Assignment 5
Semantics, WS 2007/08

Prof. Dr. Gert Smolka, Dr. Jan Schwinghammer
www.ps.uni-sb.de/courses/sem-ws07/

Read Chapter 3 of the lecture notes

Exercise 5.1 (Substitution with ρ_0) Determine $S_0\,\epsilon\,t$ for the following terms t. Assume $x = 5, y = 0$ and $z = 3$.

a) $\lambda x.x$,

b) $\lambda xy.x(x\ y)$,

c) $\lambda xyz.y(x\ z)$,

d) $\lambda yzx.z(y\ x)$,

e) $(\lambda x.y)\ y$.

Exercise 5.2 (α-renaming) Find counterexamples that falsify the following statements:

a) $\lambda x.s \sim_\alpha \lambda y.t \iff \exists z:\ [x := z]s \sim_\alpha [y := z]t$

b) $\lambda x.s \sim_\alpha \lambda y.t \iff t \sim_\alpha [x := y]s$

Exercise 5.3 (Big step semantics) Define $\Downarrow \subseteq \mathit{Ter} \times \mathit{Val}$ with inference rules for the untyped call-by-value λ calculus. Give two versions, one for $\mathit{Val} = \mathit{Val}_1$ and one for $\mathit{Val} = \mathit{Val}_2$. Make sure that for all t and v, $t \Downarrow v \iff t \to \cdots \to v$ holds for both versions.

Exercise 5.4 (Higher-order abstract syntax) In this exercise you will implement the untyped call-by-value λ calculus with Val_2 in Standard ML using higher-order abstract syntax (HOAS). Use the following declarations:

```
structure Var :>
sig
  eqtype var
  val var : unit -> var
end =
struct
  type var = int
  val r = ref ~1
  fun var () = (r:= !r +1; !r)
end

open Var
datatype ter = V of var | A of ter * ter | L of ter -> ter
```

a) Write a procedure $eval : ter \to ter$ that evaluates terms.

b) Write procedures $church : int \to ter$ and $dechurch : ter \to int$ such that $church\ n$ yields a value equivalent to c_n, and $dechurch\ t$ yields n if $t \sim c_n$.

c) Give a term $power : ter$ such that $power\ c_m\ c_n$ yields c_{m^n}.

d) Write a procedure $size : ter \to int$ that yields the size of a term.

e) Write a procedure $alpha : ter \to ter \to bool$ that that tests whether two terms are α-equivalent.

f) Write a procedure $free : var \to ter \to bool$ that tests if a variable occurs free in a term.

g) Write a procedure $subst : (var \to ter) \to ter \to ter$ that applies a substitution to a term.