EXCERCISES

CHAPTER 3

SEAN LI [1]

Definition Some rules for reference.

$$\frac{x : \sigma \in \Gamma \quad \Gamma \text{ is a } \lambda 2 \text{ context}}{\Gamma \vdash x : \sigma} \text{ (T-Var)} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\,N : \tau} \text{ (T-App)}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma.\,M : \sigma \to \tau} \text{ (T-Abst)} \qquad \frac{\alpha \in \mathbb{T}_2 \quad \text{FV } \alpha \subseteq \text{dom } \Gamma}{\alpha : * \in \Gamma} \text{ (T-Form)}$$

$$\frac{\Gamma \vdash M : \Pi_{\alpha:*}.\,A \quad \Gamma \vdash B : *}{\Gamma \vdash M\,B : A\,[\alpha := B]} \text{ (T2-App)} \qquad \frac{\Gamma, \alpha : * \vdash M : A}{\Gamma \vdash \lambda \alpha : *.\,M : \Pi_{\alpha:*} : A} \text{ (T2-Abst)}$$

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

$$(1)\ \Gamma \vdash \alpha : * \qquad (2)\ \Gamma \vdash \beta : *$$
$$(3)\ \Gamma \vdash f : \alpha \to \beta \quad (4)\ \Gamma \vdash x : \alpha$$

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

$$1 - 2 - 3 - 4 \quad 1 - 2 - 4 - 3$$
$$2 - 1 - 3 - 4 \quad 2 - 1 - 4 - 3$$

Problem

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

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

Solution.

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

16.

$\lambda\gamma : *.\lambda f : \alpha \to \beta.\lambda g : \beta \to \gamma.\lambda x : \alpha.\, g\,(f\,x)$

$\qquad : \Pi\gamma : *.(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$ **15 T2-Abst**

17.

$\lambda\beta,\gamma : *.\lambda f : \alpha \to \beta.\lambda g : \beta \to \gamma.\lambda x : \alpha.\, g\,(f\,x)$

$\qquad : \Pi\beta,\gamma : *.(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$ **16 T2-Abst**

18.

$\lambda\alpha,\beta,\gamma : *.\lambda f : \alpha \to \beta.\lambda g : \beta \to \gamma.\lambda x : \alpha.\, g\,(f\,x)$

$\quad : \Pi\alpha,\beta,\gamma : *.(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$ **17 T2-Abst**

Problem

(3.3 a) Given M in 3.2, and a context Γ such that

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

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

$$\Gamma \vdash \mathrm{succ} : \mathtt{nat} \to \mathtt{nat}$$

$$\Gamma \vdash \mathrm{even} : \mathtt{nat} \to \mathtt{bool}$$

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

Solution. Proof by deriving the term's type.

Proof.

1. $M : \Pi\alpha,\beta,\gamma : *.(\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma$ **T-Var**

2. $\mathtt{nat}, \mathtt{bool} : *$ **T-Form**

3. $M\ \mathtt{nat} : \Pi\beta,\gamma : *.(\mathtt{nat} \to \beta) \to (\beta \to \gamma) \to \mathtt{nat} \to \gamma$ **2,1 T2-App**

4. $M\ \mathtt{nat\ nat} : \Pi\gamma : *.(\mathtt{nat} \to \mathtt{nat}) \to (\mathtt{nat} \to \gamma) \to \mathtt{nat} \to \gamma$ **2,3 T2-App**

5.

