

# EXERCISES

## CHAPTER 3

SEAN LI <sup>1</sup>

### 1. Reducted

---

**Definition** Some rules for reference.

$$\begin{array}{c} \frac{x : \sigma \in \Gamma \quad \Gamma \text{ is a } \lambda 2 \text{ context}}{\Gamma \vdash x : \sigma} \text{ (T-Var)} \qquad \frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M N : \tau} \text{ (T-App)} \\[10pt] \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \rightarrow \tau} \text{ (T-Abst)} \qquad \frac{\alpha \in \mathbb{T}_2 \quad \text{FV } \alpha \subseteq \text{dom } \Gamma}{\alpha : * \in \Gamma} \text{ (T-Form)} \\[10pt] \frac{\Gamma \vdash M : \Pi_{\alpha:*.} A \quad \Gamma \vdash B : *}{\Gamma \vdash M B : A [\alpha := B]} \text{ (T2-App)} \qquad \frac{\Gamma, \alpha : * \vdash M : A}{\Gamma \vdash \lambda \alpha : *. M : \Pi_{\alpha:*.} A} \text{ (T2-Abst)} \end{array}$$

In this document, convention is that all type judgements in a proof tree, unless stated otherwise, is derived from a single and unique  $\lambda 2$  context per tree. Multiple conclusions might be drawn on a single line from usage of the same inference rule for compactness. Eg:

ex 1.  $\alpha, \beta : *$  **T-Form**

Is shorthand for

ex 1.  $\Gamma \vdash \alpha : *$  **T-Form**

ex 2.  $\Gamma \vdash \beta : *$  **T-Form**

---

### Problem

(3.1) How many  $\lambda 2$  contexts consisting of four and only four declarations

- (1)  $\Gamma \vdash \alpha : *$       (2)  $\Gamma \vdash \beta : *$   
 (3)  $\Gamma \vdash f : \alpha \rightarrow \beta$     (4)  $\Gamma \vdash x : \alpha$

*Solution.* The last two declarations depend on the first two. Therefore this is an easy combinatorics problem:  $2! \times 2! = 4$  contexts:

- 1 – 2 – 3 – 4    1 – 2 – 4 – 3  
 2 – 1 – 3 – 4    2 – 1 – 4 – 3

### Problem

(3.2) Give a full derivation in  $\lambda 2$  to show the following type term is legal:

$$M \equiv \lambda \alpha, \beta, \gamma : *. \lambda f : \alpha \rightarrow \beta. \lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x)$$

*Solution.*

1.	$\alpha : *$	<b>Bound</b>
2.	$\beta : *$	<b>Bound</b>
3.	$\gamma : *$	<b>Bound</b>
4.	$f : \alpha \rightarrow \beta$	<b>Bound</b>
5.	$g : \beta \rightarrow \gamma$	<b>Bound</b>
6.	$x : \alpha$	<b>Bound</b>
7.	$\alpha, \beta, \gamma : *$	<b>T-Form</b>
8.	$\alpha \rightarrow \beta, \beta \rightarrow \gamma : *$	<b>T-Form</b>
9.	$f : \alpha \rightarrow \beta, x : \alpha$	<b>T-Var</b>
10.	$f x : \beta$	<b>8,8 T-App</b>
11.	$g : \beta \rightarrow \gamma$	<b>T-Var</b>
12.	$g (f x) : \gamma$	<b>11,10 T-App</b>
13.	$\lambda x : \alpha. g (f x) : \alpha \rightarrow \gamma$	<b>12 T-Abst</b>
14.	$\lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x) : (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$	<b>13 T-Abst</b>
	$\lambda f : \alpha \rightarrow \beta. \lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x)$	
15.	$: (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$	<b>14 T-Abst</b>

16.	$\frac{\lambda\gamma : *. \lambda f : \alpha \rightarrow \beta. \lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x) : \Pi\gamma : *. (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma}{\lambda\beta, \gamma : *. \lambda f : \alpha \rightarrow \beta. \lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x) : \Pi\beta, \gamma : *. (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma}}$	<b>15 T2-Abst</b>
17.		<b>16 T2-Abst</b>
18.		
	$\lambda\alpha, \beta, \gamma : *. \lambda f : \alpha \rightarrow \beta. \lambda g : \beta \rightarrow \gamma. \lambda x : \alpha. g (f x) : \Pi\alpha, \beta, \gamma : *. (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$	<b>17 T2-Abst</b>

### Problem

(3.3 a) Given  $M$  in 3.2, and a context  $\Gamma$  such that

$$\Gamma \vdash \text{nat} : *$$

$$\Gamma \vdash \text{bool} : *$$

$$\Gamma \vdash \text{succ} : \text{nat} \rightarrow \text{nat}$$

$$\Gamma \vdash \text{even} : \text{nat} \rightarrow \text{bool}$$

Prove  $M \text{ nat nat bool succ even}$  is legal.

*Solution.* Proof by deriving the term's type.

*Proof.*

1.	$M : \Pi\alpha, \beta, \gamma : *. (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$	<b>T-Var</b>
2.	$\text{nat}, \text{bool} : *$	<b>T-Form</b>
3.	$M \text{ nat} : \Pi\beta, \gamma : *. (\text{nat} \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \text{nat} \rightarrow \gamma$	<b>2,1 T2-App</b>
4.	$M \text{ nat nat} : \Pi\gamma : *. (\text{nat} \rightarrow \text{nat}) \rightarrow (\text{nat} \rightarrow \gamma) \rightarrow \text{nat} \rightarrow \gamma$	<b>2,3 T2-App</b>
5.		
	$M \text{ nat nat bool} : (\text{nat} \rightarrow \text{nat}) \rightarrow (\text{nat} \rightarrow \text{bool}) \rightarrow \text{nat} \rightarrow \text{bool}$	<b>2,3 T2-App</b>
6.	$\text{succ} : \text{nat} \rightarrow \text{nat}, \text{even} : \text{nat} \rightarrow \text{bool}$	<b>T-Var</b>
7.	$M \text{ nat nat bool succ} : (\text{nat} \rightarrow \text{bool}) \rightarrow \text{nat} \rightarrow \text{bool}$	<b>6,5 T-App</b>
8.	$M \text{ nat nat bool succ even} : \text{nat} \rightarrow \text{bool}$	<b>6,7 T-App</b>

■

### Problem

(3.3 b.i) Prove  $\lambda x : \text{nat}. \text{even} (\text{succ } x)$  via 3.3 a.

*Solution.* The result of beta reduction on the term in 3.3 a is what we are proving.

