Homework 9: Types: Rules and Inference

CIS 352: Programming Languages

16 March 2018, Version 1

Administrivia

- Turn in Part I in the CIS 352 submissions box. If you trade ideas with another student, document this in your cover sheet.
- Turn in Part II via Blackboard.
- For the remaining problems, turn them in via Blackboard. Include: *(i)* the source files, *(ii)* the transcripts of test runs, and *(iii)* your cover sheet.

Grading Criteria

- The homework is out of 100 points.
- Unless otherwise stated, each programming problem is ≈ 70% correctness and ≈ 30% testing.
- Omitting your name(s) in the source code looses you 5 points.

Part I: Written Problems

❖ ***Problem 1 (8 points)*** ❖

Perform the following substitutions, correctly!

(a) $\big((\lambda x.\ (x\ z))\ x\big)\ \big[(x\ y)/z\big]$

(b) $\big(\lambda y.\big((\lambda x.(z\ x))\ \big(\lambda y.((z\ y)\ x)\big)\big)\big)\ \big[(x\ y)/z\big]$

❖ ***Problem 2 (16 points)*** ❖

Give full type derivations for the following using the LFP$^+$ typing rules given in class (and page 3).

(a) $\vdash (\textbf{if } !\ell > 4 \textbf{ then } 9 \textbf{ else } 6) + 11 \colon \textbf{int}$

(b) $x : \sigma,\ f : (\sigma \to \tau) \vdash (f\ x) : \tau$

(c) $\vdash \lambda x.\lambda y.(y\ x) : \sigma \to (\sigma \to \tau) \to \tau$

(d) $\vdash \lambda f.\lambda x.(f\ x) : (\textbf{int}{\to}\textbf{bool}) \to \textbf{int}{\to}\textbf{bool}$

(e) $\vdash \lambda f.\lambda g.\lambda x.(f\ (g\ x)) : (\textbf{int}{\to}\textbf{bool}) \to (\textbf{int}{\to}\textbf{int}) \to \textbf{int}{\to}\textbf{bool}$

❖ ***Problem 3 (32 points)*** ❖

(a) Come up with a workable typing rule for the **let** construct.[1] I.e., something along the lines of

$$\frac{??? \vdash e_1 : ??? \qquad ??? \vdash e_2 : ???}{\Gamma \vdash \textbf{let}\ \ x = e_1\ \ \textbf{in}\ \ e_2 : \tau}$$

[1] **Obvious Hint:** $(\textbf{let}\ \ x = e_1\ \ \textbf{in}\ \ e_2) \equiv ((\lambda x.e_2)\ e_1)$.

(b) Give a full type derivation for:

$$x \colon \textbf{int} \vdash \textbf{let}\ \ p = \lambda y.(x+y)\ \ \textbf{in}\ \ (\textbf{let}\ \ x = 3\ \ \textbf{in}\ \ (p\,x)) \colon \textbf{int}$$

(c) Give a full type derivation for:

$$y\colon \mathbf{int} \vdash \mathbf{let}\;\; p = \lambda x.\lambda y.(x > y)\;\; \mathbf{in}\;\; (\mathbf{let}\;\; x = y\;\; \mathbf{in}\;\; (p\,x))\colon \mathbf{int}{\to}\mathbf{bool}$$

(d) Pitts (2002, page 66) defines the **letrec** construct. Come up with a workable typing rule for the **letrec** construct.

Part II: Programming Problems

For this part of the homework you should go to Tony Field's

`http://wp.doc.ic.ac.uk/ajf/type-inference/`,

grab a copy of the *specification* and the *template file*, and solve the problems in parts I, II, and III of the specification.[2] Do part IV for extra credit and glory. Here is a schedule of my point assignments for these problems. (These are roughly proportional to Field's marks.)

[2] In working these problems, I found `lookup` (from `Data.List`) and `maybe` (from `Data.Maybe`) quite handy. Also, do not be afraid of using `case` expressions!

Part	Problem	Points
I	1	5 points
	2	3 points
	3	5 points
II		16 points
III	1	3 points
	2	12 points
IV	1	4 extra credit points
	2	2 extra credit points
	3	0 points (i.e., just glory)

Testing The file `tests09.hs` has tests I cobbled together for the above. You *do not* need to invent original tests for this assignment.

References

A. Pitts. Lecture notes on semantics of programming languages: For part IB of the Cambridge CS tripos. Technical report, University of Cambridge, 2002. URL `http://www.cl.cam.ac.uk/teaching/2001/Semantics/`.

