1 Simple Types

Task 1 (L3.1, 10 points) Fill in the blanks in the following typing judgments so the resulting judgment holds, or indicate there is no way to do so. You do not need to justify your answer or supply a typing derivation, and the types do not need to be “most general” in any sense. Remember that the function type constructor associates to the right, so that $\tau \to \sigma \to \rho = \tau \to (\sigma \to \rho)$.

(i) $\Gamma \vdash y \cdot x : \alpha$

   $y : \beta \to \alpha, x : \beta \vdash y \cdot x : \alpha$

(ii) $\Gamma \vdash x \cdot x : \alpha$

   No such context and type exist.

(iii) $\Gamma \vdash \lambda f. \lambda g. \lambda x. (f x) (g x) : (\alpha \to \alpha) \to (\alpha \to \alpha) \to \alpha$

   No such context and type exist.

(iv) $\Gamma \vdash (\lambda z. z) (\lambda x. \lambda y. \lambda p. p x y) : \alpha \to (\alpha \to \beta \to \gamma) \to \gamma$

(v) $\Gamma \vdash \lambda f. \lambda g. \lambda x. (f x) (g x) : (\alpha \to \alpha) \to (\alpha \to \alpha) \to (\alpha \to \alpha)$
2 Proof by Rule Induction

Task 2 (L4.1.1 & L4.1.2, 30 points) If we have two judgments defined simultaneously (like $e$ normal and $e$ neutral we often need to prove properties about them by simultaneous induction. In simultaneous induction you have multiple induction hypotheses and if the premise of a rule comes from a different judgment, you may apply the appropriate induction hypothesis to it. In proving property 2 below, make a note if you needed a simple or a simultaneous induction.

1. In each case below, give an example of an expressions $e$ and type $\tau$ with $\cdot \vdash e : \tau$ and also the stated property, or indicate no such expression and type exist. You do not need to justify your answer further (no need for typing derivations or proofs).

(i) $\cdot \vdash e \equiv \tau$ and also $\cdot \vdash e \Rightarrow \tau$

There are no $e$ and $\tau$ for which $\cdot \vdash e \Rightarrow \tau$ holds. (The reason is given below.)

(ii) $\cdot \vdash e \equiv \tau$ but not $\cdot \vdash e \Rightarrow \tau$

A candidate for $e : \tau$ is $e = \lambda x. x$ and $\tau = \alpha \rightarrow \alpha$. We first need to show that $\cdot \vdash \lambda x. x : \alpha \rightarrow \alpha$:

$$
\begin{array}{c}
\frac{x : \alpha \vdash x : \alpha}{\cdot \vdash \lambda x. x : \alpha \rightarrow \alpha}
\end{array}
$$

We can form a derivation for $\cdot \vdash \lambda x. x \equiv \alpha \rightarrow \alpha$ too:

$$
\begin{array}{c}
\frac{x : \alpha \vdash x \Rightarrow \alpha}{\cdot \vdash \lambda x. x \equiv \alpha \rightarrow \alpha}
\end{array}
$$

Moreover as we will explain below, $\cdot \not\vdash \lambda x. x \Rightarrow \alpha \rightarrow \alpha$.

(iii) $\cdot \vdash e \Rightarrow \tau$ but not $\cdot \vdash e \equiv \tau$

There are no $e$ and $\tau$ for which $\cdot \vdash e \Rightarrow \tau$ holds. (The reason is given below.)

(iv) Neither $\cdot \vdash e \equiv \tau$ nor $\cdot \vdash e \Rightarrow \tau$
A candidate for \( e : \tau \) is \((\lambda x. x)(\lambda y. y) : (\alpha \to \alpha)\). We first need to show \( \vdash (\lambda x. x)(\lambda y. y) : (\alpha \to \alpha) \):

\[
\begin{align*}
  x : \alpha \to \alpha & \vdash x : \alpha \to \alpha \\
  \vdash (\lambda x. x) : (\alpha \to \alpha) \to (\alpha \to \alpha) \\
  \vdash (\lambda y. y) : \alpha \to \alpha \\
  \vdash (\lambda x. x)(\lambda y. y) : \alpha \to \alpha
\end{align*}
\]

By the proof given below we know that \( \nabla \vdash (\lambda x. x)(\lambda y. y) \Rightarrow \alpha \to \alpha \). We only need to show \( \nabla \nabla \vdash (\lambda x. x)(\lambda y. y) \Rightarrow \alpha \to \alpha \):

\[
\begin{align*}
  \nabla \vdash (\lambda x. x)(\lambda y. y) \Rightarrow \alpha \to \alpha \\
  \vdash (\lambda x. x) : (\alpha \to \alpha) \Rightarrow (\alpha \to \alpha) \\
  \vdash (\lambda y. y) : \alpha \to \alpha \\
  \vdash (\lambda x. x)(\lambda y. y) \Rightarrow \alpha \to \alpha
\end{align*}
\]

It remains to prove that if \( \vdash e : \tau \) then there is no \( \tau' \) such that \( \vdash e \Rightarrow \tau' \). We can prove this, for example, by induction on the structure of the derivation of \( \vdash e : \tau \) (see below). We could also prove it even more easily by induction on the structure of \( e \), not even using the assumption that \( \vdash e : \tau \).