*Proof.*

$$\begin{aligned}
 & M \text{ nat nat bool succ even} \\
 & \equiv (\lambda \alpha, \beta, \gamma, f, g. \lambda x : \alpha. g (f x)) \text{ nat nat bool succ even} \\
 & \xrightarrow[\beta]{\rightarrow} (\lambda f : \text{nat} \rightarrow \text{nat}. \lambda g : \text{nat} \rightarrow \text{bool}. \lambda x : \text{nat}. g (f (x))) \text{ succ even} \\
 & \xrightarrow[\beta]{\rightarrow} (\lambda x : \text{nat}. \text{even} (\text{succ } x))
 \end{aligned}$$

By the subject reduction lemma,  $\lambda x : \text{nat}. \text{even} (\text{succ } x) : \text{nat} \rightarrow \text{bool}$ , thus is legal. ■

### Problem

(3.3 b.ii) Prove  $\lambda x : \text{nat}. \text{even} (\text{succ } x)$  via derivation in the context provided in 3.3 a.

*Solution.*

*Proof.*

1.	$\text{nat}, \text{bool} : *$	<b>T-Form</b>
2.	$x : \text{nat}$	<b>Bound</b>
3.	$\text{succ} : \text{nat} \rightarrow \text{nat}$	<b>T-Var</b>
4.	$x : \text{nat}$	<b>T-Var</b>
5.	$\text{succ } x : \text{nat}$	<b>3,4 T-App</b>
6.	$\text{even} : \text{nat} \rightarrow \text{bool}$	<b>T-Var</b>
7.	$\text{even} (\text{succ } x) : \text{bool}$	<b>6,5 T-App</b>
8.	$\lambda x : \text{nat}. \text{even} (\text{succ } x) : \text{nat} \rightarrow \text{bool}$	<b>7 T-Abst</b>

■

### Problem

(3.4) Give a shorthand (omit T-Var and T-Form) derivation in  $\lambda 2$  to show the following term is valid in  $\Gamma \equiv \text{nat} : *, \text{bool} : *$

$$(\lambda \alpha, \beta : *. \lambda f : \alpha \rightarrow \alpha. \lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x))) \text{ nat bool}$$

*Solution.*

*Proof.*

1.	$\alpha, \beta : *$	<b>Bound</b>
2.	$f : \alpha \rightarrow \alpha$	<b>Bound</b>
3.	$g : \alpha \rightarrow \beta$	<b>Bound</b>
4.	$x : \alpha$	<b>Bound</b>
5.	$f x : \alpha$	<b>*, T-App</b>
6.	$f (f x) : \alpha$	<b>*,5 T-App</b>
7.	$g (f (f x)) : \beta$	<b>*,6 T-App</b>
8.	$\lambda x : \alpha. g (f (f x)) : \alpha \rightarrow \beta$	<b>7 T-Abst</b>
9.	$\lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x)) : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$	<b>8 T-Abst</b>
10.	$\lambda f : \alpha \rightarrow \alpha. \lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x)) : (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$	<b>9 T-Abst</b>
11.		
	$\lambda \alpha, \beta : *. \lambda f : \alpha \rightarrow \alpha. \lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x)) : \Pi \alpha, \beta : *. (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$	<b>10 T2-Abst</b>
12.		
	$(\lambda \alpha, \beta : *. \lambda f : \alpha \rightarrow \alpha. \lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x))) \text{ nat} : \Pi \beta : *. (\text{nat} \rightarrow \text{nat}) \rightarrow (\text{nat} \rightarrow \beta) \rightarrow \text{nat} \rightarrow \beta$	<b>*,11 T2-App</b>
13.		
	$(\lambda \alpha, \beta : *. \lambda f : \alpha \rightarrow \alpha. \lambda g : \alpha \rightarrow \beta. \lambda x : \alpha. g (f (f x))) \text{ nat bool} : \Pi \beta : *. (\text{nat} \rightarrow \text{nat}) \rightarrow (\text{nat} \rightarrow \text{bool}) \rightarrow \text{nat} \rightarrow \text{bool}$	<b>*,12 T2-App</b>

■

### Problem

(3.5 a) Let  $\perp \equiv \Pi \alpha : *. \alpha$ . Prove  $\perp$  is legal.

*Solution.* Here a notion called kind checking is introduced. This has not yet been discussed in this book (?)

*Proof.*

1.  $\alpha : *$  **Bound**
2.  $\frac{}{\alpha : *}$  **T-Form**
3.  $\Pi \alpha : *. \alpha : * \rightarrow *$  **T2-Abst**

■

### Problem

(3.5 b) Consider the context  $\Gamma \equiv \beta : *, x : \perp$ . Find an inhabitant of type  $\beta$  under  $\Gamma$ .

*Solution.*  $x \beta$  is. Because  $x$  is of second-order type, it must be parametric to a type, thus  $x$  is of form  $\lambda \alpha : *. M$  where  $\Gamma, \alpha : * \vdash M : \alpha$ .

*Proof.*

1.  $x : \Pi \alpha : *. \alpha$  **T-Var**
2.  $\beta : *$  **T-Form**
3.  $x \beta : \beta$  **1,2 T2-App**

■

### Problem

(3.5 c) Give three inhabitants of  $\beta \rightarrow \beta$  in  $\beta$ -nf under  $\Gamma$  in 3.5 b.

*Solution.*

1.  $\lambda y : \beta. y$ .

*Proof.*

1.  $y : \beta$  **Bound**
2.  $\frac{}{y : \beta}$  **T-Var**
3.  $\lambda y : \beta. y : \beta \rightarrow \beta$  **2 T-Abst**

■

2.  $\lambda y : \beta. x \beta$ .

*Proof.*

1.	$y : \beta$	<b>Bound</b>
2.	$x : \Pi \alpha : * . \alpha$	<b>T-Var</b>
3.	$\beta : *$	<b>T-Form</b>
4.	$x \beta : \beta$	<b>2,3 T2-App</b>
5.	$\lambda y : \beta. x \beta : \beta \rightarrow \beta$	<b>4 T-Abst</b>

■

3.  $\lambda y : \beta. x (\beta \rightarrow \beta) y.$

*Proof.*

1.	$y : \beta$	<b>Bound</b>
2.	$x : \Pi \alpha : * . \alpha$	<b>T-Var</b>
3.	$\beta \rightarrow \beta : *$	<b>T-Form</b>
4.	$x (\beta \rightarrow \beta) : \beta \rightarrow \beta$	<b>2,3 T2-App</b>
5.	$y : \beta$	<b>T-Var</b>
6.	$x (\beta \rightarrow \beta) y : \beta$	<b>4,5 T-App</b>
7.	$\lambda y : \beta. x (\beta \rightarrow \beta) y : \beta \rightarrow \beta$	<b>5 T-Abst</b>

■

### Problem

(3.5 d) Prove that the following terms inhabit the same type in  $\Gamma$ :

$$\lambda f : \beta \rightarrow \beta \rightarrow \beta. f (x \beta)(x \beta)$$

$$x ((\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta)$$

*Solution.* We simply derive the types.

