

Sémantique des langages de programmation

Mastère PLMT

examen

Yves Bertot

Décembre 2004

1 Démontrer des propriétés

On considère le langage d'expressions suivant (les variables e_i dénotent des expressions).

- n (n est un entier),
- $e_1 + e_2$.

L'évaluation des expressions de ce langage est décrite par les deux règles suivantes:

$$\frac{\vdash e_1 \rightarrow v_1 \quad \vdash e_2 \rightarrow v_2}{\vdash e_1 + e_2 \rightarrow v_1 + v_2}$$
$$\overline{\vdash n \rightarrow n}$$

On dispose d'une machine dont l'état peut être modélisé par une pile (ou une liste) de valeurs entières. Cette machine dispose de deux instructions, `imm n` , qui ajoute une valeur entière au sommet de la pile et `add` qui additionne les deux valeurs en sommet de pile. Ce comportement est décrit par les règles suivantes:

$$\overline{\sigma \vdash \text{imm } n \mapsto n \cdot \sigma}$$
$$\overline{n_1 \cdot n_2 \cdot \sigma \vdash \text{add} \mapsto (n_1 + n_2) \cdot \sigma}$$

Cette machine exécute des séquences d'instructions, ce qui se décrit par les règles suivantes:

$$\sigma \vdash \emptyset \mapsto^* \sigma$$
$$\frac{\sigma \vdash i \mapsto \sigma' \quad \sigma' \vdash p \mapsto^* \sigma''}{\sigma \vdash i \cdot p \mapsto^* \sigma''}$$

On écrit un compilateur des expressions vers les séquences d'instructions pour cette machine. Ce compilateur peut être modélisé par les règles d'inférence suivantes, où le

judgement $p \vdash e \rightsquigarrow p'$ peut être lu comme “le programme p' calcule la valeur de e , met le résultat en sommet de pile, puis exécute p ”.

$$\frac{\text{add} \cdot p \vdash e_1 \rightsquigarrow p' \quad p' \vdash e_2 \rightsquigarrow p''}{p \vdash e_1 + e_2 \rightsquigarrow p''}$$

$$\frac{}{p \vdash n \rightsquigarrow (\text{imm } n) \cdot p}$$

Démontrez que ce compilateur est correct, c'est-à-dire

$$\forall e, p, \emptyset \vdash e \rightsquigarrow p \Rightarrow \forall n, \vdash e \rightarrow n \Rightarrow \emptyset \vdash p \mapsto^* n \cdot \emptyset$$

Consider the expression language with the following constructs (the variables e_i denote expression):

- n (n is an integer),
- $e_1 + e_2$.

The evaluation of expressions in this language is described by the following two rules:

$$\frac{\vdash e_1 \rightarrow v_1 \quad \vdash e_2 \rightarrow v_2}{\vdash e_1 + e_2 \rightarrow v_1 + v_2}$$

$$\frac{}{\vdash n \rightarrow n}$$

We have a machine, the state of which can be modeled by a stack (or a list) of integer values. This machine has two instructions, $\text{imm } n$, which pushes an integer value on top of the stack and add which adds the two value on the top of the stack and then pushes the result back on the stack. This behavior is described the following two rules:

$$\frac{}{\sigma \vdash \text{imm } n \mapsto n \cdot \sigma}$$

$$\frac{}{n_1 \cdot n_2 \cdot \sigma \vdash \text{add} \mapsto (n_1 + n_2) \cdot \sigma}$$

This machine executes instruction sequences, as described by the following rules:

$$\frac{\sigma \vdash \emptyset \mapsto^* \sigma \quad \sigma \vdash i \mapsto \sigma' \quad \sigma' \vdash p \mapsto^* \sigma''}{\sigma \vdash i \cdot p \mapsto^* \sigma''}$$

A compiler from expressions to instruction sequences is described by the following rules. In these rules the judgement $p \vdash e \rightsquigarrow p'$ can be read as “the program p' computes the value of e , places the result on top of the stack, and then executes p ”.

$$\frac{\text{add} \cdot p \vdash e_1 \rightsquigarrow p' \quad p' \vdash e_2 \rightsquigarrow p''}{p \vdash e_1 + e_2 \rightsquigarrow p''}$$

$$\frac{}{p \vdash n \rightsquigarrow (\text{imm } n) \cdot p}$$

Prove that this compiler is correct:

$$\forall e, p, \emptyset \vdash e \rightsquigarrow p \Rightarrow \forall n, \vdash e \rightarrow n \Rightarrow \emptyset \vdash p \mapsto^* n \cdot \emptyset$$