ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 10: Small Step Semantics

Submission deadline: May 16th, 2011

Please submit your solution before **9:15am** on the submission date specified above. Solutions can be submitted via e-mail or by using the boxes to the left of **RZ F1**. Make sure that the first page (and preferably each sheet) always contains your name, the exercise sheet number as well as your tutor's name and the weekday (Tuesday or Wednesday) of your exercise group. Don't forget to staple your pages if you submit more than one page.

Notation: As in slide 105 of the lectures, we use the vector notation $\sigma[\vec{y} \mapsto \vec{v}]$, as a shorthand for a (possibly empty) sequence of state updates. i.e., for some $m \geq 0$, $\sigma[\vec{y} \mapsto \vec{v}]$ abbreviates $\sigma[y_1 \mapsto v_1][y_2 \mapsto v_2] \ldots [y_m \mapsto v_m]$, for some sequences of variables $\vec{y} = y_1, y_2, \ldots, y_m$ and corresponding values $\vec{v} = v_1, v_2, \ldots, v_m$. For empty sequences (i.e., $m = 0$), then $\sigma[\vec{y} \mapsto \vec{v}]$ is just σ.

Assignment 1 - implementing SOS

In this assignment you will extend the simple IMP interpreter with the structural operational semantics. Download the skeleton file `simpi_skeleton2.lhs` from the course web page and implement the function

```
transSOS :: Config -> Config
```

that encodes the rules presented in the lecture for the structural operational semantics. The places where you have to insert your code in the skeleton file are marked by TODO. Compare your implementation of transSOS with the function transNS that implements the rules for the natural semantics.

Please mail your solution of this assignment to your tutor. The email addresses of the tutors are:

Alex Summers	`alexander.summers@inf.ethz.ch`
Yannis Kassios	`ioannis.kassios@inf.ethz.ch`
Pietro Ferrara	`pietro.ferrara@inf.ethz.ch`
Stefan Heule	`stheule@student.ethz.ch`

Assignment 2 - states and state updates

Recall that states σ are total functions from variables to values. Two states σ_1 and σ_2 are equal (written $\sigma_1 = \sigma_2$ as usual) if they define the same function. That is, $\sigma_1 = \sigma_2$ if and only if

$\forall x \in \mathrm{Var}, \sigma_1(x) = \sigma_2(x)$. There can be many different ways of defining equal states, and this question deals with a few such cases.

(i) Prove that (for all states σ, variables x and values v_1, v_2), $\sigma[x \mapsto v_1][x \mapsto v_2] = \sigma[x \mapsto v_2]$.

(ii) Prove that (for all states σ, variables x, y and values v_1, v_2), if $x \neq y$, then $\sigma[x \mapsto v_1][y \mapsto v_2] = \sigma[y \mapsto v_2][x \mapsto v_1]$. Is the condition $x \neq y$ necessary?

(iii) Prove that for all variables x, values v_1, v_2, for all sequences (of length $m \geq 0$) of variables $\vec{y} = y_1, y_2, \ldots, y_m$ and corresponding values $\vec{v'} = v'_1, v'_2, \ldots, v'_m$, and for all states σ': $\sigma'[x \mapsto v_1][\vec{y} \mapsto \vec{v'}][x \mapsto v_2] = \sigma'[\vec{y} \mapsto \vec{v'}][x \mapsto v_2]$.

Note: the last result tells you that you can "clean up" states as you write a derivation/derivation sequence; if you encounter a state to which many state updates to the same variable have been applied, you can always leave out all except the last one, and you'll still define exactly the same state.

