UFR 919 Sciences de l'ingénieur
Master Informatique – Spécialité STL
Analyse de Programmes et Sémantique (APS)

TD7 — Sémantique dénotationnelle
Extension du mini-langage

Jacques Malenfant, Olena Rogovchenko

1 Expression d'alternative « e_1 ? e_2 : e_3 »

Ajoutez au mini-language une expression d'alternative `? :`. Donnez une syntaxe abstraite, puis donnez-en la sémantique dénotationnelle.

2 Instruction `swap`

Ajoutez au mini-language une instruction `swap` qui, appliquée à deux variables, échange leurs valeurs. Donnez une syntaxe abstraite, puis donnez-en la sémantique dénotationnelle.

3 Alternative entières à trois voies

Fortran IV, un langage qui n'offrait pas d'instructions structurées comme les alternatives et les boucles, affectionnait toutes les variantes de sauts de type *goto* à des étiquettes posées sur des instructions. Il proposait en particulier l'alternative arithmétique (« *arithmetic-IF* ») à trois voies suivante :

```
IF (<expression>) etiq1, etiq2, etiq3
```

L'expression est évaluée, et le programme saute à l'étiquette `etiq1` si le résultat de l'expression est négatif, à `etiq2` s'il est nul, et à `etiq3` s'il est positif.

Nous vous demandons d'introduire dans le mini-langage une version structurée de cette alternative arithmétique puis d'en donner une sémantique dénotationnelle. Il s'agira donc d'une instruction d'alternative qui aura trois alternants qui sont également des instructions : l'un si la valeur de la condition est négative, le second si elle est nulle et le troisième si elle est positive. Proposez une syntaxe abstraite, puis donnez sa sémantique.

4 Définition de variables par `let`

Introduire dans le mini-langage une instruction `let` pour permettre la définition de variables locales. Pour cela, il faudra être en mesure d'allouer de l'espace en mémoire, c'est-à-dire de trouver

une adresse libre. Nous avons déjà vu dans la sémantique opérationnelle structurelle qu'il est possible de faire cela en utlisant deux marqueurs, `unused` pour les adresses libres et `undefined` pour les adresses allouées mais non encore initialisées. Comment modifier le traitement de la mémoire pour permettre de faire ces allocations ?

Donnez ensuite un syntaxe abstraite permettant de définir une variable et d'exécuter une instruction dans ce nouveau contexte, puis donnez-en la sémantique dénotationnelle.

5 Définition et application de fonctions

Ajoutez au mini-langage la définition et l'application de fonctions. Supposez que les fonctions sont définies à la manière du `let` par une instruction `defun` et qu'elles n'ont qu'un seul paramètre. Proposez la syntaxe abstraite de cette instruction et de l'appel de fonction, puis donnez-en leur sémantique dénotationnelle.

6 Définition et l'appel de procédures

Ajoutez au mini-langage la définition et l'appel de procédures. Supposez que les procédures sont définies à la manière du `let` par une instruction `defproc` et qu'elles n'ont qu'un seul paramètre. Proposez la syntaxe abstraite de cette instruction et de l'appel de procédure, puis donnez-en leur sémantique dénotationnelle.

7 Partie TME : implantation des extensions

Pour chacune des extensions du mini-langage vue en TD, implantez la sémantique obtenue en Scheme. Construisez des exemples de programmes et leurs arbres de syntaxe abstraite pour tester vos implantations.