Semantics, WS 2003 – Assignment 2

Prof. Dr. Gert Smolka, Dipl.-Inform. Guido Tack
http://www.ps.uni-sb.de/courses/sem-ws03/

Recommended reading: Types and programming languages, chapters 5 & 6

Exercise 2.1: Substitution-based interpreter We implement the De Bruijn representation of terms as follows:

```
type var = int

datatype term = V of var            (* variable *)
              | L of term           (* lambda abstraction *)
              | A of term * term    (* application *)
              | I of int            (* integer constant *)
              | S                   (* successor *)
```

These terms include constants and the successor function for integers.

(a) Write a procedure `shift : int -> term -> term` that increments all free variables in a term by a given number.

(b) Write a procedure `subst: term -> var -> term -> term` that yields $t[x := t']$.

(c) Write a procedure `reduce: term -> term` that yields the normal form of a closed term if it exists.

(d) Write a procedure `church: int -> term` that yields the Church numeral c_n for $n \geq 0$.

(e) Write a procedure `int: term -> int` that yields for a term t the number n if $t \approx c_n$.

Exercise 2.2: Environment-based interpreter We assume the De Bruijn representation of terms as in the previous exercise and represent closures and environments as follows:

```
datatype env = E of closure list
withtype closure = term * env
```

In addition, they have the following properties:

i) The environment of a closure must bind all free variables of the closure's term.

ii) The terms of the closures of an environment must be values.

(a) Write a procedure `closureToTerm: closure -> term` that yields the closed term represented by a closure.

(b) Write a procedure `reduce: closure -> closure` such that `closureToTerm(reduce(t,E []))` yields the normal form of t if it exists.

Exercise 2.3: Semantic equivalence I Give contexts that show that the following terms are not equivalent:

$$\begin{aligned} A &= \lambda xy.x \\ B &= \lambda xy.y \\ C &= \lambda x.x \end{aligned}$$

Exercise 2.4: Semantic equivalence II Which of the following terms are semantically equivalent?

(a) $(\lambda x.xx)(\lambda x.xx)$

(b) $(\lambda x.xx)(\lambda x.xx)(\lambda x.xx)$

(c) $(\lambda x.f(xx))(\lambda x.f(xx))$

(d) $\mathrm{fix}(\lambda f.f(\lambda x.x))$

(e) c_0

(f) $\mathrm{prd}(\mathrm{scc}\ c_0)$

(g) fls

(h) $\mathrm{prd}\ c_0$

(i) $\lambda xy.y$

(j) $\lambda xyz.yz$

Exercise 2.5: Normal forms and De Bruijn representation Write the normal forms of the following terms in De Bruijn representation:

(a) c_0

(b) plus $c_1 c_3$

(c) prd c_0

(d) fix

(e) fix Id

(f) $(\lambda x.y)[y := x]$ where $x = 0$

(g) $(\lambda xy.xxy)(\lambda y.zy)$ where $z = 0$

(h) $(\lambda xy.y(\lambda z.x)x)(\lambda y.zy)$ where $z = 7$

Exercise 2.6: Recursion operator Give a closed value Z such that

For every value f and every term t there exists a value t' such that

i) $Z\ f\ t \to^* f\ t'\ t$

ii) $Z\ f \to^* t'$

Does `fix` as defined in the book satisfy this property? Prove that your answer is correct.