Exercises on Concepts of Programming Languages

Christian Rinderknecht

9 February 2006

1 Calculator

Assume the following system of inference rules:

$$\frac{e_1 \rightarrow e_1'}{e_1 \times e_2 \rightarrow e_1' \times e_2}\ \langle\textsc{Mult}_1\rangle \qquad \frac{e_2 \rightarrow e_2'}{e_1 \times e_2 \rightarrow e_1 \times e_2'}\ \langle\textsc{Mult}_2\rangle$$

$$\frac{e_1 \rightarrow e_1'}{e_1 + e_2 \rightarrow e_1' + e_2}\ \langle\textsc{Add}_1\rangle \qquad \frac{e_2 \rightarrow e_2'}{e_1 + e_2 \rightarrow e_1 + e_2'}\ \langle\textsc{Add}_2\rangle$$

$$\frac{e_1 \rightarrow e_1'}{e_1 - e_2 \rightarrow e_1' - e_2}\ \langle\textsc{Sub}_1\rangle \qquad \frac{e_2 \rightarrow e_2'}{e_1 - e_2 \rightarrow e_1 - e_2'}\ \langle\textsc{Sub}_2\rangle$$

$$\frac{e_1 \rightarrow e_1'}{e_1/e_2 \rightarrow e_1'/e_2}\ \langle\textsc{Div}_1\rangle \qquad \frac{e_2 \rightarrow e_2'}{e_1/e_2 \rightarrow e_1/e_2'}\ \langle\textsc{Div}_2\rangle$$

We also assume that we have an infinity of rules for multiplying, adding, subtracting and dividing (integer division) numbers.

1. Is this system deterministic?
2. Reduce the following expressions in all the possible ways and give at each rewrite step the corresponding substitution:
   (a) $(1 + 2) \times (3 + (4 \times 5))$
   (b) $((1 + 2)/0) \times (3 + 4)$
   (c) $(1 + (2 \times 3))/((4 + 5) - 9)$

Assume now the following variant and answer the previous questions again.

$$\frac{e_1 \to e_1'}{e_1 \times e_2 \to e_1' \times e_2}\ \langle\textsc{Mult}_1\rangle \qquad \frac{e \to e'}{v \times e \to v \times e'}\ \langle\textsc{Mult}_2\rangle$$

$$\frac{e \to e'}{e + v \to e' + v}\ \langle\textsc{Add}_1\rangle \qquad \frac{e_2 \to e_2'}{e_1 + e_2 \to e_1 + e_2'}\ \langle\textsc{Add}_2\rangle$$

$$\frac{e_1 \to e_1'}{e_1 - e_2 \to e_1' - e_2}\ \langle\textsc{Sub}_1\rangle \qquad \frac{e \to e'}{v - e \to v - e'}\ \langle\textsc{Sub}_2\rangle$$

$$\frac{e \to e'}{e/v \to e'/v}\ \langle\textsc{Div}_1\rangle \qquad \frac{e_2 \to e_2'}{e_1/e_2 \to e_1/e_2'}\ \langle\textsc{Div}_2\rangle$$

Assume now that we have a different rule $\langle\textsc{Div}_1\rangle$ and new rules for handling the division by zero:

$$\frac{v \neq 0 \qquad e \to e'}{e/v \to e'/v}\ \langle\textsc{Div}_1\rangle \qquad e/0 \to \mathsf{NaN} \quad \langle\textsc{DivZero}\rangle$$

$$\mathsf{NaN} \times e \to \mathsf{NaN} \quad \langle\textsc{Mult-Err}_1\rangle \qquad e \times \mathsf{NaN} \to \mathsf{NaN} \quad \langle\textsc{Mult-Err}_2\rangle$$

$$\mathsf{NaN} + e \to \mathsf{NaN} \quad \langle\textsc{Add-Err}_1\rangle \qquad e + \mathsf{NaN} \to \mathsf{NaN} \quad \langle\textsc{Add-Err}_2\rangle$$

$$\mathsf{NaN} - e \to \mathsf{NaN} \quad \langle\textsc{Sub-Err}_1\rangle \qquad e - \mathsf{NaN} \to \mathsf{NaN} \quad \langle\textsc{Sub-Err}_2\rangle$$

$$\mathsf{NaN}/e \to \mathsf{NaN} \quad \langle\textsc{Div-Err}_1\rangle \qquad e/\mathsf{NaN} \to \mathsf{NaN} \quad \langle\textsc{Div-Err}_2\rangle$$

Answer the same questions again.

