Foundations of Software Science (ソフトウェア基礎科学) /
Foundations of Computer Software (ソフトウェア基礎)
Exercises (<u>no need to submit</u>)
October 30, 2009
Eijiro Sumii (instructor)

Fill in the blanks below after the following definitions.

<u>Definitions</u>

The syntax of λ-terms M, N, ... is given by:

M	::=	i	(integers)
	|	x	(variables)
	|	$\lambda x.M$	(functions)
	|	$M_1\ M_2$	(function applications)
	|	(M_1,M_2)	(pairs)
	|	$\mathrm{fst}(M)$	(first projections)
	|	$\mathrm{snd}(M)$	(second projections)
	|	$\mathrm{Left}(M)$	(left variants)
	|	$\mathrm{Right}(M)$	(right variants)
	|	case M of $\mathrm{Left}(x) \Rightarrow N_1 \mid \mathrm{Right}(y) \Rightarrow N_2$	(case branches)

The syntax of <u>types</u> τ, σ, ... is:

τ	::=	int	(integer type)
	|	$\tau_1 \to \tau_2$	(function types)
	|	$\tau_1 \times \tau_2$	(pair types)
	|	$\tau_1 + \tau_2$	(variant types)

A <u>type environment</u> Γ, Δ, ... is a (partial) map from variables to types. It is often written like

$$\Gamma = x_1{:}\tau_1, x_2{:}\tau_2, \ldots, x_n{:}\tau_n$$

when the domain of Γ is $\{ x_1,x_2,\ldots,x_n \}$ and $\Gamma(x_i) = \tau_i$ for $i = 1,2,\ldots,n$.

A <u>type judgment</u> $\Gamma \vdash M : \tau$ is the smallest relation among type environments,

λ-terms and types that satisfies the following typing rules:

- Rule T-Int: $\Gamma \vdash i : \mathrm{int}$, for any type environment Γ and integer i (the quantification "for any ..." will be omitted in the other rules)
- Rule T-Var: $\Gamma \vdash x : \tau$ if $\Gamma(x) = \tau$

- Rule T-Fun: $\Gamma \vdash \lambda x.M : \tau_1 \to \tau_2$ if $\Gamma, x{:}\tau_1 \vdash M : \tau_2$
- Rule T-App: $\Gamma \vdash M_1\ M_2 : \tau_2$ if $\Gamma \vdash M_1 : \tau_1 \to \tau_2$ and $\Gamma \vdash M_2 : \tau_1$

- Rule T-Pair: $\Gamma \vdash (M_1,M_2) : \tau_1 \times \tau_2$ if $\Gamma \vdash M_1 : \tau_1$ and $\Gamma \vdash M_2 : \tau_2$
- Rule T-Fst: $\Gamma \vdash \mathrm{fst}(M) : \tau_1$ if $\Gamma \vdash M : \tau_1 \times \tau_2$
- Rule T-Snd: $\Gamma \vdash \mathrm{snd}(M) : \tau_2$ if $\Gamma \vdash M : \tau_1 \times \tau_2$

- Rule T-Left: $\Gamma \vdash \mathrm{Left}(M) : \tau_1 + \tau_2$ if $\Gamma \vdash M : \tau_1$
- Rule T-Right: $\Gamma \vdash \mathrm{Right}(M) : \tau_1 + \tau_2$ if $\Gamma \vdash M : \tau_2$
- Rule T-Case: $\Gamma \vdash (\mathrm{case}\ M\ \mathrm{of}\ \mathrm{Left}(x) \Rightarrow N_1 \mid \mathrm{Right}(y) \Rightarrow N_2) : \tau$ if $\Gamma \vdash M : \tau_1 + \tau_2$ and $\Gamma, x{:}\tau_1 \vdash N_1 : \tau$ and $\Gamma, y{:}\tau_2 \vdash N_2 : \tau$

Exercise 1

Let us prove $\Gamma \vdash \lambda x.\lambda y.x : \tau_1 \to \tau_2 \to \tau_1$ (for any Γ, τ_1, and τ_2). By rule T-Fun, it suffices to prove $\Gamma, x{:}\tau_1 \vdash \lambda y.x : \tau_2 \to \tau_1$. Thus, again by rule T-Fun, it suffices to prove Γ, $x{:}\tau_1$, ______________ $\vdash x :$ ______________. This follows from rule ___________.

Exercise 2

Let us prove $\Gamma \vdash \lambda f.\lambda x.f(fx) : (\tau \to \tau) \to \tau \to \tau$. By rule T-Fun, it suffices to prove Γ, ________________ $\vdash \lambda x.f(fx) :$ ________________. Thus, again by rule T-Fun, it suffices to prove Γ, ___________________________ $\vdash f(fx) :$ ___________. To prove this by rule T-App, it suffices to show

Γ, __________________________ $\vdash f :$ ________________ and

Γ, _____________________________ $\vdash fx :$ __________________. The former is immediate from rule __________. To prove the latter by rule __________, it suffices to show Γ, __________________________ $\vdash f :$ ________________ and Γ, ____________________________ $\vdash x :$ _________________, both of which are immediate from rule __________.

<u>Exercise 3</u>

Prove $\Gamma \vdash \lambda f.\lambda g.\lambda x.g(fx) : (\tau_1 \to \tau_2) \to (\tau_2 \to \tau_3) \to (\tau_1 \to \tau_3)$.

Exercise 4

Prove that there are no Γ and τ such that $\Gamma \vdash (\lambda x.xx)(\lambda x.xx) : \tau$.