Assignment 14
Semantics, WS 2007/08

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

Read Chapter 6 of the lecture notes, and *A Theory of Primitive Objects (Untyped and First-order Systems)* by M. Abadi and L. Cardelli, available from http://lucacardelli.name/Papers/PrimObj1stOrder.A4.pdf

Exercise 14.1 (Intuitionistic proofs) Find intuitionistic proofs for the following formulas:

a) $0 \to X$

b) $(\forall X.S) \to S^X_T$

c) $S^X_T \to \exists X.S$, where $\exists X.S \;:=\; \neg\forall X.\neg S$

d) $(S + T) \to (T + S)$

Exercise 14.2 (Polymorphic lists) Express the following objects in F_ω:

$$
\begin{aligned}
\mathit{list} &: * \to * \\
\mathit{nil} &: \forall X.\ \mathit{list}\ X \\
\mathit{cons} &: \forall X.\ X \to \mathit{list}\ X \to \mathit{list}\ X \\
\mathit{foldl} &: \forall X\ Y.\ (X \to Y \to Y) \to Y \to \mathit{list}\ X \to Y \\
\mathit{length} &: \forall X.\ \mathit{list}\ X \to \mathit{nat}
\end{aligned}
$$

Exercise 14.3 (Lists over nat) Consider an ADT that implements lists over *nat*:

$$
\begin{aligned}
\mathit{list} &: * \\
\mathit{nil} &: \mathit{list} \\
\mathit{cons} &: \mathit{nat} \to \mathit{list} \to \mathit{list} \\
\mathit{foldl} &: \mathit{list} \to \forall Z\ (\mathit{nat} \to Z \to Z) \to Z \to Z
\end{aligned}
$$

a) Represent the signature of the ADT as a type function.

b) Write a procedure *impl* that implements the ADT. Use the objects from exercise 14.2.

Exercise 14.4 (Polymorphic lists) Consider an ADT that implements polymorphic lists as in Exercise 14.2 (omit *length*).

a) Represent the signature of the ADT as a type function.

b) Write a procedure *impl* that implements the ADT. Use the objects from exercise 14.2.

Exercise 14.5 (Untyped natural numbers objects) Extend the object-oriented natural numbers by

a) a method $\mathsf{add} = \varsigma(x)\lambda(n)\ldots$ that adds another number to the number represented by the object;

b) a method $\mathsf{mult} = \varsigma(x)\lambda(n)\ldots$ that multiplies another number to the number represented by the object;

Feel free to use booleans and lambda notation as shorthand whenever necessary.

Exercise 14.6 (Operational semantics of the sigma calculus)

a) Make sure that you can state the operational semantics of the ς-calculus: what are the values, evaluation contexts, and proper reduction rules?

b) Give an equivalent big-step semantics.

Exercise 14.7 (Simple types)

a) Make sure that you can state the key typing rules of the system.

b) Show that $\emptyset \vdash [l = \varsigma(x{:}[l{:}A])x.l] : [l{:}A]$, for any type A.

c) Assume that the ς-calculus has been extended with a base type *int*, and let $P_1 := [\mathsf{x} : \mathit{int}]$ and $P_2 := [\mathsf{x} : \mathit{int}, \mathsf{y} : \mathit{int}]$ be the types of 1- and 2-dimensional points, resp. Show $\emptyset \vdash [\mathsf{x} = 1] : P_1$ and $\emptyset \vdash [\mathsf{x} = 1, \mathsf{y} = 1] : P_2$.

d) Assume that arithmetic operations $+, (\,)^2$ and *sqrt* are given, with their usual types. Convince yourself that, for $B := [\mathsf{p} : P_2, \mathsf{dist} : \mathit{int}]$,

$$\emptyset \vdash [\mathsf{p} = [\mathsf{x} = 1, \mathsf{y} = 1],\ \mathsf{dist} = \varsigma(s{:}B)\mathit{sqrt}((s.\mathsf{p}.\mathsf{x})^2 + (s.\mathsf{p}.\mathsf{y})^2)] : B$$

e) Use (c) and (d) to show that (covariant) subtyping in depth is not sound:

$$\frac{A_i <: B_i \quad \forall\, i = 1 \ldots n}{[l_i{:}{A_i}^{i=1\ldots n+k}] <: [l_i{:}{B_i}^{i=1\ldots n}]} \qquad \text{(wrong!!!)}$$

Exercise 14.8 (Encoding simply typed lambda calculus) Show that the encoding of lambda terms as objects carries over to the typed case, where $S \to T$ is translated to $(S \to T)^* = [\mathsf{arg} : S^*, \mathsf{val} : T^*]$. More precisely, prove that the encoding maps well-typed terms of the simply typed lambda calculus to well-typed terms of ς-calculus. You may use, without proof, that a substitution lemma holds for ς-calculus, analogous to the one for λ-calculus.

Is the statement also true for simply typed lambda calculus with subtyping?