LFP^+ typing rules

Notes:

- We use σ and τ as meta-variables over types.
- $\Gamma, x : \tau \quad \equiv \quad \Gamma \cup \{\, x : \tau \,\}$
- $\vdash e : \tau \quad \equiv \quad \emptyset \vdash e : \tau.$
- $iop \in \{\, +, -, * \,\}$
- $cop \in \{\, =, \neq, <, >, \leq, \geq \,\}$

$$\textit{:-int:}\ \frac{}{\Gamma \vdash n \colon \mathbf{int}}\ (n \in \mathbb{Z})$$

$$\textit{:-bool:}\ \frac{}{\Gamma \vdash b \colon \mathbf{bool}}\ (b \in \mathbb{B})$$

$$\textit{:-loc:}\ \frac{}{\Gamma \vdash \ell \colon \mathbf{loc}}\ (\ell \in \mathbb{L})$$

$$\textit{:-iop:}\ \frac{\Gamma \vdash e_1 \colon \mathbf{int} \quad \Gamma \vdash e_2 \colon \mathbf{int}}{\Gamma \vdash e_1\ iop\ e_2 \colon \mathbf{int}}$$

$$\textit{:-cop:}\ \frac{\Gamma \vdash e_1 \colon \mathbf{int} \quad \Gamma \vdash e_2 \colon \mathbf{int}}{\Gamma \vdash e_1\ cop\ e_2 \colon \mathbf{bool}}$$

$$\textit{:-skip:}\ \frac{}{\Gamma \vdash \mathbf{skip} : \mathbf{cmd}}$$

$$\textit{:-get:}\ \frac{\Gamma \vdash e \colon \mathbf{loc}}{\Gamma \vdash !e \colon \mathbf{int}}$$

$$\textit{:-set:}\ \frac{\Gamma \vdash e_1 \colon \mathbf{loc} \quad \Gamma \vdash e_2 \colon \mathbf{int}}{\Gamma \vdash e_1 \leftarrow e_2 \colon \mathbf{cmd}}$$

$$\textit{:-if:}\ \frac{\Gamma \vdash e_o \colon \mathbf{bool} \quad \Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \mathbf{if}\ e_0\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 : \tau}$$

$$\textit{:-seq:}\ \frac{\Gamma \vdash e_1 \colon \mathbf{cmd} \quad \Gamma \vdash e_2 \colon \mathbf{cmd}}{\Gamma \vdash e_1;\ e_2 \colon \mathbf{cmd}}$$

$$\textit{:-whi:}\ \frac{\Gamma \vdash e_1 \colon \mathbf{bool} \quad \Gamma \vdash e_2 \colon \mathbf{cmd}}{\Gamma \vdash \mathbf{while}\ e_1\ \mathbf{do}\ e_2 \colon \mathbf{cmd}}$$

$$\textit{:-var:}\ \frac{}{\Gamma,\ x : \tau \vdash x : \tau}$$

$$\textit{:-fn:}\ \frac{\Gamma,\ x : \tau \vdash e' : \tau'}{\Gamma \vdash \lambda x . e' : \tau \to \tau'}$$

$$\textit{:-app:}\ \frac{\Gamma \vdash e_1 : \tau \to \tau' \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash (e_1\ e_2) : \tau'}$$

$$\textit{:-rec:}\ \frac{\Gamma,\ x : \tau \vdash e : \tau}{\Gamma \vdash \mathbf{rec}\, x . e : \tau}$$

A sample typing derivation

$$\textit{fn:}\ \dfrac{\textit{fn:}\ \dfrac{+:\ \dfrac{\textit{var:}\ \dfrac{}{x \colon \mathbf{int},\ y \colon \mathbf{int} \vdash x \colon \mathbf{int}} \qquad *:\ \dfrac{\textit{int:}\ \dfrac{}{x \colon \mathbf{int},\ y \colon \mathbf{int} \vdash 3 \colon \mathbf{int}} \qquad \textit{var:}\ \dfrac{}{x \colon \mathbf{int},\ y \colon \mathbf{int} \vdash y \colon \mathbf{int}}}{x \colon \mathbf{int},\ y \colon \mathbf{int} \vdash (3 * y) \colon \mathbf{int}}}{x \colon \mathbf{int},\ y \colon \mathbf{int} \vdash (x + (3 * y)) \colon \mathbf{int}}}{x \colon \mathbf{int} \vdash \lambda y . (x + (3 * y)) \colon \mathbf{int} \to \mathbf{int}}}{\vdash \lambda x . \lambda y . (x + (3 * y)) \colon \mathbf{int} \to \mathbf{int} \to \mathbf{int}}$$