Semantics, WS 2003 – Assignment 13

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

Recommended reading: TAPL, chapters 23, 24, 26

Exercise 13.1: Uninhabited Types Give a closed type T of System F for which there is no term t such that $\varnothing \vdash t : T$.

Exercise 13.2: Data structures in System F Represent the following types and values in System F:

(a) *Bool*, *true*

(b) *Unit*, ()

(c) *Nat*, 0, *succ*

(d) *List* X, *nil*, *cons*

(e) $X \times Y$, (x, y), $\#1$

(f) $X + Y$, *inr*

Exercise 13.3: ADTs

(a) Give the type of clients for ADTs with signature X, T.

(b) Give the type of ADTs with signature X, T.

(c) Give a term that implements an ADT with signature X, T and implementation T_0, t_I.

Exercise 13.4: Generic ADTs In the lecture you have seen how pairs can be provided in System F by means of a generic ADT.

(a) Give the type of a generic ADT that provides variants of type $X + Y$.

(b) Give a term that implements the generic ADT for variants.

Exercise 13.5: Existential Types Existential types are a derived notion in System F. Assume that the abbreviations

$\{\exists X, T\} \rightsquigarrow \forall Z.(\forall X.T \to Z) \to Z$

$\{*T_0, t\}\ as\ \{\exists X, T\} \rightsquigarrow \lambda Z.\lambda f : \forall X.T \to Z.f\ T_0\ t$

$let\ \{*X, x\} = t\ in\ t' \rightsquigarrow t\ T'\ (\lambda X.\lambda x : T.t')$ with T' type of t'

are provided syntactically.

(a) How must the definition of types, terms, and values be extended to support the new syntax?

(b) Give the typing and reduction rules needed for the new syntax.

Exercise 13.6: $F_{<:}$

(a) Give the typing rules for $\lambda X <: T.t$ and $t\ T$ in $F_{<:}$.

(b) Try to understand the structure of the type environments Γ for $F_{<:}$. Try to formalize type environments for $F_{<:}$.