Sémantique et TDL.
Table des Symboles

1 Gestion de l'espace de nom avec une table des symboles

Considérons la grammaire décrivant les instructions du langage Bloc :

1. $S \rightarrow B$
2. $B \rightarrow \{\ LI\ \}$
3. $LI \rightarrow I\ LI$
4. $LI \rightarrow \Lambda$
5. $I \rightarrow$ **const** T **id** $= V$;
6. $I \rightarrow T$ **id** $= E$;
7. $I \rightarrow$ **id** $= E$;
8. $I \rightarrow$ **if** (E) B **else** B
9. $I \rightarrow$ **if** (E) B
10. $I \rightarrow$ **while** (E) B
11. $I \rightarrow$ **print** E ;
12. $T \rightarrow \ldots$
13. $E \rightarrow \ldots$
14. $E \rightarrow$ **id**
15. $V \rightarrow$ **true**
16. $V \rightarrow$ **false**
17. $V \rightarrow$ **entier**

Soit le programme :

```
test {
  int i = 1;
  const int j = 2;
  < int, int> p = < 3, 4>;
  int k = fst p;
  if ( i < 5 ) {
    int j = 5;
    j = i * (snd p);
    i = j + 1;
    while ( k < 10 ) {
      int p = 3;
      k = k + i;
    }
  } else {
    if ( i + j > 10 ) {
      const boolean p = false;
      print p;
    }
    print p;
  }
  print j;
}
```

Nous désirons implanter la sémantique de gestion de l'espace de nom pour Bloc en respectant les mêmes principes que Java, en utilisant une table des symboles.

Nous souhaitons vérifier que tous les identificateurs de variables ou de constantes utilisés sont bien définis et visibles lors de leur utilisation et que les constantes ne sont jamais modifiées par des affectations. Nous autorisons la redéfinition de variables et de constantes dans des blocs imbriqués mais nous interdisons la redéfinition dans un même bloc. Cette sémantique étend celle des expressions étudiée en cours.

1. Définir la table des symboles que vous souhaitez construire pour le programme précédent. Vous indiquerez pour chaque ligne du programme quels sont les identificateurs visibles et la définition qui leur est associé.

2. Définir les activités de vérification qui doivent être effectuées pour le programme précédent. Vous indiquerez celles-ci pour chaque ligne du programme.

3. En déduire les règles sémantiques nécessaires.

2 Retour sur la construction de l'arbre abstrait

Nous souhaitons compléter la sémantique de construction de la table des symboles pour modifier l'arbre abstrait pour :

- construire un lien entre l'utilisation d'une variable dans une affectation ou une expression et la définition de cette variable ;
- remplacer l'utilisation d'une constante (objet `VariableUse`) par un lien avec sa valeur dans la définition de la constante.

1. Modifier les règles sémantiques dans ce but.