Assignment 11
Semantics, WS 2013/14

Prof. Dr. Gert Smolka, Steven Schäfer

www.ps.uni-saarland.de/courses/sem-ws13/

Read in the lecture notes: Chapter 5

For a presentation of System T and F you should consult the lecture notes (Chapter 5.2 and 5.3). There is an alternative presentation in Harper's "Practical Foundations for Programming Languages" Chapters 9 and 20. Send your Coq solutions for exercises 11.7 - 11.9 to schaefer@ps.uni-saarland.de until Thursday 12:00pm. The solutions will not be graded. You might want to reuse the definition of substitution for ULC.

Exercise 11.1 It is possible to formulate System T with simpler terms as follows:

$$s, t ::= O \mid S \mid \mathsf{prec} \mid x \mid st \mid \lambda x{:}A.s$$

In this version the operator *prec* relies on abstractions obtained with λ. There are advantages and disadvantages to this approach.

a) Express the term $\mathsf{prec}\ s\ t\ (xy.u)$ of the old system in the new system.

b) Express the operator prec of the new system in the old system.

c) Give the typing rules for the terms S and prec in the new system.

d) Give the reduction rules for redexes obtained with prec in the new system.

e) Explain why the unique type property is lost in the new system.

f) State the canonical form property of the new system.

Exercise 11.2 Express primitive recursion in F. Use Coq to type check your solution. Also Use Coq to compute with your solution.

Exercise 11.3 Give a type preserving translation of System T into System F. In particular show how to translate a T type A into an F type $\overline{A}$ and a T term s into an F term $\overline{s}$ such that:

- $x_1 : A_1, \ldots, x_n : A_n \vdash s : B$ in T implies $x_1 : \overline{A_1}, \ldots, x_n : \overline{A_n} \vdash \overline{s} : \overline{A}$ in F.
- $s \equiv t$ in T implies $\overline{s} \equiv \overline{t}$ in F.

Exercise 11.4 Subject reduction for F requires the side condition $\Delta \vdash \Gamma$ for non-empty contexts Γ:

$$\Delta \vdash \Gamma \to \Delta\,\Gamma \vdash s : A \to s \succ s' \to \Delta\,\Gamma \vdash s' : A$$

Find a counterexample that shows that the side condition is needed.

Exercise 11.5 Typing judgements can always be reduced to typing judgements for empty contexts. Let $\Delta = \{X_1, \ldots, X_n\}$, $\Gamma = \{x_1 : A_1, \ldots, x_m : A_m\}$, and $\Delta \vdash \Gamma$. Given a term s and a type A find a term t and a type B such that $\Delta\,\Gamma \vdash s : A$ if and only if $\vdash t : B$.

Exercise 11.6 Explain why $\vdash \mathsf{U} : \mathsf{U}$ does not hold in the single sorted presentation of F.

Exercise 11.7 (Coq) We define the syntax of the simply typed lambda calculus in de Bruijn style.

$$A, B ::= X \mid A \to B$$
$$s, t, u ::= x \mid s\,t \mid \lambda A.s$$

Note that in $\lambda A.s$, there is a new bound variable in s.

a) Define substitution for terms.
b) Define the typing relation.
c) Define the reduction relation.

Exercise 11.8 (Coq) We define the syntax of de Bruijn System T as follows:

$$A, B ::= nat \mid A \to B$$
$$s, t, u ::= O \mid S\ s \mid prec\ s\ t\ u \mid x \mid s\,t \mid \lambda A.s$$

Note that in $\lambda A.s$, there is a new bound variable in s, and that in $prec\ s\ t\ u$ there are two bound variables in u.

a) Define substitution for terms.
b) Define the typing relation.
c) Define the reduction relation.

Exercise 11.9 (Coq) We encode the syntax of System F as follows:

$$A, B, s, t ::= x \mid s\,t \mid \lambda A.s \mid A \to B \mid \forall.A \mid U$$

In particular, we use a combined syntax for types and terms and only a single lambda. To encode the type binder we introduce a universe U. Type abstraction is encoded as $\lambda U.s$.

a) Define substitution.
b) Define the typing relation.
c) Define the reduction relation.