Homework 8b

COS 441
Princeton University
Due 12/8/98

Earlier, we determined rules for subtyping for different types. Here we give a formal presentation of these rules. We begin with the definition of type constraint systems, C, which provide assumptions for the use of type variables, and the definition of $C \vdash \tau\colon \mathit{TYPE}$ (read as, "under the assumptions C on type variables, τ is a type).

A type constraint is a declaration of the form $t{:}\mathit{TYPE}$ or $t <: \tau$, where t is a type variable.

Definition 0.1 *A type constraint system is defined as follows:*

1. *$\emptyset$ is a type constraint system.*
2. *If C is a type constraint system and t does not occur in C, then $C \cup \{t\colon \mathit{TYPE}\}$ is a type constraint system.*
3. *If C is a type constraint system, $C \vdash \tau\colon \mathit{TYPE}$, and t does not occur in C, then $C \cup \{t <: \tau\}$ is a type constraint system.*

Notice that the definition of a type constraint system depends on the system for proving type expressions are really types, so the previous and following definitions are actually mutually recursive.

Let L be a collection of labels for record fields.

$$C \vdash N\colon \mathit{TYPE}$$

$$C \cup \{t\colon \mathit{TYPE}\} \vdash t\colon \mathit{TYPE}$$

$$C \cup \{t <: \tau\} \vdash t\colon \mathit{TYPE}$$

$$\frac{C \vdash \sigma\colon \mathit{TYPE} \qquad C \vdash \tau\colon \mathit{TYPE}}{C \vdash \sigma \to \tau\colon \mathit{TYPE}}$$

$$\frac{C \vdash \sigma\colon \mathit{TYPE} \qquad C \vdash \tau\colon \mathit{TYPE}}{C \vdash \sigma * \tau\colon \mathit{TYPE}}$$

$$\frac{C \vdash \tau_i\colon \mathit{TYPE}\ \mathit{and}\ l_i \in \mathcal{L}\ \mathit{for}\ 1 \le i \le n}{C \vdash \{l_1{:}\,\tau_1, \ldots, l_n{:}\,\tau_n\}\colon \mathit{TYPE}}$$

$$\frac{C \vdash \tau\colon \mathit{TYPE}}{C \vdash \mathtt{Ref}\ \tau\colon \mathit{TYPE}}$$

$$\frac{C \cup \{t\colon \mathit{TYPE}\} \vdash \tau\colon \mathit{TYPE}}{C \vdash \forall t.\tau\colon \mathit{TYPE}}$$

Existentials
$$\frac{C \cup \{t\colon TYPE\} \vdash \tau\colon TYPE}{C \vdash \exists t.\tau\colon TYPE}$$

It is easy to see that if the type expression τ contains a type variable t which is not mentioned in C, then $C \vdash \tau\colon TYPE$ fails.

The following are the subtyping rules for the typed lambda calculus:

Hyp
$$C \cup \{t <\colon \tau\} \vdash t <\colon \tau$$

Reflexivity
$$\frac{C \vdash \tau\colon TYPE}{C \vdash \tau <\colon \tau}$$

Transitivity
$$\frac{C \vdash \sigma <\colon \tau \qquad C \vdash \tau <\colon \gamma}{C \vdash \sigma <\colon \gamma}$$

Functions
$$\frac{C \vdash \sigma <\colon \hat{\sigma} \qquad C \vdash \hat{\tau} <\colon \tau}{C \vdash \hat{\sigma} \to \hat{\tau} <\colon \sigma \to \hat{\tau}}$$

Pairs
$$\frac{C \vdash \hat{\sigma} <\colon \sigma \qquad C \vdash \hat{\tau} <\colon \tau}{C \vdash \hat{\sigma} * \hat{\tau} <\colon \sigma * \tau}$$

Records
$$\frac{C \vdash \hat{\tau}_i <\colon \tau_i \quad for\ 1 \leq i \leq n, \quad C \vdash \hat{\tau}_i\colon TYPE \quad for\ n+1 \leq i \leq n+m}{C \vdash \{l_1\colon \hat{\tau}_1, \ldots, l_n\colon \hat{\tau}_n, \ldots, l_{n+m}\colon \hat{\tau}_{n+m}\} <\colon \{l_1\colon \tau_1, \ldots, l_n\colon \tau_n\}}$$

Poly − functions
$$\frac{C \cup \{t\colon TYPE\} \vdash \hat{\tau} <\colon \tau}{C \vdash \forall t.\hat{\tau} <\colon \forall t.\tau}$$

Existentials
$$\frac{C \cup \{t\colon TYPE\} \vdash \hat{\tau} <\colon \tau}{C \vdash \exists t.\hat{\tau} <\colon \exists t.\tau}$$

Notice that there is no specific rule for reference types since a reference type may only be a subtype of itself.

Exercises:

1. Provide a convincing argument for the *poly-functions* subtyping rule. I.e., show that any element of type $\forall t.\hat{\tau}$ can masquerade as an element of type $\forall t.\tau$, if $\hat{\tau}$ can masquerade as τ. Your argument need not be formal, but it should be convincing.

2. Provide a convincing argument for the *existential* subtyping rule.

3. Suppose we introduce a new term representing bounded parametric polymorphism: $\Lambda(t <\colon \sigma).e$. The intuition is that this term is similar to the regular parametric polymorphism, but may only be applied to terms whose type is a subtype of σ. The type of such terms may be written as $\forall(t <\colon \sigma).\tau$. Please write a subtyping rule for types of this form. Provide a convincing argument for its correctness.