

Travaux Dirigés de compilation n°2

Licence d'informatique

Traduction et attributs hérités

Le but de ce TD est de se familiariser avec la traduction descendante et avec les attributs hérités.

► Exercice 1. Traduction par descente récursive

1. Complétez `decl-var.c` en ajoutant les fonctions nécessaires pour obtenir un analyseur par descente récursive qui implémente la grammaire algébrique ci-dessous.

$$\begin{aligned}
 DeclVar &\rightarrow Type\ Vars\ ; \\
 Type &\rightarrow \text{int} \\
 Type &\rightarrow \text{float} \\
 Vars &\rightarrow \text{id}\ \text{resteVars} \\
 \text{resteVars} &\rightarrow \text{, id}\ \text{resteVars} \\
 \text{resteVars} &\rightarrow \varepsilon
 \end{aligned}$$

Cette grammaire algébrique simplifiée engendre les déclarations de variables en C. Pour vos tests vous pouvez utiliser `decl-var.tpc`.

2. Complétez votre programme en un traducteur par descente récursive qui implémente la grammaire attribuée ci-dessous.

$$\begin{aligned}
 DeclVar &\rightarrow Type\ Vars\ ; & Vars.type &:= Type.type \\
 Type &\rightarrow \text{int} & Type.type &:= \text{integer} \\
 Type &\rightarrow \text{float} & Type.type &:= \text{real} \\
 Vars &\rightarrow \text{id}\ \text{resteVars} & \text{addVar}(\text{id.name}, Vars.type); \\
 & & \text{resteVars.type} &:= Vars.type \\
 \text{resteVars} &\rightarrow \text{, id}\ \text{resteVars} & \text{addVar}(\text{id.name}, \text{resteVars.type}); \\
 & & \text{resteVars}_1.type &:= \text{resteVars.type} \\
 \text{resteVars} &\rightarrow \varepsilon
 \end{aligned}$$

Cette grammaire attribuée simplifiée traduit les déclarations de variables en C.

► Exercice 2. Traduction ascendante avec des attributs hérités

1. La grammaire attribuée de l'exercice 1 est-elle L-attribuée ?
2. Faites avec `bison` un traducteur pour cette même grammaire attribuée. Dans les actions, récupérez la valeur des attributs hérités sans utiliser de variable globale.
3. Même exercice en sauvegardant les attributs hérités dans une variable globale.