There are three cases to consider.

Case:

\[
\begin{align*}
  x_1 : \tau_1 & \vdash e_2 : \tau_2 \\
  \vdash \lambda x_1.e_2 : \tau_1 \to \tau_2
\end{align*}
\]

Here \( e = \lambda x_1.e_2, \tau = \tau_1 \to \tau_2 \) and \( \Gamma = \cdot \). We can see that there is no rule to infer \( \vdash \lambda x_1.e_2 \Rightarrow \tau_1 \to \tau_2 \) since \( e \) is neither an application, nor a variable. In other words, by inversion on the rules on page L4.6, \( \nabla \vdash \lambda x_1.e_2 \Rightarrow \tau_1 \to \tau_2 \).

Case:

\[
\begin{align*}
  x : \tau & \in (\cdot) \\
  \vdash x : \tau
\end{align*}
\]

Here \( e = x \) and \( \Gamma = \cdot \). This case is not applicable since \( x \in \tau \not\in (\cdot) \).

Case:

\[
\begin{align*}
  \vdash e_1 : \tau_2 \to \tau & \quad \vdash e_2 : \tau_2 \\
  \vdash e_1 e_2 : \tau
\end{align*}
\]

Here \( e = e_1 e_2 \) and \( \Gamma = \cdot \).

\[
\begin{align*}
  \vdash e_1 e_2 & \Rightarrow \tau \\
  \vdash e_1 & \Rightarrow \tau_2' \to \tau \text{ for some } \tau_2' \\
  \nabla \vdash e_1 & \Rightarrow \tau_2' \to \tau \\
  \nabla \vdash e_1 e_2 & \Rightarrow \tau
\end{align*}
\]

By inversion on page L4.6 rules (syn/app)
2. Prove that the bidirectional typing rules are sound, that is, we verify or synthesize only correct types.

(i) If $\Gamma \vdash e \equiv \tau$ then $\Gamma \vdash e : \tau$ and $e$ normal.

(ii) If $\Gamma \vdash e \Rightarrow \tau$ then $\Gamma \vdash e : \tau$ and $e$ neutral.

We prove statements (i) and (ii) by simultaneous induction on the structure of the derivations of $\Gamma \vdash e \equiv \tau$ and $\Gamma \vdash e \Rightarrow \tau$. There are four cases to consider:

Case:

\[
\frac{\Gamma, x_1 : \tau_1 \vdash e_2 \equiv \tau_2}{\Gamma \vdash \lambda x_1. e_2 \equiv \tau_1 \rightarrow \tau_2} \quad \text{chk/\(\lambda\)m}
\]

where $e = \lambda x_1. e_2$, and $\tau = \tau_1 \rightarrow \tau_2$.

(1) $\Gamma, x_1 \vdash e_2 : \tau_2$ By ind. hyp. on the premise (statement (i))

(2) $e_2$ normal By ind. hyp. on the premise (statement (i))

$\Gamma \vdash \lambda x_1. e_2 : \tau_1 \rightarrow \tau_2$ By rule $\lambda$m from line (1)

$\lambda x_1. e_2$ normal By rule norm/$\lambda$m from line (2)

Case:

\[
\frac{\Gamma \vdash e \Rightarrow \tau}{\Gamma \vdash e \equiv \tau} \quad \text{chk/syn}
\]

(1) $e$ neutral By ind. hyp. on the premise (statement (ii))

$\Gamma \vdash e : \tau$ By ind. hyp. on the premise (statement (ii))

$e$ normal By rule norm/neut from line (1)

Case:

\[
\frac{x : \tau \in \Gamma}{\Gamma \vdash x \Rightarrow \tau} \quad \text{syn/var}
\]

where $e = x$.

(1) $x : \tau \in \Gamma$ Premise

$\Gamma \vdash x : \tau$ By rule var from Line (1)

$x$ neutral By rule neut/var

Case:

\[
\frac{\Gamma \vdash e_1 \Rightarrow \tau_2 \rightarrow \tau \quad \Gamma \vdash e_2 \equiv \tau_2}{\Gamma \vdash e_1 \ e_2 \Rightarrow \tau} \quad \text{syn/app}
\]

where $e = e_1 \ e_2$.  