$M\ \mathtt{nat\ nat\ bool} : (\mathtt{nat} \to \mathtt{nat}) \to (\mathtt{nat} \to \mathtt{bool}) \to \mathtt{nat} \to \mathtt{bool}$ **2,3 T2-App**

6. $\mathrm{succ} : \mathtt{nat} \to \mathtt{nat}, \mathrm{even} : \mathtt{nat} \to \mathtt{bool}$ **T-Var**

7. $M\ \mathtt{nat\ nat\ bool}$ succ $: (\mathtt{nat} \to \mathtt{bool}) \to \mathtt{nat} \to \mathtt{bool}$ **6,5 T-App**

8. $M\ \mathtt{nat\ nat\ bool}$ succ even $: \mathtt{nat} \to \mathtt{bool}$ **6,7 T-App**

■

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

Proof.

$$M\ \texttt{nat}\ \texttt{nat}\ \texttt{bool}\ \text{succ}\ \text{even}$$

$$\equiv (\lambda \alpha, \beta, \gamma, f, g\ .\lambda x : \alpha.\ g\ (f\ x))\ \texttt{nat}\ \texttt{nat}\ \texttt{bool}\ \text{succ}\ \text{even}$$

$$\underset{\beta}{\twoheadrightarrow} (\lambda f : \texttt{nat} \to \texttt{nat}.\lambda g : \texttt{nat} \to \texttt{bool}.\lambda x : \texttt{nat}.\ g\ (f\ (x)))\ \text{succ}\ \text{even}$$

$$\underset{\beta}{\to} (\lambda x : \texttt{nat}.\ \text{even}\ (\text{succ}\ x))$$

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

Solution.

Proof.

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

∎

Solution.

Proof.

1.　$\alpha, \beta : *$　　　　　　　　　　　　　　　　　　**Bound**

2.　$f : \alpha \to \alpha$　　　　　　　　　　　　　　　　　**Bound**

3.　$g : \alpha \to \beta$　　　　　　　　　　　　　　　　　**Bound**

4.　$x : \alpha$　　　　　　　　　　　　　　　　　　**Bound**

5.　$f\,x : \alpha$　　　　　　　　　　　　　　　　　　***,* T-App**

6.　$f\,(f\,x) : \alpha$　　　　　　　　　　　　　　　　　***,5 T-App**

7.　$g\,(f\,(f\,x)) : \beta$　　　　　　　　　　　　　　　***,6 T-App**

8.　$\lambda x : \alpha.\,g\,(f\,(f\,x)) : \alpha \to \beta$　　　　　　　　**7 T-Abst**

9.　$\lambda g : \alpha \to \beta.\lambda x : \alpha.\,g\,(f\,(f\,x)) : (\alpha \to \beta) \to \alpha \to \beta$　　**8 T-Abst**

$\lambda f : \alpha \to \alpha.\lambda g : \alpha \to \beta.\lambda x : \alpha.\,g\,(f\,(f\,x))$

10.　　$: (\alpha \to \alpha) \to (\alpha \to \beta) \to \alpha \to \beta$　　　**9 T-Abst**

11.

$\lambda \alpha, \beta : *.\lambda f : \alpha \to \alpha.\lambda g : \alpha \to \beta.\lambda x : \alpha.\,g\,(f\,(f\,x))$

　　$: \Pi \alpha, \beta : *.(\alpha \to \alpha) \to (\alpha \to \beta) \to \alpha \to \beta$　　　**10 T2-Abst**

12.

$(\lambda \alpha, \beta : *.\lambda f : \alpha \to \alpha.\lambda g : \alpha \to \beta.\lambda x : \alpha.\,g\,(f\,(f\,x)))\ \mathtt{nat}$

　　$: \Pi \beta : *.(\mathtt{nat} \to \mathtt{nat}) \to (\mathtt{nat} \to \beta) \to \mathtt{nat} \to \beta$　　***,11 T2-App**

13.

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

　　$: \Pi \beta : *.(\mathtt{nat} \to \mathtt{nat}) \to (\mathtt{nat} \to \mathtt{bool}) \to \mathtt{nat} \to \mathtt{bool}$　　***,12 T2-App**

∎

Problem

(3.5 a) Let $\bot \equiv \Pi \alpha : *.\alpha$. Prove $\bot$ 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.　$\alpha : *$　　　　　　**T-Form**

3. $\Pi\alpha : * . \alpha : * \rightarrow *$ **T2-Abst**

∎

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

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**

∎

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

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

Proof.

