CS 599 A1: Assignment 3

Due: Friday, March 7, 2025

Total: 100 pts

Ankush Das

- This assignment is due on the above date and it must be submitted electronically on Gradescope.
- Please use the template provided on the course webpage to typeset your assignment and please include your name and BU ID in the Author section (above).
- Although it is not recommended, you can submit handwritten answers that are scanned as a PDF and clearly legible.
- You will be provided a tex file, named asgn3.tex. It contains an environment called solution. Please enter your solutions inside these environments.

Derivations and Programs

This assignment will be all about writing derivations and their corresponding programs. For your convenience, I have provided the typing rules of both languages.

Constructive Logic

Syntax

$$
\begin{array}{rl}
\text{Expressions} & e ::= \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid \mathsf{match}\ x\ \mathsf{with}\ \{\, \mathsf{inl}(y) \rightarrow e \mid \mathsf{inr}(z) \rightarrow e \,\} \mid \langle e, e \rangle \mid \mathsf{let}\ \langle y, z \rangle = x\ \mathsf{in}\ e \\
& \quad \mid \ \lambda x.\, e \mid \mathsf{let}\ y = f\ e\ \mathsf{in}\ e \\
\text{Types} & \tau ::= \tau \vee \tau \mid \tau \wedge \tau \mid \tau \Rightarrow \tau
\end{array}
$$

Typing Rules

$$
\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathsf{inl}(e) : \tau_1 \vee \tau_2}\ \vee\mathrm{R}_1
\qquad\qquad
\frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathsf{inr}(e) : \tau_1 \vee \tau_2}\ \vee\mathrm{R}_2
$$

$$
\frac{\Gamma, y : \tau_1 \vdash e_1 : \tau \qquad \Gamma, z : \tau_2 \vdash e_2 : \tau}{\Gamma, x : \tau_1 \vee \tau_2 \vdash \mathsf{match}\ x\ \mathsf{with}\ \{\, \mathsf{inl}(y) \rightarrow e_1 \mid \mathsf{inr}(z) \rightarrow e_2 \,\} : \tau}\ \vee\mathrm{L}
$$

$$
\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \wedge \tau_2}\ \wedge\mathrm{R}
\qquad\qquad
\frac{\Gamma, y : \tau_1, z : \tau_2 \vdash e : \tau}{\Gamma, x : \tau_1 \wedge \tau_2 \vdash \mathsf{let}\ \langle y, z \rangle = x\ \mathsf{in}\ e : \tau}\ \wedge\mathrm{L}
$$

$$
\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x.\, e : \tau_1 \Rightarrow \tau_2}\ \Rightarrow\mathrm{R}
\qquad\qquad
\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau_2 \vdash e : \tau}{\Gamma, f : \tau_1 \Rightarrow \tau_2 \vdash \mathsf{let}\ x = f\ e_1\ \mathsf{in}\ e_2 : \tau}\ \Rightarrow\mathrm{L}
$$

Linear Logic

Syntax

$$
\begin{array}{rl}
\text{Expressions} & P, Q ::= x.\mathsf{inl} \,;\, P \mid x.\mathsf{inr} \,;\, P \mid \mathsf{case}\ x\ (\mathsf{inl} \Rightarrow P \mid \mathsf{inr} \Rightarrow Q) \\
& \quad \mid \ \mathsf{send}\ x\ y \,;\, P \mid y \leftarrow \mathsf{recv}\ x \,;\, P \mid \mathsf{wait}\ x \,;\, P \mid \mathsf{close}\ x \\
\text{Types} & A, B ::= A \oplus B \mid A \,\&\, B \mid A \otimes B \mid A \multimap B \mid \mathbf{1}
\end{array}
$$

