Travaux dirigés

Introduction à la Compilation

3-Syntaxe abstraite

Jean-Christophe Le Lann, Quentin Ducasse

Janvier 2023

Le but des exercices qui suivent est de compléter le parseur du TD précédent sur `Mini-C`. Jusqu'ici votre parseur ne réalisait que la consommation des lexèmes. Ici, le parseur cherche à créer, en mémoire, une représentation arborescente du programme, sous une forme abstraite. Il s'agit de l'**arbre de syntaxe abstraite** ou **AST**.

Notre paradigme de programmation orienté-objet nous invite à décrire chaque noeud de l'arbre comme une instance d'une classe apropriée à l'élément syntaxique considéré. Considérons par exemple le code suivant, représentatif du `if` du langage `Mini-C`.

```
class IfStatement(Statement):
  def __init__(self, cond, body, else_):
      self.cond=cond
      self.body=body
      self.else_=else_
```

Il s'agit d'une classe, dont les variables d'instances (ou *attributs*) sont :

- `cond` : la **condition** du `if`. *In fine*, il s'agira d'une *Expression*[1], dont la classe sera décrite plus tard, mais notre langage Python, interprété dynamiquement, ne nous force pas à le préciser ici, lors de la conception.
- `body` : le **corps** du `if`. Il s'agit d'un objet qui contient toutes les instructions exécutées sur la condition précédente.
- `else_` : Il s'agit d'un objet qui contient toutes les instructions exécutées lorsque la condition précédente n'est pas vérifée. Cet objet pourra lui-même être décrit par ailleurs dans une classe `Else`

Exercice 1 : Syntaxe abstaite du `Mini-C`

A l'aide de l'exemple du `If`, écrire l'ensemble des classes décrivant la syntaxe abstraite du `Mini-C`.

1. Il s'agit plus précisément d'une référence vers cet objet Expression, mais le langage Python vous fait grâce de ces subtilités

Exercice 2 : Utilisation manuelle

Un programme `Mini-C` appelé “exo2.c” est fourni sous Moodle. Décrire ce programme à l’aide de votre syntaxe abstraite, en instanciant correctement les classes précédentes.

Exercice 3 : Parseur complet

Désormais munis de la syntaxe asbtraite du `Mini-C`, modifier le parseur du TD précédent de manière à ce qu’il génère en mémoire un AST complet.