 ASSIGNMENT 2          DUE TUESDAY, SEPTEMBER 24, 2019
The Simply Typed $\lambda$-Calculus
Sample Solution

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) $\Gamma \vdash e_1 : \tau_2 \to \tau$</td>
<td>By ind. hyp. on the 1st premise (statement (ii))</td>
</tr>
<tr>
<td>(2) $e_1 neutral$</td>
<td>By ind. hyp. on the 1st premise (statement (ii))</td>
</tr>
<tr>
<td>(3) $\Gamma \vdash e_2 : \tau_2$</td>
<td>By ind. hyp. on the 2nd premise (statement (i))</td>
</tr>
<tr>
<td>(4) $e_2 normal$</td>
<td>By ind. hyp. on the 2nd premise (statement (i))</td>
</tr>
<tr>
<td>$\Gamma \vdash e_1 e_2 : \tau$</td>
<td>By rule lapp from lines (1), (3)</td>
</tr>
<tr>
<td>$e_1 e_2 normal$</td>
<td>By rule neut/app from lines (2), (4)</td>
</tr>
</tbody>
</table>

**Task 3 (L4.2, 20 points)** Prove the following theorems.

1. If $e nf$ then $e normal$.

We prove this part by induction on the structure of the derivation of $e nf$. There are three cases to consider:

**Case:**

\[
\frac{e_1 nf}{\lambda x. e_1 nf} \quad \text{nf/lam}
\]

where $e = \lambda x.e_1$.

(1) $e_1 normal$

$\lambda x. e_1 normal$

By rule norm/lam from line (1)

**Case:**

\[
\frac{\_}{x nf} \quad \text{nf/var}
\]

where $e = x$.

(1) $x neutral$

$x normal$

By rule neut/var

**Case:**

\[
\frac{e_1 \neq \lambda_\_ \quad e_1 nf \quad e_2 nf}{e_1 e_2 nf} \quad \text{nf/app}
\]

where $e = e_1; e_2$.

(1) $e_1 \neq \lambda_\_\neq \lambda_\_\neq \lambda_\_\neq \lambda_\_

(2) $e_1 normal$

(3) $e_2 normal$

(4) $e_1 e_2 neutral$

$e_1 e_2 normal$

By rule neut/app from lines (2), (3)

By rule norm/neut from line (4)
* In this step, we apply inversion on the rule inferred $e_1 \ normal$. Since $e_1 \neq \lambda_-$, the only possible rule to infer $e_1 \ normal$, is norm/neut. So we have a derivation for $e_1 \ neutral$.

2. If $e \ normal$ then $e \ nf$.

We first generalize this statement as follows:

(i) If $e \ neutral$ then $e \ nf$.
(ii) If $e \ normal$ then $e \ nf$.

We prove the generalized statement by simultaneous induction on the structure of derivations of $e \ neutral$ and $e \ normal$. There are four cases to consider:

**Case:**

\[
\frac{e_1 \ normal}{\lambda x. e_1 \ normal} \] 

\[
\text{norm/lam}
\]

where $e = \lambda x. e_1$.

(1) $e_1 \ nf$ 

By ind. hyp. on $e_1 \ normal$

(2) $\lambda x. e_1 \ nf$ 

By rule norm/lam from line (1)

**Case:**

\[
\frac{e \ neutral}{e \ normal} \] 

\[
\text{norm/neut}
\]

$e \ nf$ 

By ind. hyp. on $e \ neutral$

**Case:**

\[
\frac{e_1 \ neutral \ e_2 \ normal}{e_1 e_2 \ neutral} \] 

\[
\text{neut/app}
\]

where $e = e_1 e_2$.

(1) $e_1 \ neutral$ 

1st premise

(2) $e_1 \ nf$ 

By ind. hyp. on the 1st premise

(3) $e_2 \ nf$ 

By ind. hyp. on the 2nd premise

(4) $e_1 = \lambda x. e_3$ 

Assumption

(4) No derivation for $e_1 \ neutral$ 

By inversion on page L4.5 rules

(4) $e_1 \neq \lambda_-$ 

By contradiction [lines (1), (4)]

(4) $e_1 e_2 \ nf$ 

By rule nf/app from lines (2), (3), (4)
### Case:

\[
\begin{array}{c}
\text{neutral} \\
\underline{x} \\
\text{neut/var}
\end{array}
\]

where \( e = x \).

\( x \ nf \) By rule nf/var