*First Term.*

1.	$f : \beta \rightarrow \beta \rightarrow \beta$	<b>Bound</b>
2.	$f : \beta \rightarrow \beta \rightarrow \beta$	<b>T-Var</b>
3.	$x : \Pi \alpha : * . \alpha$	<b>T-Var</b>
4.	$\beta : *$	<b>T-Form</b>
5.	$x \beta : \beta$	<b>3,4 T2-App</b>
6.	$f (x \beta) : \beta \rightarrow \beta$	<b>2,5 T-App</b>

7.  $\frac{}{f(x\ \beta)(x\ \beta) : \beta}$  **6,5 T-App**
8.  $\lambda f : \beta \rightarrow \beta \rightarrow \beta. f(x\ \beta)(x\ \beta) : (\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta$  **6 T-Abst**

■

*Second Term.*

1.  $(\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta : *$  **T-Form**
2.  $x : \Pi \alpha : *. \alpha$  **T-Var**
3.  $x((\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta) : (\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta$  **2,1 T2-App**

■

The two terms were shown to both inhabit  $(\beta \rightarrow \beta \rightarrow \beta) \rightarrow \beta$ .

### Problem

(3.6 a) Find inhabitant of type

$$\Pi \alpha, \beta : *. (\text{nat} \rightarrow \alpha) \rightarrow (\alpha \rightarrow \text{nat} \rightarrow \beta) \rightarrow \text{nat} \rightarrow \beta$$

In context  $\Gamma \equiv \text{nat} : *$ .

*Solution.*

$$\lambda \alpha, \beta : *. \lambda x : \text{nat} \rightarrow \alpha. \lambda y : (\alpha \rightarrow \text{nat} \rightarrow \beta). \lambda z : \text{nat}. y(x\ z)\ z$$

*Proof.*

1.  $\alpha, \beta : *$  **Bound**
2.  $\text{nat} \rightarrow \alpha : *$  **T-Form**
3.  $x : \text{nat} \rightarrow \alpha$  **Bound**
4.  $\alpha \rightarrow \text{nat} \rightarrow \beta : *$  **T-Form**
5.  $y : \alpha \rightarrow \text{nat} \rightarrow \beta$  **Bound**
6.  $\text{nat} : *$  **Bound**
7.  $z : \text{nat}$  **Bound**
8.  $y : \alpha \rightarrow \text{nat} \rightarrow \beta$  **T-Var**
9.  $x : \text{nat} \rightarrow \alpha$  **T-Var**
10.  $z : \text{nat}$  **T-Var**
11.  $x\ z : \alpha$  **9,10 T-App**
12.  $y(x\ z) : \text{nat} \rightarrow \beta$  **8,11 T-App**
13.  $y(x\ z)\ z : \beta$  **12,10 T-App**



14.	$\lambda z : \mathbf{nat}. y (x z) z : \mathbf{nat} \rightarrow \beta$	<b>13 T-Abst</b>
15.	$\lambda y : \alpha \rightarrow \mathbf{nat} \rightarrow \beta. \lambda z : \mathbf{nat}. y (x z) z$ $: (\alpha \rightarrow \mathbf{nat} \rightarrow \beta) \rightarrow \mathbf{nat} \rightarrow \beta$	<b>14 T-Abst</b>
16.	$\lambda x : \mathbf{nat} \rightarrow \alpha. y : \alpha \rightarrow \mathbf{nat} \rightarrow \beta. \lambda z : \mathbf{nat}. y (x z) z$ $: (\mathbf{nat} \rightarrow \alpha) \rightarrow (\alpha \rightarrow \mathbf{nat} \rightarrow \beta) \rightarrow \mathbf{nat} \rightarrow \beta$	<b>15 T2-Abst</b>
17.		
	$\lambda \alpha, \beta : *. x : \mathbf{nat} \rightarrow \alpha. y : \alpha \rightarrow \mathbf{nat} \rightarrow \beta. \lambda z : \mathbf{nat}. y (x z) z$ $: \Pi \alpha, \beta : *. (\mathbf{nat} \rightarrow \alpha) \rightarrow (\alpha \rightarrow \mathbf{nat} \rightarrow \beta) \rightarrow \mathbf{nat} \rightarrow \beta$	<b>16 T2-Abst</b>

■

### Problem

(3.6 b) Find inhabitant of type

$$\Pi \delta : *. ((\alpha \rightarrow \gamma) \rightarrow \delta) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \delta$$

In context  $\Gamma \equiv \alpha : *, \beta : *, \gamma : *$

*Solution.*

$$\lambda \delta : *. \lambda x : (\alpha \rightarrow \gamma) \rightarrow \delta. \lambda y : \alpha \rightarrow \beta. \lambda z : \beta \rightarrow \gamma. x (\lambda u : \alpha. z (y u))$$

A derivation in shorthand will be given (omitting T-Form / T-Var)

*Proof.*

1.	$\delta : *$	<b>Bound</b>
2.	$x : (\alpha \rightarrow \gamma) \rightarrow \delta$	<b>Bound</b>
3.	$y : \alpha \rightarrow \beta$	<b>Bound</b>
4.	$z : \beta \rightarrow \gamma$	<b>Bound</b>
5.	$u : \alpha$	<b>Bound</b>
6.	$y u : \beta$	<b>*, T-App</b>
7.	$z (y u) : \gamma$	
	<b>*,6 T-App</b>	
8.	$\lambda u : \alpha. z (y u) : \alpha \rightarrow \gamma$	<b>7 T-Abst</b>
9.	$x (\lambda u : \alpha. z (y u)) : \delta$	<b>8 T-Abst</b>
10.	$\lambda z : \beta \rightarrow \gamma. x (\lambda u : \alpha. z (y u)) : (\beta \rightarrow \gamma) \rightarrow \delta$	<b>9 T-Abst</b>

$$11. \quad \left| \begin{array}{l} \lambda y : \alpha \rightarrow \beta. \lambda z : \beta \rightarrow \gamma. x (\lambda u : \alpha. z (y u)) \\ : (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \delta \end{array} \right.$$

**10 T-Abst**

$$12. \quad \left| \begin{array}{l} \lambda x : (\alpha \rightarrow \gamma) \rightarrow \delta. \lambda y : \alpha \rightarrow \beta. \lambda z : \beta \rightarrow \gamma. x (\lambda u : \alpha. z (y u)) \\ : ((\alpha \rightarrow \gamma) \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \delta \end{array} \right.$$

**11 T-Abst**

13.

$$\lambda \delta : *. \lambda x : (\alpha \rightarrow \gamma) \rightarrow \delta. \lambda y : \alpha \rightarrow \beta. \lambda z : \beta \rightarrow \gamma. x (\lambda u : \alpha. z (y u)) \\ : \Pi \delta : *. ((\alpha \rightarrow \gamma) \rightarrow \delta) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \delta$$

**12 T2-Abst**

■

### Problem

(3.6 c) Find inhabitant of type

$$\Pi \alpha, \beta, \gamma : *. (\alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$$

In the empty context

*Solution.*

$$\lambda \alpha, \beta, \gamma : *. \lambda f : (\alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma). \lambda x : \alpha. f x (\lambda u : \beta. x)$$

