Semantics, WS 2003 – Assignment 3

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

Recommended reading: Types and programming languages, chapters 8,9

Exercise 3.1: Reference sheet I Write a reference sheet (one page only) for the simply typed λ-calculus with Bool and Nat containing the definitions of

(a) Syntax: $T \in \text{Typ}$, $x \in \text{Var}$, $t \in \text{Ter}$, $v \in \text{Val}$, $nv \in \text{NVal}$

(b) Reduction: $t \to t'$

(c) Typing: $\Gamma \in \text{TE}$, $\Gamma \vdash t : T$

Bring this reference sheet to the lecture and use it for all proofs.

Exercise 3.2: Reference sheet II Write a reference sheet (one page only) for the simply typed λ-calculus with Bool and Nat containing the formulation of the following properties:

(a) Uniqueness

(b) Progress

(c) Substitution

(d) Preservation

(e) Normalization

Remark for each of the properties by which induction they can be proved.

Exercise 3.3: Big step semantics The so-called *big step semantics* is defined as follows: $t \Downarrow v :\Leftrightarrow t \to^* v$

Define the big step semantics of the λ-calculus with Bool and Nat independently of $\to$ by inference rules. Exercise 3.5.17 in the book can give you some hints.

Exercise 3.4: Interpreter for the simply typed λ-calculus We extend the syntax of terms by boolean constants, an IF expression and types in the following way:

```
datatype typ = ARROW of typ * typ
             | BOOL
             | INT

type var = int

datatype term = V of var
              | L of typ * term
              | A of term * term
              | FALSE
              | TRUE
              | IF of term * term * term
              | ZERO
              | SUCC
              | PRED
              | ISZERO
```

Type environments are implemented as a function `gamma : var -> typ`.

(a) Write a procedure `typeof : term -> typ` that returns the type of a term or raises an exception if the term has no type.

(b) Adjust the procedures `shift` and `subst` from the previous assignment sheet to the extended term datatype.

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

Exercise 3.5: Induction Theorem Write the precise formulation of the Induction Theorem. Try to prove it.

Exercise 3.6: Rule induction Write the precise definitions of R, $\hat{R}$, and I_R for ground rule systems. Give the precise formulation of the Rule Induction Theorem. Try to prove it by induction on derivations.

The set I_R can be characterized as the least subset of X satisfying a certain property. State this property in full detail.

Exercise 3.7: Substitution Lemma Here is a statement of the Substitution Lemma:

$\Gamma[x : S] \vdash t : T \ \wedge\ \Gamma \vdash s : S \quad \Longrightarrow \quad \Gamma \vdash t[x := s] : T$

The variables Γ, x, s, t, T, S are all universally quantified over their canonical domains.

The lemma can be proved by induction over the rules defining the typing relation (see the book, page 106). State the set P that the proof is using (with respect to the Rule Induction Proposition). Make sure that you describe P in full detail but as concisely as possible.

Exercise 3.8: Reversed Type Preservation Prove the following statement:

$\neg(t \to t' \ \wedge\ \Gamma \vdash t' : T \quad \Longrightarrow \quad \Gamma \vdash t : T)$

Intuitively, the statement says that type preservation doesn't hold from right to left.