Exercises

Spring 2005

(Hanne Riis Nielson)

March 7, 2005

Chapter 4 of [NN] describes the generation of code for the While language to the abstract machine AM. The translation is given by three function $\mathcal{CA}$, $\mathcal{CB}$ and $\mathcal{CS}$; they are all defined compositionally. We now want to specify the same translation using three inference systems. At the same time we want to estimate how high the evaluation stack might be during the execution of the generated code.

For arithmetic expressions we will have two kinds of configurations:

$$\begin{array}{rcl} a & \in & \mathbf{Aexp} \\ (c,i) & \in & \mathbf{Code} \times \mathbf{N} \end{array}$$

The final configurations have the latter form. The inference relations have the form:

$$a \mapsto_{\text{Aexp}} (c,i)$$

This means that the code for a is c and that during the execution of c there will be a need for an evaluation stack of height at most i. The

inference system is given by:

$$n \mapsto_{\text{Aexp}} (\text{PUSH-}n, 1)$$

$$x \mapsto_{\text{Aexp}} (\text{FETCH-}x, 1)$$

$$\frac{a_1 \mapsto_{\text{Aexp}} (c_1, i_1) \quad a_2 \mapsto_{\text{Aexp}} (c_2, i_2)}{a_1 + a_2 \mapsto_{\text{Aexp}} (c_2 : c_1 : \text{ADD}, \max(i_1 + 1, i_2))}$$

$$\frac{a_1 \mapsto_{\text{Aexp}} (c_1, i_1) \quad a_2 \mapsto_{\text{Aexp}} (c_2, i_2)}{a_1 * a_2 \mapsto_{\text{Aexp}} (c_2 : c_1 : \text{MULT}, \max(i_1 + 1, i_2))}$$

$$\frac{a_1 \mapsto_{\text{Aexp}} (c_1, i_1) \quad a_2 \mapsto_{\text{Aexp}} (c_2, i_2)}{a_1 - a_2 \mapsto_{\text{Aexp}} (c_2 : c_1 : \text{SUB}, \max(i_1 + 1, i_2))}$$

Question 1: Construct an inference tree showing that

$$((\mathtt{x} * 2) * \mathtt{y}) + 3 \mapsto_{\text{Aexp}} \langle \text{PUSH-}3 : \text{FETCH-}\mathtt{y} : \text{PUSH-}2 : \text{FETCH-}\mathtt{x} : \text{MULT} : \text{MULT} : \text{ADD}, 4 \rangle$$

□

Question 2: Specify similar transition systems $b \mapsto_{\text{Bexp}} (c, i)$ and $S \mapsto_{\text{Stm}} (c, i)$ for boolean expressions and statements. Make sure that the estimates for the height of the evaluation stack are not unnecesary pessimistic. □

We want that the maximal height of the evaluation stack during the execution of the code is as small as possible. As shown in Question 1 above we have

$$((\mathtt{x} * 2) * \mathtt{y}) + 3 \mapsto_{\text{Aexp}} \langle \text{PUSH-}3 : \text{FETCH-}\mathtt{y} : \text{PUSH-}2 : \text{FETCH-}\mathtt{x} : \text{MULT} : \text{MULT} : \text{ADD}, 4 \rangle$$

meaning that an evaluation stack of height 4 is required. However, as shown in Question 3 below it is indeed safe to replace the above code with

$$\text{PUSH-}2 : \text{FETCH-}\mathtt{x} : \text{MULT} : \text{FETCH-}\mathtt{y} : \text{MULT} : \text{PUSH-}3 : \text{ADD}$$

so an evaluation stack of height 2 will do.

Question 3: Argue that we have

$$\langle \text{PUSH-3} : \text{FETCH-}\texttt{y} : \text{PUSH-2} : \text{FETCH-}\texttt{x} : \text{MULT} : \text{MULT} : \text{ADD}, e, s\rangle \\ \rhd^* \langle \epsilon, e', s'\rangle$$

if and only if

$$\langle \text{PUSH-2} : \text{FETCH-}\texttt{x} : \text{MULT} : \text{FETCH-}\texttt{y} : \text{MULT} : \text{PUSH-3} : \text{ADD}, e, s\rangle \\ \rhd^* \langle \epsilon, e', s'\rangle$$

for all states s and s' and all stacks e and e'. □

Question 4: Based on the above considerations suggest a modification of the translation of arithmetic expressions that will have less requirements to the height of the evaluation stack. Argue that the new code generation is correct. □