*Proof.*

1.	$\alpha, \beta, \gamma$	<b>Bound</b>
2.	$f : \alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma$	<b>Bound</b>
3.	$x : \alpha$	<b>Bound</b>
4.	$f x : (\beta \rightarrow \alpha) \rightarrow \gamma$	<b>*, * T-App</b>
5.	$u : \beta$	<b>Bound</b>
6.	$x : \alpha$	<b>T-Var</b>
7.	$\lambda u : \beta. x : \beta \rightarrow \alpha$	<b>6 T-Abst</b>
8.	$f x (\lambda u : \beta. x) : \gamma$	<b>4,7 T-App</b>
9.	$\lambda x : \alpha. f x (\lambda u : \beta. x) : \alpha \rightarrow \gamma$	<b>8 T-Abst</b>
	$\lambda f : \alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma. \lambda x : \alpha. f x (\lambda u : \beta. x)$	
10.	$: (\alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$	<b>9 T-Abst</b>

11.

$$\lambda\alpha, \beta, \gamma : *. \lambda f : \alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma. \lambda x : \alpha. f x (\lambda u : \beta. x)$$

$$: \Pi\alpha, \beta, \gamma : *. (\alpha \rightarrow (\beta \rightarrow \alpha) \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$$

**10 T2-Abst**

■

### Problem

(3.7) Let  $\perp \equiv \Pi\alpha : *. \alpha$  and context  $\Gamma \equiv \alpha : *, \beta : *, x : \alpha \rightarrow \perp, f : (\alpha \rightarrow \alpha) \rightarrow \alpha$ . Give a derivation that succesively calculate an inhabitant of  $\alpha$  and  $\beta$ , both in context  $\Gamma$ .

*Solution.* Have  $M : \alpha := f (\lambda n : \alpha. n)$ . Then  $\Gamma \vdash x M \beta : \beta$ .

*Typing M.*

1.  $f : (\alpha \rightarrow \alpha) \rightarrow \alpha$  **T-Var**
2.  $n : \alpha$  **Bound**
3.  $\frac{}{\lambda n : \alpha. n : \alpha \rightarrow \alpha}$  **T-Var**
4.  $\lambda n : \alpha. n : \alpha \rightarrow \alpha$  **3 T-Abst**
5.  $f (\lambda n : \alpha. n) : \alpha$  **1,4 T-App**

■

*Typing  $x M \beta$ .*

1.  $M : \alpha$  **T-Var**
2.  $x : \alpha \rightarrow \Pi\alpha : *. \alpha$  **T-Var**
3.  $M x : \Pi\alpha : *. \alpha$  **2,1 T-App**
4.  $M x \beta : \beta$  **3,\* T2-App**

■

### Problem

(3.8) Recall  $K \equiv \lambda x y . x \in \Lambda$  from untyped lambda calculus. Consider the following types

$$T_1 \equiv \Pi\alpha, \beta : * . \alpha \rightarrow \beta \rightarrow \alpha \quad T_2 \equiv \Pi\alpha : * . \alpha \rightarrow (\Pi\beta : * . \beta \rightarrow \alpha)$$

Find inhabitants of both type  $t_1 : T_1$  and  $t_2 : T_2$  under the empty context, which may be considered the closed  $\lambda 2$  form of  $K \in \Lambda_{T2}$ .

*Solution.*

$$\lambda\alpha, \beta : * . \lambda x : \alpha . \lambda y : \beta . x$$

$$\lambda\alpha : * . \lambda x : \alpha . \lambda\beta : * . \lambda y : \beta . x$$

*First Form.*

1.	$\alpha, \beta : *$	<b>Bound</b>
2.	$x : \alpha$	<b>Bound</b>
3.	$y : \beta$	<b>Bound</b>
4.	$x : \alpha$	<b>T-Var</b>
5.	$\lambda y : \beta . x : \beta \rightarrow \alpha$	<b>4 T-Abst</b>
6.	$\lambda x : \alpha . \lambda y : \beta . x : \alpha \rightarrow \beta \rightarrow \alpha$	<b>5 T-Abst</b>
7.	$\lambda\alpha, \beta : * . \lambda x : \alpha . \lambda y : \beta . x : \Pi\alpha, \beta : * . \alpha \rightarrow \beta \rightarrow \alpha$	<b>5 T2-Abst</b>

■

*Second Form.*

1.	$\alpha : *$	<b>Bound</b>
2.	$x : \alpha$	<b>Bound</b>
3.	$\beta : *$	<b>Bound</b>
4.	$y : \beta$	<b>Bound</b>
5.	$x : \alpha$	<b>T-Var</b>
6.	$\lambda y : \beta . x : \beta \rightarrow \alpha$	<b>5 T-Abst</b>
7.	$\lambda\beta : * . \lambda y : \beta . x : \Pi\beta : * . \beta \rightarrow \alpha$	<b>6 T2-Abst</b>
8.	$\lambda x : \alpha . \lambda\beta : * . \lambda y : \beta . x : \alpha \rightarrow (\Pi\beta : * . \beta \rightarrow \alpha)$	<b>7 T-Abst</b>
9.	$\lambda\alpha : * . \lambda x : \alpha . \lambda\beta : * . \lambda y : \beta . x : \Pi\alpha : * . \alpha \rightarrow (\Pi\beta : * . \beta \rightarrow \alpha)$	<b>8 T2-Abst</b>

■

### Problem

(3.9) Pretype the combinator

$$S \equiv \lambda x y z . x z (y z)$$

In closed form (typable in an empty context) in  $\Lambda_{\mathbb{T}2}$ .

*Solution.*

$$S \equiv \lambda \alpha, \beta, \gamma : * . \lambda x : \alpha \rightarrow \beta \rightarrow \gamma . \lambda y : \alpha \rightarrow \beta . \lambda z : \alpha . x z (y z)$$

*Proof.*

1.	$\alpha, \beta, \gamma : *$	<b>Bound</b>
2.	$x : \alpha \rightarrow \beta \rightarrow \gamma$	<b>Bound</b>
3.	$y : \alpha \rightarrow \beta$	<b>Bound</b>
4.	$z : \alpha$	<b>Bound</b>
5.	$x z : \beta \rightarrow \gamma$	<b>*, T-App</b>
6.	$y x : \beta$	<b>*, T-App</b>
7.	$x z (y x) : \gamma$	<b>5,6 T-App</b>
8.	$\lambda z : \alpha . x z (y x) : \alpha \rightarrow \gamma$	<b>7 T-Abst</b>
9.	$\lambda y : \alpha \rightarrow \beta . \lambda z : \alpha . x z (y x) : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$	<b>8 T-Abst</b>
10.	$\lambda x : \alpha \rightarrow \beta \rightarrow \gamma . \lambda y : \alpha \rightarrow \beta . \lambda z : \alpha . x z (y x) : (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$	<b>9 T-Abst</b>
11.		

$$\lambda \alpha, \beta, \gamma : * . \lambda x : \alpha \rightarrow \beta \rightarrow \gamma . \lambda y : \alpha \rightarrow \beta . \lambda z : \alpha . x z (y x)$$

$$: \Pi \alpha, \beta, \gamma : * . (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma \quad \mathbf{10 \ T2-Abst}$$

■

### Problem

(3.10 a) Consider the term

$$M \equiv \lambda x : \Pi \alpha : * . \alpha \rightarrow \alpha . x (\sigma \rightarrow \sigma)(x \sigma)$$

Prove that  $M$  is legal.

