Minigloca

Vladislas de Haldat

24 janvier 2023

Table des matières

1 Un langage impératif simple

Pour commencer, définissons une syntaxe abstraite minimale que nous utiliserons tout le long de cette étude. Cette syntaxe se composera de trois blocs fondamentaux que sont les expressions arithmétiques, les expressions booléennes ainsi que les déclarations. Nous n'incluons pas pour le moment la déclaration de routines au sein de cette syntaxe.

1.1 Expressions arithmétiques

Les expressions arithmétiques sont définies sur l'ensemble des entiers relatifs. On se donne les opérateurs de l'addition, de la soustraction ainsi que de la multiplication. À ces opérateurs l'on pourra appliquer des entiers ainsi que des identifiants de variables.

$$
\begin{aligned}
Int &\to n && n \in \mathbb{Z} \\
Id &\to x \mid y \mid z \mid \cdots \\
a &\to Int \mid Id \mid op_A(a_1, a_2) && op_A \in \{+, -, \times\}
\end{aligned}
$$

L'identifiant d'une variable est, de manière générale, une chaîne de caractères.

1.2 Expressions booléennes

Les expressions booléennes nous permettent d'introduire la comparaison entre deux expressions arithmétiques, ainsi que les opérateurs booléens sur les expressions booléennes.

$$
\begin{aligned}
b &\to \textbf{true} \mid \textbf{false} \\
&\mid op_R(a_1, a_2) && op_R \in \{<, =\} \\
&\mid op_B(b_1, b_2) && op_B \in \{\wedge, \vee\} \\
&\mid \neg b
\end{aligned}
$$

1.3 Déclarations

Les déclarations sont définies de la manière suivante :

$$
\begin{aligned}
s \rightarrow\ & Id := a \\
& \mid s_1 \,;\, s_2 \\
& \mid \\
& \mid \text{if } b \text{ then } s_1 \text{ else } s_2 \\
& \mid \text{while } b \text{ do } s_1
\end{aligned}
$$

1.4 Exemple

Voici un premier exemple sur ce langage

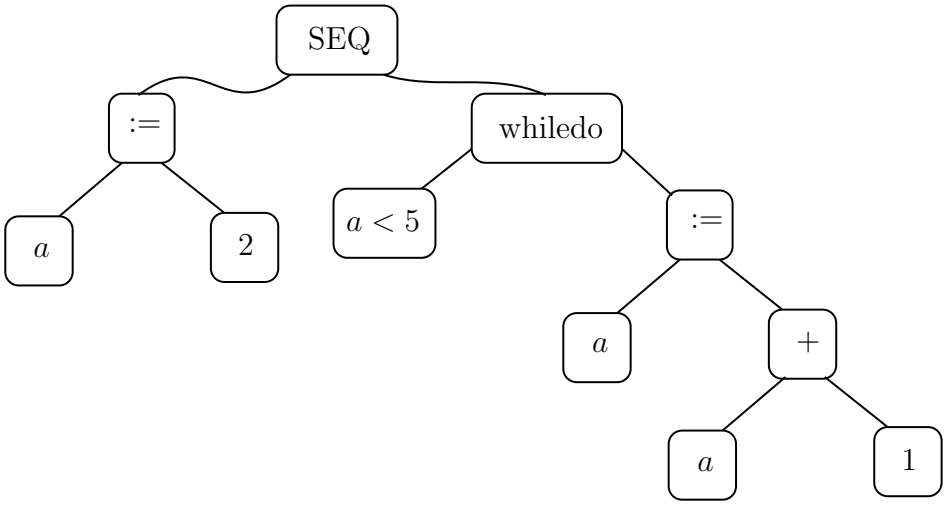
```
a := 1;
b := 20;

if a = 3 then
        c := 4
else
        c := 6
endif;

while b < 100 do
        b := b + 1
done
```

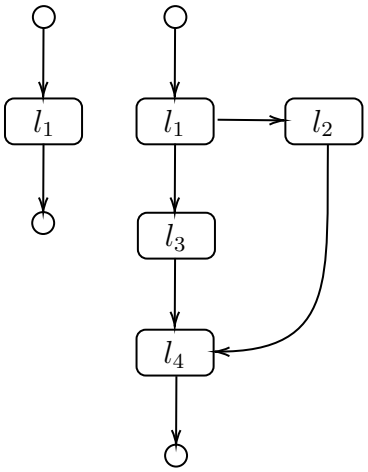
1.5 Arbre de syntaxe abstraite

Ce modèle de représentation permet de construire un arbre de syntaxe abstraite (AST) du programme en question, cette structure étant un moyen pratique de le représenter. Cela nous servira particulièrement lors des différentes analyses qui seront effectuées, et notamment l'analyse par flot de contrôle. Cette dernière ne requièrant pas la connaissance de l'ordre d'exécution du programme, les AST sont tout-à-fait adaptés !



1.6 Graphe de flot de contrôle

Un graphe de flot de contrôle est un graphe orienté dont les noeuds représentent une déclaration et les arcs un flot de contrôle. Chaque noeud est une unique opération de notre programme. Dans notre cas, le graphe de flot de contrôle aura un seul point d'entrée et un seul point de sortie. Ils sont considérés comme des noeuds "sans opération". De manière à construire ce graphe, on peut assigner à chaques déclarations procurant une unique exécution, un label. Dans notre cas, ces déclarations sont l'assignation, la condition ainsi que la boucle. En voici quelques exemples :



1.7 Implémentation

Détails sur l'implémentation.