Typing Rules

$$\frac{\Delta \vdash P :: (x : A_1)}{\Delta \vdash (x.\mathsf{inl}\, ;\, P) :: (x : A_1 \oplus A_2)} \oplus R_1 \qquad \frac{\Delta \vdash P :: (x : A_2)}{\Delta \vdash (x.\mathsf{inr}\, ;\, P) :: (x : A_1 \oplus A_2)} \oplus R_2$$

$$\frac{\Delta, x : A_1 \vdash Q_1 :: (z : C) \qquad \Delta, x : A_2 \vdash Q_2 :: (z : C)}{\Delta, x : A_1 \oplus A_2 \vdash (\mathsf{case}\ x\ (\mathsf{inl} \Rightarrow Q_1 \mid \mathsf{inr} \Rightarrow Q_2)) :: (z : C)} \oplus L$$

$$\frac{\Delta \vdash P_1 :: (x : A_1) \qquad \Delta \vdash P_2 :: (x : A_2)}{\Delta \vdash (\mathsf{case}\ x\ (\mathsf{inl} \Rightarrow P_1 \mid \mathsf{inr} \Rightarrow P_2)) :: (x : A_1 \,\&\, A_2)} \,\&\, R$$

$$\frac{\Delta, x : A_1 \vdash Q :: (z : C)}{\Delta, x : A_1 \,\&\, A_2 \vdash (x.\mathsf{inl}\, ;\, Q) :: (z : C)} \,\&\, L_1 \qquad \frac{\Delta, x : A_2 \vdash Q :: (z : C)}{\Delta, x : A_1 \,\&\, A_2 \vdash (x.\mathsf{inr}\, ;\, Q) :: (z : C)} \,\&\, L_2$$

$$\frac{\Delta \vdash P :: (x : B)}{\Delta, y : A \vdash (\mathsf{send}\ x\ y\, ;\, P) :: (x : A \otimes B)} \otimes R \qquad \frac{\Delta, y : A, x : B \vdash Q :: (z : C)}{\Delta, x : A \otimes B \vdash (y \leftarrow \mathsf{recv}\ x\, ;\, Q) :: (z : C)} \otimes L$$

$$\frac{\Delta, y : A \vdash P :: (x : B)}{\Delta \vdash (y \leftarrow \mathsf{recv}\ x\, ;\, P) :: (x : A \multimap B)} \multimap R \qquad \frac{\Delta, x : B \vdash Q :: (z : C)}{\Delta, x : A \multimap B, y : A \vdash (\mathsf{send}\ x\ y\, ;\, Q) :: (z : C)} \multimap L$$

$$\frac{}{\cdot \vdash (\mathsf{close}\ x) :: (x : \mathbf{1})} \mathbf{1}R \qquad \frac{\Delta \vdash Q :: (z : C)}{\Delta, x : \mathbf{1} \vdash (\mathsf{wait}\ x\, ;\, Q) :: (z : C)} \mathbf{1}L \qquad \frac{}{x : A \vdash (y \leftrightarrow x) :: (y : A)} \mathsf{id}$$

$$\frac{\mathsf{decl}\ f : \overline{y' : A'} \vdash (x : A) \in \Sigma \qquad \Delta, x : A \vdash Q :: (z : C)}{\Delta, \overline{y : A'} \vdash (x \leftarrow f\ \overline{y}\, ;\, Q) :: (z : C)} \mathsf{def}$$

1 Inference in Constructive Logic [50 pts]

For the following propositions, determine if they are true or false. If the proposition is true, do a (single) derivation with the corresponding program (using the rules above). If the proposition is false, briefly explain why.

1. $(\tau_1 \wedge (\tau_2 \vee \tau_3)) \Rightarrow (\tau_1 \wedge \tau_2) \vee (\tau_1 \wedge \tau_3)$
2. $((\tau_1 \Rightarrow \tau_2) \Rightarrow \tau_3) \Rightarrow (\tau_1 \Rightarrow (\tau_2 \Rightarrow \tau_3))$
3. $\tau \Rightarrow (\tau \wedge \tau)$
4. $(\tau \wedge \tau) \Rightarrow \tau$
5. $\tau_1 \Rightarrow \tau_2 \Rightarrow (\tau_1 \wedge \tau_2)$

2 Inference in Linear Logic [50 pts]

For the following propositions, determine if they are true or false. If the proposition is true, do a (single) derivation with the corresponding program (using the rules above). If the proposition is false, briefly explain why.

1. $A \,\&\, (B \otimes C) \multimap (A \,\&\, B) \otimes (A \,\&\, C)$
2. $((A \multimap B) \multimap C) \multimap (A \multimap (B \multimap C))$
3. $(A \multimap A) \multimap A$
4. $A \multimap A \multimap (A \otimes A)$
5. $(A \multimap (B \otimes C)) \multimap (A \multimap B) \otimes (A \multimap C)$