*Solution.* For a term to be legal there must exist a context so that the term could be typed. Here, a witness context is  $\Gamma \equiv \sigma : *$ .

*Proof.*

- |    |  |                  |
|----|--|------------------|
| 1. | $x : \Pi\alpha : * . \alpha \rightarrow \alpha$  | <b>Bound</b>     |
| 2. | $x (\sigma \rightarrow \sigma) : (\sigma \rightarrow \sigma) \rightarrow (\sigma \rightarrow \sigma)$<br>$x \sigma : \sigma \rightarrow \sigma$<br>$x (\sigma \rightarrow \sigma)(x \sigma) : \sigma \rightarrow \sigma$ | <b>*, T2-App</b> |
| 3. |  | <b>*, T2-App</b> |
| 4. |  | <b>2,3 T-App</b> |
| 5. |  |                  |
- $\lambda x : \Pi\alpha : * . \alpha \rightarrow \alpha. x (\sigma \rightarrow \sigma)(x \sigma)$   
 $: (\Pi\alpha : * . \alpha \rightarrow \alpha) \rightarrow \sigma \rightarrow \sigma$
- 4 T-Abst**

■

### Problem

(3.10 b) Find a term  $N$  such that  $M N$  is legal and may be considered to be a way to add type information to  $(\lambda x . x x)(\lambda y . y)$

*Solution.*

$$M \sigma N \equiv (\lambda x : \Pi\alpha : * . \alpha \rightarrow \alpha. x (\sigma \rightarrow \sigma)(x \sigma))\sigma(\lambda y : \sigma. y)$$

Is the same as  $(\lambda x . x x)(\lambda y . y)$  modulo type annotations.

*Proof.*

- |    |   |                   |
|----|---|-------------------|
| 1. | $M : (\Pi\alpha : * . \alpha \rightarrow \alpha) \rightarrow \sigma \rightarrow \sigma$ | <b>T-Var</b>      |
| 2. | $M \sigma : (\sigma \rightarrow \sigma) \rightarrow \sigma \rightarrow \sigma$          | <b>1*, T2-App</b> |
| 3. | $y : \sigma$  | <b>Bound</b>      |
| 4. | $y : \sigma$  | <b>T-Var</b>      |
| 5. | have $N := \lambda y : \sigma. y : \sigma \rightarrow \sigma$                           | <b>4 T-Abst</b>   |
| 6. | $M \sigma N : \sigma \rightarrow \sigma$  | <b>2,5 T-Abst</b> |

■

### Problem

(3.11) Recall  $\perp \equiv \Pi\alpha : * . \alpha$  from 3.5. Type and prove the following term legal:

$$\lambda x : \perp. x (\perp \rightarrow \perp \rightarrow \perp)(x (\perp \rightarrow \perp) x)(x (\perp \rightarrow \perp \rightarrow \perp) x x)$$

*Solution.*

*Proof.* The type  $\perp \rightarrow \perp$  is closed and well formed. Therefore, the term is legal.

1.	$\perp : * \equiv \Pi \alpha : * . \alpha$	<b>T-Form</b>
2.	$x : \perp$	<b>Bound</b>
3.	$x (\perp \rightarrow \perp \rightarrow \perp) : \perp \rightarrow \perp \rightarrow \perp$	<b>*, T2-App</b>
4.	$x (\perp \rightarrow \perp) : \perp \rightarrow \perp$	<b>*, T2-App</b>
5.	$x (\perp \rightarrow \perp) x : \perp$	<b>4, T-App</b>
6.	$x (\perp \rightarrow \perp \rightarrow \perp) (x (\perp \rightarrow \perp) x) : \perp \rightarrow \perp$	<b>3,5 T-App</b>
7.	$x (\perp \rightarrow \perp \rightarrow \perp) x : \perp \rightarrow \perp$	<b>3, T-App</b>
8.	$x (\perp \rightarrow \perp \rightarrow \perp) x x : \perp$	<b>7, T-App</b>
9.	$x (\perp \rightarrow \perp \rightarrow \perp) (x (\perp \rightarrow \perp) x) (x (\perp \rightarrow \perp \rightarrow \perp) x x) : \perp$	<b>6,8 T-App</b>
10.		
	$\lambda x : \perp. x (\perp \rightarrow \perp \rightarrow \perp) (x (\perp \rightarrow \perp) x)$	
	$(x (\perp \rightarrow \perp \rightarrow \perp) x x) : \perp \rightarrow \perp$	<b>9 T-Abst</b>

■

### Problem

(3.12) Given the Polymorphic Church Numerals:

$$\mathbf{nat} \in \mathbb{T}_2 := \Pi \alpha : * . (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$$

$$\bar{0} \equiv \lambda \alpha : * . \lambda f : \alpha \rightarrow \alpha . \lambda x : \alpha . x : \mathbf{nat}$$

$$\bar{1} \equiv \lambda \alpha : * . \lambda f : \alpha \rightarrow \alpha . \lambda x : \alpha . f x : \mathbf{nat}$$

$$\bar{2} \equiv \lambda \alpha : * . \lambda f : \alpha \rightarrow \alpha . \lambda x : \alpha . f (f x) : \mathbf{nat}$$

$$\mathbf{succ} \equiv \lambda n : \mathbf{nat} . \lambda \beta : * . \lambda f : \beta \rightarrow \beta . \lambda x : \beta . f (n \beta f x)$$

Prove that

$$\mathbf{succ} \bar{0} \underset{\beta}{=} \bar{1}$$

$$\mathbf{succ} \bar{1} \underset{\beta}{=} \bar{2}$$

*Solution.*

$$\begin{aligned}
& \text{succ } \bar{0} \\
& \equiv (\lambda n : \mathbf{nat}. \lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f (n \beta f x)) (\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. x) \\
& \xrightarrow[\beta]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f ((\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. x) \beta f x)) \\
& \xrightarrow[\beta_{T_2}]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f ((\lambda f : \beta \rightarrow \beta. \lambda x : \beta. x) f x)) \\
& \xrightarrow[\beta]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f x) \stackrel{\beta \rightarrow \alpha}{\alpha_{T_2}} \equiv \bar{1} \\
& \text{succ } \bar{1} \\
& \equiv (\lambda n : \mathbf{nat}. \lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f (n \beta f x)) (\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. f x) \\
& \xrightarrow[\beta]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f ((\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. f x) \beta f x)) \\
& \xrightarrow[\beta_{T_2}]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f ((\lambda f : \beta \rightarrow \beta. \lambda x : \beta. f x) f x)) \\
& \xrightarrow[\beta]{} (\lambda \beta : *. \lambda f : \beta \rightarrow \beta. \lambda x : \beta. f (f x)) \stackrel{\beta \rightarrow \alpha}{\alpha_{T_2}} \equiv \bar{2}
\end{aligned}$$

