaire algébrique $(\{S, X\}, \{A\}, S, R)$ dont les règles de production R sont les suivantes :

$S \rightarrow A C S B S$

$S \rightarrow A C S$

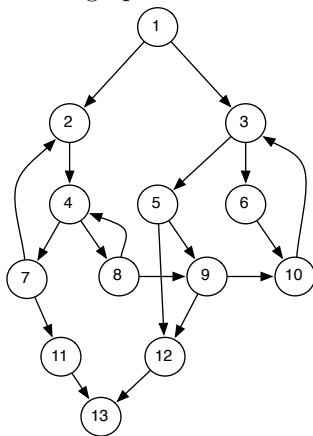
$S \rightarrow D$

Questions

1. Décrire l'analyse Earley de l'expression $ACACDBD$
2. La séquence produit deux analyses distinctes. Montrer comment l'on peut favoriser l'une des analyses avec la précedence d'opérateurs $B < A$ (B est prioritaire sur A).

Exercice 3 : 5 points

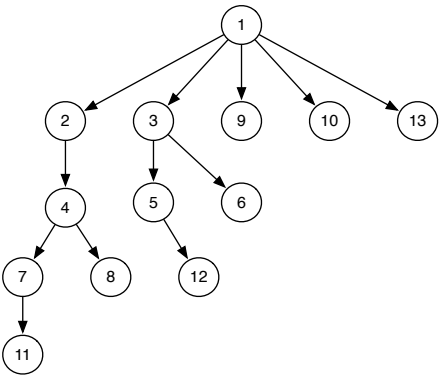
Soit le graphe de flot de contrôle suivant, où 1 représente le bloc initial :



1. Calculer le graphe des dominants

Réponse

Sommets	1	2	3	4	5	6	7	8	9	10	11	12	13
	1	1 2	1 2 3	1 2 3 4	1 2 3 4 5	1 2 3 4 5 6	1 2 3 4 5 6 7	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 10	1 2 3 4 5 6 7 8 11	1 2 3 4 5 6 7 8 9 12	1 2 3 4 5 6 7 8 9 10 11 13



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

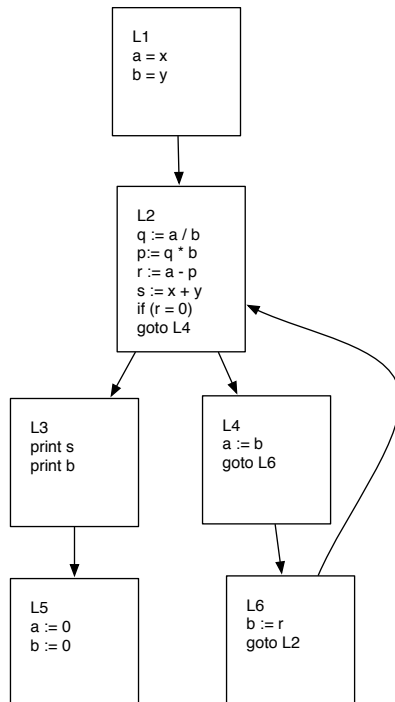
On considère les lignes de code suivantes :

```
1 L1:
2 a := x
3 b := y
4 L2:
5 q := a/b
6 p := q*b
7 r := a-p
8 s := x+y
9 if (r > 0) jump L4
10 L3:
11 print s
12 print b
13 jump L5
14 L4:
15 a := b
16 jump L6
17 L5:
18 a := 0
19 b := 0
20 jump L7
21 L6:
22 b := r
23 jump L2
24 L7:
```

Questions

1. Dessiner le graphe de flot de contrôle correspondant.

Réponse



2. Pour chaque bloc de contrôle, indiquer les variables vivantes en entrée et en sortie.
3. Comment peut-on utiliser ce résultat pour optimiser le code ?

Réponse

$s := x + y$ est un invariant qui peut être placé dans une en-tête de la boucle (2, 4, 6)

4. Que calcule ce programme ?

Réponse

Pgcd de x et y