Assignment 3 (from last year's exam)

Consider the following **IMP** statement s:

```
while n # 0 do
    a := a+n;
    b := b*n;
    n := n-1
end
```

Let σ be a state such that $\sigma(\mathtt{a}) = 0$, $\sigma(\mathtt{b}) = 1$, and $\sigma(\mathtt{n}) = 2$. Prove using the structural operational semantics that there is a state σ' with $\sigma'(\mathtt{a}) = 3$, $\sigma'(\mathtt{b}) = 2$, and $\sigma'(\mathtt{n}) = 0$ such that $\langle s, \sigma\rangle \to_1^* \sigma'$. Hint: provide the complete derivation sequence. You have not to provide the derivation tree for each individual transition.

Assignment 4 - composing executions

Let s_1 and s_2 be statements, σ and σ' states, and k a positive integer. Prove that if $\langle s_1, \sigma\rangle \to_1^k \sigma'$ then $\langle s_1; s_2, \sigma\rangle \to_1^k \langle s_2, \sigma'\rangle$.

Assignment 5 - executing in similar states

Prove that, for all states σ, σ', statements s and configurations γ, if $\forall y \in \mathit{FV}(S), (\sigma(y) = \sigma'(y))$ and also $\langle s, \sigma\rangle \to_1^* \gamma$, then there exist variables $\vec{x}$ and corresponding values $\vec{v}$ such that the $\vec{x}$ are free variables of s (i.e., $\{\vec{x}\} \subseteq \mathit{FV}(s)$) such that *either*:

1. $\gamma = \sigma[\vec{x} \mapsto \vec{v}]$ and $\langle s, \sigma'\rangle \to_1^* \sigma'[\vec{x} \mapsto \vec{v}]$, *or*
2. there exists a statement s' such that $\gamma = \langle s', \sigma[\vec{x} \mapsto \vec{v}]\rangle$ and $\langle s, \sigma'\rangle \to_1^* \langle s', \sigma'[\vec{x} \mapsto \vec{v}]\rangle$.

Hint: you should first prove the analogous result for single step reductions (i.e., replace all $\rightarrow_1^*$ with $\rightarrow_1$ in the above), and then use this result to prove the general statement above.

You can assume (cf. Sheet 8, question 3), that for any expression e, if $\forall y \in \mathit{FV}(e), (\sigma(y) = \sigma'(y))$ then $\mathcal{A}[\![e]\!]\sigma = \mathcal{A}[\![e]\!]\sigma'$, and similarly for the evaluation of boolean expressions.

Assignment 6 - Headache of the week: reordering programs

Prove that, for all s_1, s_2, if $\mathit{FV}(s_1) \cap \mathit{FV}(s_2) = \emptyset$, then, for all states σ and σ', if $\langle s_1 ; s_2, \sigma_1 \rangle \rightarrow_1^* \sigma_2$ then $\langle s_2 ; s_1, \sigma_1 \rangle \rightarrow_1^* \sigma_2$.
Hint: you will need to use the results from earlier in this sheet. You will also need the lemma from slide 132 of the lectures, and you may want to use the following two lemmas (the first was proved in your exercise session, and second is a generalisation of question 2 (ii) on this sheet):

Lemma 1: For all states σ_1, σ_2, statements s and configurations γ, if $\langle s, \sigma_1 \rangle \rightarrow_1^* \sigma_2$ then there exist variables $\vec{x}$ and corresponding values $\vec{v}$ such that the $\vec{x}$ are free variables of s (i.e., $\{\vec{x}\} \subseteq \mathit{FV}(s)$) and $\sigma_2 = \sigma_1[\vec{x} \mapsto \vec{v}]$.

Lemma 2: For any two sequences of variables $\vec{x}$ and $\vec{y}$ that are *disjoint* (i.e., $\{\vec{x}\} \cap \{\vec{y}\} = \emptyset$), and for any corresponding two sequences of values $\vec{v}$ and $\vec{v'}$, and for all states σ:
$\sigma[\vec{x} \mapsto \vec{v}][\vec{y} \mapsto \vec{v'}] = \sigma[\vec{y} \mapsto \vec{v'}][\vec{x} \mapsto \vec{v}]$.