### Problem

(3.13 a) We define addition in Polymorphic Church Numerals as

$$\text{add} \equiv \lambda m, n : \mathbf{nat}. \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. m \alpha f (n \alpha f x)$$

Show that

$$\text{add } \bar{1} \ \bar{1} \stackrel{\beta}{=} \bar{2}$$

*Solution.*

$$\begin{aligned}
& \text{add } \bar{1} \ \bar{1} \\
& \equiv (\lambda m, n : \mathbf{nat}. \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. m \alpha f (n \alpha f x)) \bar{1} \ \bar{1} \\
& \xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. \bar{1} \alpha f (\bar{1} \alpha f x) \\
& \equiv \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. \bar{1} \alpha f ((\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. f x) \alpha f x) \\
& \xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. \bar{1} \alpha f (f x) \\
& \equiv \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. (\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. f x) \alpha f (f x) \\
& \xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \mathbf{nat}. f (f x) \stackrel{\alpha}{=} \bar{2}
\end{aligned}$$



## Problem

(3.13 b) Find a term `mul` simulates multiplication on `nat`.

*Solution.*

$$\text{mul} \equiv \lambda m, n : \mathbf{nat}. \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. m \alpha (n \alpha f) x$$

*Proof.* We derive the type first to prove a legal term.

1.	$m, n : \mathbf{nat}$	<b>Bound</b>
2.	$\alpha : *$	<b>Bound</b>
3.	$f : \alpha \rightarrow \alpha$	<b>Bound</b>
4.	$x : \alpha$	<b>Bound</b>
5.	$m \alpha : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$	<b>*, T2-App</b>
6.	$n \alpha : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$	<b>*, T2-App</b>
7.	$n \alpha f : \alpha \rightarrow \alpha$	<b>6, T-App</b>
8.	$m \alpha (n \alpha f) : \alpha \rightarrow \alpha$	<b>5,7 T-App</b>
9.	$m \alpha (n \alpha f) x : \alpha$	<b>8, T-App</b>
10.	$\lambda x : \alpha. m \alpha (n \alpha f) x : \alpha \rightarrow \alpha$	<b>9 T-Abst</b>
11.	$\lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. m \alpha (n \alpha f) x : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$	<b>10 T-Abst</b>
12.	$\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. m \alpha (n \alpha f) x : \mathbf{nat}$	<b>11 T2-Abst</b>
13.		

$$\lambda m, n : \mathbf{nat}. \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. m \alpha (n \alpha f) x$$

**12 T-Abst**

This proves that the term does indeed produce a natural number from two. Next let's prove that

$$\forall \bar{n}, \bar{m} : \mathbf{nat} \quad \text{mul } \bar{n} \ \bar{m} = \overline{n \times m}$$

It could be proven by induction that

$$\forall \bar{a} : \mathbf{nat} \quad \bar{a} \equiv \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. f^a x$$

$$\begin{aligned} \text{mul } \bar{n} \ \bar{m} &\equiv (\lambda m, n : \mathbf{nat}. \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. m \alpha (n \alpha f) x) (\bar{n}) (\bar{m}) \\ &\xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. \bar{n} \alpha (\bar{m} \alpha f) x \\ &\xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. \bar{n} \alpha (\lambda u : \alpha. f^m u) x \\ &\xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (\lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. f^n x) (\lambda u : \alpha. f^m u) x \\ &\xrightarrow[\beta]{} \lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. (f^m u)^n x \end{aligned}$$

By induction this can be further beta-reduced to

$$\xrightarrow[\beta]{\rightarrow} \lambda \alpha : * . \lambda f : \alpha \rightarrow \alpha . f^{mn} x \equiv \overline{m} \overline{n}$$

■

### Problem

(3.14) We present the Church-Encoded Boolean:

$$\mathbf{bool} \in \mathbb{T}_2 := \Pi \alpha : * . \alpha \rightarrow \alpha \rightarrow \alpha$$

$$\mathbf{true} \equiv \lambda \alpha : * . \lambda x, y : \alpha . x$$

$$\mathbf{false} \equiv \lambda \alpha : * . \lambda x, y : \alpha . y$$

Construct a  $\lambda 2$  term  $\mathbf{neg}$  such that  $\mathbf{neg} \mathbf{true} \equiv_{\beta} \mathbf{false}$  and  $\mathbf{neg} \mathbf{false} \equiv_{\beta} \mathbf{true}$ .

*Solution.*

$$\mathbf{neg} \equiv \lambda b : \mathbf{bool} . \lambda \alpha : * . b \alpha (\mathbf{false} \alpha) (\mathbf{true} \alpha)$$

*Neg True.*

$$\mathbf{neg} \mathbf{true} \equiv (\lambda b : \mathbf{bool} . \lambda \alpha : * . b \alpha (\mathbf{false} \alpha) (\mathbf{true} \alpha)) \mathbf{true}$$

$$\xrightarrow[\beta]{\rightarrow} \lambda \alpha : * . (\lambda x, y : \alpha . x) (\mathbf{false} \alpha) (\mathbf{true} \alpha)$$

$$\xrightarrow[\beta]{\rightarrow} \lambda \alpha : * . \mathbf{false} \alpha \xrightarrow[\eta]{\rightarrow} \mathbf{false}$$

■

*Neg False.*

$$\mathbf{neg} \mathbf{false} \equiv (\lambda b : \mathbf{bool} . \lambda \alpha : * . b \alpha (\mathbf{false} \alpha) (\mathbf{true} \alpha)) \mathbf{false}$$

$$\xrightarrow[\beta]{\rightarrow} \lambda \alpha : * . (\lambda x, y : \alpha . y) (\mathbf{false} \alpha) (\mathbf{true} \alpha)$$

$$\xrightarrow[\beta]{\rightarrow} \lambda \alpha : * . \mathbf{true} \alpha \xrightarrow[\eta]{\rightarrow} \mathbf{true}$$

■

### Problem

(3.15) Define

$$M \equiv \lambda u, v : \mathbf{bool} . \lambda \beta : * . \lambda x, y : \beta . u \beta (v \beta x y) (v \beta y y)$$

And reduce  $M \mathbf{true} \mathbf{true}$ ,  $M \mathbf{true} \mathbf{false}$ ,  $M \mathbf{false} \mathbf{true}$ ,  $M \mathbf{false} \mathbf{false}$ , and decide which logical operator is represented by  $M$ .

