P. Müller and D. Basin

Formal Methods and Functional Programming

Session Sheet 12: Small Step Semantics

Assignment 1 (Applying Small-Step Semantics)

Let s be the following statement:

```
y := 1;
while x > 0 do
  y := y * 2;
  x := x - 1
end
```

Task. Let σ be a state with $\sigma(\mathrm{x}) = 2$. Prove that there is a state σ' with $\sigma'(\mathrm{y}) = 4$ such that $\langle s, \sigma \rangle \rightarrow_1^* \sigma'$ using the SOS rules of **IMP**.

Assignment 2 (Equivalence Lemma)

In this exercise, we consider the two lemmas from the lecture that formalize the equivalence of small-step and big-step semantics.

Task 2.1. We *partially* prove the following statement:

$$\forall \sigma, \sigma', s \cdot \vdash \langle s, \sigma \rangle \rightarrow \sigma' \implies \langle s, \sigma \rangle \rightarrow_1^* \sigma'$$

Here, we only consider the $\textsc{Ass}_{\text{NS}}$-rule and the $\textsc{WhT}_{\text{NS}}$-rule; the remaining cases are left for the exercise sheet.

Task 2.2 We partially prove the following statement:

$$\forall \sigma, \sigma', s, k \cdot \langle s, \sigma \rangle \rightarrow_1^k \sigma' \implies \vdash \langle s, \sigma \rangle \rightarrow \sigma'$$

Here, we only consider the $\textsc{Ass}_{\text{SOS}}$-rule and the $\textsc{Seq1}_{\text{SOS}}$, and $\textsc{Seq2}_{\text{SOS}}$-rules; the remaining cases are left for the exercise sheet.

Assignment 3 (`break` Statement)

In Assignment 4 of the *optional exercises* sheet 11, we defined big-step semantics rules for a `break` statement.

Task. Define small-step semantics rules for a `break` statement.

Assignment 4 (Bonus: `do-times` Statements)

In a previous session we defined different NS rules for the IMP extension `do` e `times` s, where e is an arithmetic expression and s a statement.

Task. We now would like to define SOS rules for this type of loop.