Épreuve de Compilation

4/11/2015 Durée 2 h

Cours, TD autorisés

Exercice 1 : Interprétation Mini Jaja (*15 points : 2, 2, 2, 2, 1, 4, 2*)

Le but est de réaliser les modifications nécessaires pour que le code ci-contre soit analysable, interprétable et traduisible en JaJa Code. L'appel de la fonction `random` calcule un nombre entier choisi au hasard entre les deux valeurs données en argument. Vous devez modifier ou enrichir certains éléments du langage (questions 1 et 2) et les utiliser ou les définir en répondant aux questions 3, 4, 5, 6 et 7. La nouvelle instruction à prendre en compte est :

switch (①) {
  case ② : ③ ;
  ...
  case ② : ③ ;
  }

avec :
① et ② : des expressions,
③ : des listes d'instructions.
L'instruction `switch` contient au moins un cas et a la sémantique suivante :

`switch (e){`
  `case` v_1 : li_1 ;
  ...
  `case` v_n : li_n ;
  }

exécute la liste d'instructions li_j dans le cas où $e == v_j$ et ce pour tout $j \in 1..n$. On supposera que toutes les valeurs $v_1, v_2, \ldots, v_n$ sont différentes deux à deux.

```
Class P_2015 {
final int N = 7;
int D[N];
int j; int k=N;

main
    { j=random(0, 6);
    while (❶ k > 0) {
        k = k-1; D[k] = k;
        }; ❷
    switch (j) {
        case 0 : D[0]=D[1];
        case 1 : D[1]=D[3];
        case 2 : D[2]=D[6];
        case 3 : D[3]=D[4];
        case 4 : D[4]=D[2];
        case 6 : D[6]=D[5];
        }; ❸
    }
}
```

Questions :

1. Donner les modifications des règles de la grammaire concrète MiniJaja (forme BNF), ainsi que le ou les attributs synthétisés définissant le ou les nouveaux nœuds de syntaxe abstraite nécessaires pour prendre en compte l'instruction `switch`.
2. Définir le typage du ou des nouveaux nœuds de syntaxe abstraite introduits.
3. Dessiner l'arbre de syntaxe abstraite(ASA) du `while`. Numéroter tous les nœuds par un parcours en profondeur d'abord.
4. Pour interpréter l'instruction `switch` il y a deux solutions, soit traduire l'instruction `switch` par une composition d'instructions de la grammaire Mini-Jaja décrite en cours, soit définir des règles d'interprétation des arbres abstraits définis en réponse à la deuxième question. Décrire l'interprétation, soit en décrivant des règles de réécriture d'une instruction `switch` quelconque dans le langage Mini-jaja, soit en décrivant des règles d'interprétation des nouveaux noeuds de syntaxe abstraite.
5. Soit m_0, l'état mémoire obtenu au point ❶ après 5 boucles complètes. Décrire cet état mémoire en décrivant sa pile et la valeur de D dans le tas. Supposez que la valeur tirée au hasard (par **random**) de j est 1.

6. Décrire l'interprétation de l'instruction **while** entre ❶ et ❷ à partir de l'état mémoire m_1 obtenu au point ❶ après 6 boucles complètes. Donner la valeur de la mémoire (pile et tas) résultant de cette interprétation.
7. Décrire l'état mémoire (pile et tas) après l'exécution de l'instruction `switch`, quand l'exécution est au point ❸.

Exercice 2 : Automate (*3 points*)

Soit l'expression régulière suivante : $(a|b)^+\ a\ (a|b\ a)$. Décrire un automate non déterministe qui reconnait ce langage. Décrire sa forme déterministe, puis sa forme déterministe minimale.

Exercice 3 : Jaja Code (*2 points*)

Décrire un programme Jaja Code ayant le même effet que le programme Mini Jaja suivant :

while (k > 0) {
{
k = k-1 ;
x = k - (x + 2) ;
} ;