*Solution.*

$$\begin{aligned}
M \text{ true true} &\equiv (\lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta x y) (v \beta y y)) \text{ true true} \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. \text{true } \beta (\text{true } \beta x y) (\text{true } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. (\text{true } \beta x y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. x \equiv_{\alpha} \text{true}
\end{aligned}$$

$$\begin{aligned}
M \text{ true false} &\equiv (\lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta x y) (v \beta y y)) \text{ true false} \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. \text{true } \beta (\text{false } \beta x y) (\text{false } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. (\text{false } \beta x y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. y \equiv_{\alpha} \text{false}
\end{aligned}$$

$$\begin{aligned}
M \text{ false true} &\equiv (\lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta x y) (v \beta y y)) \text{ false true} \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. \text{false } \beta (\text{true } \beta x y) (\text{true } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. (\text{true } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. y \equiv_{\alpha} \text{true}
\end{aligned}$$

$$\begin{aligned}
M \text{ false false} &\equiv (\lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta x y) (v \beta y y)) \text{ false false} \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. \text{false } \beta (\text{false } \beta x y) (\text{false } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. (\text{false } \beta y y) \\
&\xrightarrow[\beta]{} \lambda \beta : *. \lambda x, y : \beta. y \equiv_{\alpha} \text{false}
\end{aligned}$$

Therefore  $M$  is equivalent to logical AND.

### Problem

(3.16) Find  $\lambda 2$  term representing the logical OR, XOR, IMP.

*Solution.*

$$\begin{aligned}
\text{OR} &\equiv \lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta x (v \beta x y) \\
\text{XOR} &\equiv \lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta y x) (v \beta x y) \\
\text{IMP} &\equiv \lambda u, v : \text{bool}. \lambda \beta : *. \lambda x, y : \beta. u \beta (v \beta x y) x
\end{aligned}$$

All of them could be checked by finite enumeration over  $\text{bool} \times \text{bool}$ .

### Problem

(3.17) Find  $\text{isZero} : \text{nat} \rightarrow \text{bool}$  such that  $\forall n : \text{nat}, \text{isZero } n \stackrel{\beta}{=} \text{false}$  except when  $n \equiv \bar{0}$ .

*Solution.*

$$\text{isZero} \equiv \lambda n : \text{nat}. n \text{ bool } (\lambda u : \text{bool}. \text{false}) \text{ true}$$

*Proof.*

$$\begin{aligned} \text{isZero } \bar{0} &\equiv (\lambda n : \text{nat}. n \text{ bool } (\lambda u : \text{bool}. \text{false}) \text{ true}) \bar{0} \\ &\xrightarrow[\beta]{} (\lambda \alpha : *. \lambda f : \alpha \rightarrow \alpha. \lambda x : \alpha. x) \text{ bool } (\lambda u : \text{bool}. \text{false}) \text{ true} \\ &\xrightarrow[\beta]{} (\lambda f : \text{bool} \rightarrow \text{bool}. \lambda x : \text{bool}. x) (\lambda u : \text{bool}. \text{false}) \text{ true} \\ &\xrightarrow[\beta]{} \text{true} \end{aligned}$$

■

By induction it could be proven that any other natural numbers must be applied  $\lambda u : \text{bool}. \text{false}$  to the body, making the result false, except for  $\bar{0}$ , where the function  $f : \alpha \rightarrow \alpha$  never got applied.

### Problem

(3.18 a) Define type

$$\text{tree} \equiv \Pi \alpha : *. (\text{bool} \rightarrow \alpha) \rightarrow (\text{bool} \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha) \rightarrow \alpha$$

Then we construct an inhabitant

$$\lambda \alpha. *. \lambda u : \text{bool} \rightarrow \alpha. \lambda v : \text{bool} \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha. M$$

Is a node of a binary tree. Sketch graphs of trees where  $M$  is

$$\begin{aligned} &u \text{ false} \\ &v \text{ true } (u \text{ false}) (u \text{ true}) \\ &v \text{ true } (u \text{ true}) (v \text{ false } (u \text{ true}) (u \text{ false})) \end{aligned}$$

*Solution.* A binary tree is usually defined as this:

```
inductive Tree (a : Type) where
| leaf (value : a) : Tree a
| node (left right : Tree a) : Tree a
```

With two constructors: a leaf or a node. Here  $\alpha$  is the type of the payload at each node. There are two constructors:  $u$  is the left constructor, taking a `bool` value for the direction of the node. The  $v$  term is the node constructor, taking a `bool` as the direction, two  $\alpha$ -typed terms as it's children.



### Problem

(3.18 b) Give a  $\lambda 2$  term, which, given input a polymorphic boolean  $p$  and two trees  $s$  and  $t$ , delivers the combined tree with  $p$  on top, left subtree  $s$  and right subtree  $t$ .

*Solution.*

$$\begin{aligned} \text{leaf} &:= \lambda p : \text{bool}. \lambda s, t : \text{tree}. \\ &\quad \lambda \alpha. \lambda u. \text{bool} \rightarrow \alpha. \lambda v : \text{bool} \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha. \\ &\quad v \ p \ (s \ \alpha \ u \ v) (t \ \alpha \ u \ v) \end{aligned}$$

*Proof.* We suppose

$$\begin{aligned} s &\equiv \lambda \beta : *. \lambda u_s : \text{bool} \rightarrow \beta. \lambda v_s : \text{bool} \rightarrow \beta \rightarrow \beta \rightarrow \beta. S \\ t &\equiv \lambda \gamma : *. \lambda u_t : \text{bool} \rightarrow \gamma. \lambda v_t : \text{bool} \rightarrow \gamma \rightarrow \gamma \rightarrow \gamma. T \end{aligned}$$

We want

$$\text{leaf } p \ s \ t \stackrel{\beta}{=} \lambda \alpha : *. \lambda u : \text{bool} \rightarrow \alpha. \lambda v : \text{bool} \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha. v \ p \ S \ T$$

For compactness we denote

$$\begin{aligned} \alpha : * \vdash \tau_{\text{mkleaf}} &\equiv \text{bool} \rightarrow \alpha \\ \alpha : * \vdash \tau_{\text{mknode}} &\equiv \text{bool} \rightarrow \alpha \rightarrow \alpha \rightarrow \alpha \end{aligned}$$

By beta reduction we have

$$\begin{aligned}
\text{leaf } p \ s \ t &\equiv (\lambda p : \text{bool} . \lambda s, t : \text{tree} \dots) \ p \ s \ t \\
&\xrightarrow[\beta]{} (\lambda \alpha : * . \lambda u . \tau_{\text{mkleaf}} . \lambda v : \tau_{\text{mknode}} . v \ p \ (s \ \alpha \ u \ v) (t \ \alpha \ u \ v)) \\
&\xrightarrow[\beta]{} (\lambda \alpha : * . \lambda u . \tau_{\text{mkleaf}} . \lambda v : \tau_{\text{mknode}} . v \ p \ (s \ \alpha \ u \ v) (t \ \alpha \ u \ v)) \\
&\xrightarrow[\beta]{} \lambda \alpha : * . \lambda u . \tau_{\text{mkleaf}} . \lambda v : \tau_{\text{mknode}} . v \ p \ S \ T
\end{aligned}$$

■

### Problem

(3.19) If  $\Gamma \vdash L : \sigma$ , then  $\Gamma$  is a valid  $\lambda_2$  context.

