Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Sascha Böhme, Alexander Krauss

WS 2010/11
29. 11. 2010

Semantics of Programming Languages

Exercise Sheet 6

This exercise builds on theory `Small_Step`.
To save some typing, download the theory `Ex06_Template` and fill in the gaps.

Exercise 6.1 Small step equivalence

We define an equivalence relation $\approx$ on programs that uses the small-step semantics. Unlike with $\sim$, we also demand that the programs take the same number of steps.

The following relation is the n-steps reduction relation:

inductive
  n_steps :: "*com* $*$ *state* $\Rightarrow$ *nat* $\Rightarrow$ *com* $*$ *state* $\Rightarrow$ *bool*"
  ("_ $\rightarrow$^_ _" [60,1000,60]999)
where
  zero_steps: "*cs* $\rightarrow$^0 *cs*" |
  one_step: "*cs* $\rightarrow$ *cs'* $\Longrightarrow$ *cs'* $\rightarrow$^*n* *cs''* $\Longrightarrow$ *cs* $\rightarrow$^(*Suc n*) *cs''*"

Prove the following lemmas:

lemma *small_steps_n*: "*cs* $\rightarrow*$ *cs'* $\Longrightarrow$ ($\exists$ *n*. *cs* $\rightarrow$^*n* *cs'*)"
lemma *n_small_steps*: "*cs* $\rightarrow$^*n* *cs'* $\Longrightarrow$ *cs* $\rightarrow*$ *cs'*"

The equivalence relation is defined as follows:

definition
  small_step_equiv :: "*com* $\Rightarrow$ *com* $\Rightarrow$ *bool*" (**infix** "$\approx$" 50) **where**
  "*c* $\approx$ *c'* == ($\forall$ *s t n*. (*c*,*s*) $\rightarrow$^*n* (*SKIP*, *t*) = (*c'*, *s*) $\rightarrow$^*n* (*SKIP*, *t*))"

Prove the following lemma:

lemma *small_eqv_implies_big_eqv*: "*c* $\approx$ *c'* $\Longrightarrow$ *c* $\sim$ *c'*"

How about the reverse implication?

Homework 6

Submission until Wednesday, December 8, 2010, 12:00 (noon).

In this execercise we extend our language with nondeterminism. We want to include a command c_1 *OR* c_2, which expresses the nondeterministic choice between two commands. That is, when executing c_1 *OR* c_2 either c_1 or c_2 may be executed, and it is not specified which one.

(a) Modify the datatype *com* to include a new constructor *Or*.

(b) Adapt the big step semantics to include rules for the new construct.

(c) Prove that $c_1 \ \mathit{OR} \ c_2 \sim c_2 \ \mathit{OR} \ c_1$.

(d) Adapt the small step semantics, and the equivalence proof of big and small step semantics.

Note: It is easiest if you take the existing theories and modify them. Please mark the places where you did any modification, such that they can be immediately recognized.