Is it better to have the following rule?

$$\frac{e_1 \to e_1'}{e_1/e_2 \to e_1'/e_2}\ \langle\textsc{Div}_1\rangle$$

2 Boolean expressions

Consider the system of inference rules

$$\mathsf{true} \wedge e \to e \quad \langle \wedge_{\textsc{True}} \rangle \qquad\qquad \mathsf{false} \wedge e \to \mathsf{false} \quad \langle \wedge_{\textsc{False}} \rangle$$

$$\frac{e_1 \to e_1'}{e_1 \wedge e_2 \to e_1' \wedge e_2} \langle \wedge \rangle \qquad\qquad \neg\mathsf{false} \to \mathsf{true} \quad \langle \textsc{Not-False} \rangle$$

$$\neg\mathsf{true} \to \mathsf{false} \quad \langle \textsc{Not-True} \rangle \qquad\qquad \frac{e \to e'}{\neg e \to \neg e'} \langle \textsc{Not} \rangle$$

$$e_1 \vee e_2 \to \neg(\neg e_1 \wedge \neg e_2) \quad \langle \textsc{Or} \rangle$$

1. Is this system deterministic?
2. Reduce the following expressions in several ways if possible, and give at each rewrite step the corresponding substitution:
   (a) $(\mathsf{true} \wedge (\mathsf{false} \vee \mathsf{true})) \wedge \neg(\mathsf{true} \vee (\mathsf{true} \wedge \mathsf{true}))$
   (b) $\neg((\mathsf{true} \wedge \mathsf{true}) \vee \neg(\mathsf{false} \vee \neg\mathsf{true}))$

3 Arithmetic

Let us model the integers. The number 0 is noted Zero. If an integer is noted n, then Succ(n) denotes the next integer and Pred(n) the previous. For example

New notation	Mathematical notation
Zero	0
Succ(Zero)	1
Succ(Succ(Zero))	2
...	...
Pred(Zero)	−1
Pred(Pred(Zero))	−2
...	...

Let us define now a function IsZero which returns the boolean true if the argument is Zero and false if the argument is not Zero:

$$\begin{aligned} \textsc{IsZero}(\textsc{Zero}) &\to \mathsf{true} \\ \textsc{IsZero}(\textsc{Succ}(n)) &\to \mathsf{false} \\ \textsc{IsZero}(\textsc{Pred}(n)) &\to \mathsf{false} \end{aligned}$$

But this definition is broken, because it implies, for instance

$$\textsc{IsZero}(\textsc{Pred}(\textsc{Succ}(\textsc{Zero}))) \to \mathsf{false}$$

What solution do you propose to fix it? Answer:

$$\begin{aligned}\textsc{Succ}(\textsc{Pred}(n)) &\to n \\ \textsc{Pred}(\textsc{Succ}(n)) &\to n\end{aligned}$$

4 Stacks

$$\textsc{IsEmpty}(\textsc{Push}(x, s)) \to \mathsf{false} \quad \langle \textsc{IsNotEmpty} \rangle$$

$$\textsc{IsEmpty}(\textsc{Empty}) \to \mathsf{true} \quad \langle \textsc{IsEmpty} \rangle$$

$$\textsc{Length}(\textsc{Empty}) \to 0 \quad \langle \textsc{ZeroLength} \rangle$$

$$\textsc{Length}(\textsc{Push}(x, s)) \to 1 + \textsc{Length}(s) \quad \langle \textsc{Length} \rangle$$

$$\textsc{Rev}(\textsc{Empty}) \to \textsc{Empty} \quad \langle \textsc{Rev-E} \rangle$$

$$\textsc{Rev}(\textsc{Push}(x, s)) \to \textsc{Append}(\textsc{Rev}(s), \textsc{Push}(x, \textsc{Empty})) \quad \langle \textsc{Rev-P} \rangle$$