*Solution.* The definition of “valid” here would be taken as relative to a judgement as the fact to be able to derive the judgement. Thus, it meant a complete inference path could be made using only statements and judgements derived from the context. Now proof by induction on inference rule that deduced  $L : \sigma$ .

*Case 1 : T-Var.* The premise is that  $\Gamma$  is a  $\lambda_2$  context. ■

*Case 2 : T-App.* Therefore  $L \equiv M \ N$  for some  $M, N \in \Lambda_{\mathbb{T}_2}$ . Therefore,

$$\Gamma \vdash M : \tau \rightarrow \sigma \quad \Gamma \vdash N : \tau$$

for some  $\tau \in \mathbb{T}_2$ . By the inductive hypothesis on any premise  $\Gamma$  is a valid  $\lambda_2$  context. ■

*Case 3 : T-Abst.* Therefore  $L \equiv \lambda x : \alpha . N$  such that

$$\Gamma, x : \alpha \vdash N : \beta$$

for some  $\alpha, \beta \in \mathbb{T}_2$  such that  $\sigma \equiv \alpha \rightarrow \beta$ . By the inductive hypothesis,  $\Gamma, x : \alpha$ . By the recursive definition of  $\lambda_2$  contexts, for some valid context  $\Delta$ ,  $\forall n \in \text{dom } \Delta$ ,  $n$  could not depend on statement declared after  $n$  in the context. Therefore, no statement in  $\Gamma$  could depend on  $x : \alpha$ . Therefore,  $\Gamma$  is a valid context. ■

*Case 4 : T-Form.* The premise is that  $\Gamma$  is a valid  $\lambda_2$  context. ■

*Case 5 : T2-App.* Therefore  $L \equiv N \ B$  for some  $N, B \in \mathbb{V}_2$  such that

$$\Gamma \vdash N : \Pi \alpha : * . \sigma \quad \Gamma \vdash B : *$$

By the inductive hypothesis on the any premise  $\Gamma$  is a valid  $\lambda_2$  context. ■

*Case 6 : T2-Abst.* Therefore  $L \equiv \lambda \alpha : * . M$  for some  $M \in \Lambda_{\mathbb{T}_2}$  such that

$$\Gamma, \alpha : * \vdash M : \beta$$

and  $\sigma \equiv \Pi\alpha : * . \beta$ . By reasoning in *Case 3* no statement in the context could depend on any statement before the latter's declaration. Therefore no statement in  $\Gamma$  could depend on  $\alpha : *$ , making it a valid context. ■

### Problem

(3.20) Prove the free variable lemma for  $\lambda 2$ .

$$\Gamma \vdash L : \sigma \Rightarrow \text{FV } L \subseteq \text{dom } \Gamma$$

*Solution.* Proof by induction on inference rules that deduced  $L : \sigma$ . The only rule not considered is T-Form since all terms apparent in  $\mathbb{T}_2$ .

*Case 1 : T-Var.* Therefore  $L$  is the only free variable in  $L$ . By the generation lemma  $L : \sigma \in \Gamma$ , so  $\{L\} \subseteq \text{dom } \Gamma$  ■

*Case 2 : T-App.* Therefore by the generation lemma  $L \equiv M N$  for some  $M, N \in \Lambda_{\mathbb{T}_2}$  such that

$$\Gamma \vdash M : \tau \rightarrow \sigma \quad \Gamma \vdash N : \tau$$

For some type  $\tau$ . By the inductive hypothesis  $\text{FV } M \subseteq \Gamma$  and  $\text{FV } N \subseteq \text{dom } \Gamma$ . Therefore  $\text{FV } L = (\text{FV } M) \cup (\text{FV } N) \subseteq \text{dom } \Gamma$ . ■

*Case 3 : T-Abst.* Therefore by the generation lemma  $L \equiv \lambda x : \alpha. M$  for some  $M \in \Lambda_{\mathbb{T}_2}$  such that

$$\Gamma, x : \alpha \vdash M : \beta$$

and  $\sigma \equiv \alpha \rightarrow \beta$ . By the inductive hypothesis  $\text{FV } M \subseteq \text{dom } \Gamma \cup \{x\}$ . Therefore

$$\text{FV } L = \text{FV } M \setminus \{x\} \subseteq (\text{dom } \Gamma \cup \{x\}) \setminus \{x\} = \text{dom } \Gamma$$

*Case 4 : T2-App.* Therefore  $L \equiv B$  for some  $N, B \in \mathbb{V}_2$  such that

$$\Gamma \vdash N : \Pi\alpha : * . \sigma \quad \Gamma \vdash B : *$$

By the inductive hypothesis  $\text{FV } N \subseteq \text{dom } \Gamma$ . Since  $B \in \mathbb{T}_2$  then  $\text{FV } B = \emptyset$ . Therefore  $\text{FV } L = \text{FV } N \subseteq \text{dom } \Gamma$ . ■

*Case 5 : T2-Abst.* Therefore  $L \equiv \lambda\alpha : * . M$  for some  $M \in \Lambda_{\mathbb{T}_2}$  such that

$$\Gamma, \alpha : * \vdash M : \beta$$

and  $\sigma \equiv \Pi\alpha : * . \beta$ . Because  $\alpha \in \mathbb{T}_2$  so  $\alpha \notin \text{FV } M$  and  $\text{FV } L = \text{FV } M$ , thus  $\text{FV } M = \text{FV } L \subseteq \Gamma$ . ■

### Problem

(3.21) Give a recursive definition for  $\text{FTV} : \mathbb{T}_2 \cup \Lambda_{\mathbb{T}_2} \rightarrow \mathbb{V}_2$

*Solution.* Here  $\alpha \in \mathbb{V}_2$ ,  $A, B \in \mathbb{T}_2$ ,  $x \in V$  and  $M \in \Lambda_{\mathbb{T}_2}$ .

Form	Value
$\text{FTV } \alpha$	$\{\alpha\}$
$\text{FTV } x$	$\emptyset$
$\text{FTV } (A \rightarrow B)$	$\text{FTV}(A) \cup \text{FTV}(B)$
$\text{FTV } (\Pi \alpha : * . A)$	$\text{FTV}(A) \setminus \{\alpha\}$
$\text{FTV } (M N)$	$\text{FTV}(M) \cup \text{FTV}(N)$
$\text{FTV } (\lambda x : A . M)$	$\text{FTV}(A) \cup \text{FTV}(M)$
$\text{FTV } (M B)$	$\text{FV}(M) \cup \text{FTV}(B)$
$\text{FTV } (\lambda \alpha : * . M)$	$\text{FTV}(M) \setminus \{\alpha\}$

—

Completed Dec 16 10:11 pm.