1. $y : \beta$ **Bound**

2. $\quad\big|\;y : \beta$ **T-Var**

3. $\lambda y : \beta.\,y : \beta \rightarrow \beta$ **2 T-Abst**

∎

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

Proof.

1. $y : \beta$ **Bound**

2. $\quad\big|\;x : \Pi\alpha : * . \alpha$ **T-Var**

3. $\quad\big|\;\beta : *$ **T-Form**

4. $\quad\big|\;x\,\beta : \beta$ **2,3 T2-App**

5. $\lambda y : \beta.\, x\, \beta : \beta \to \beta$ **4 T-Abst**

∎

3. $\lambda y : \beta.\, x\, (\beta \to \beta)\, y.$

Proof.

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

∎

Problem

(3.5 d) Prove that the following terms inhabit the same type in Γ:

$$\lambda f : \beta \to \beta \to \beta.\, f\, (x\, \beta)(x\, \beta)$$

$$x\, ((\beta \to \beta \to \beta) \to \beta)$$

Solution. We simply derive the types.

First Term.

1. $f : \beta \to \beta \to \beta$ **Bound**
2. $f : \beta \to \beta \to \beta$ **T-Var**
3. $x : \Pi\alpha : *\,.\alpha$ **T-Var**
4. $\beta : *$ **T-Form**
5. $x\, \beta : \beta$ **3,4 T2-App**
6. $f\, (x\, \beta) : \beta \to \beta$ **2,5 T-App**
7. $f\, (x\, \beta)(x\, \beta) : \beta$ **6,5 T-App**
8. $\lambda f : \beta \to \beta \to \beta.\, f\, (x\, \beta)(x\, \beta) : (\beta \to \beta \to \beta) \to \beta$ **6 T-Abst**

∎

Second Term.

1. $(\beta \to \beta \to \beta) \to \beta : *$ **T-Form**

2. $x : \Pi\alpha : * .\alpha$ **T-Var**

3. $x \, ((\beta \to \beta \to \beta) \to \beta) : (\beta \to \beta \to \beta) \to \beta$ **2,1 T2-App**

∎

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

<div style="background: teal;">

Problem

(3.6 a) Find inhabitant of type

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

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

</div>

Solution.

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

Proof.

1. $\alpha, \beta : *$ **Bound**

2. $\text{nat} \to \alpha : *$ **T-Form**

3. $x : \text{nat} \to \alpha$ **Bound**

4. $\alpha \to \text{nat} \to \beta : *$ **T-Form**

5. $y : \alpha \to \text{nat} \to \beta$ **Bound**

6. $\text{nat} : *$ **Bound**

7. $z : \text{nat}$ **Bound**

8. $y : \alpha \to \text{nat} \to \beta$ **T-Var**

9. $x : \text{nat} \to \alpha$ **T-Var**

10. $z : \text{nat}$ **T-Var**

11. $x \, z : \alpha$ **9,10 T-App**

12. $y \, (x \, z) : \text{nat} \to \beta$ **8,11 T-App**

13. $y \, (x \, z) \, z : \beta$ **12,10 T-App**

14. $\lambda z : \text{nat.} \, y \, (x \, z) \, z : \text{nat} \to \beta$ **13 T-Abst**

15. $\lambda y : \alpha \to \text{nat} \to \beta.\lambda z : \text{nat.} \, y \, (x \, z) \, z$
$: (\alpha \to \text{nat} \to \beta) \to \text{nat} \to \beta$ **14 T-Abst**

16. $\lambda x : \text{nat} \to \alpha. \, y : \alpha \to \text{nat} \to \beta.\lambda z : \text{nat.} \, y \, (x \, z) \, z$
$: (\text{nat} \to \alpha) \to (\alpha \to \text{nat} \to \beta) \to \text{nat} \to \beta$ **15 T2-Abst**

17.

$$\lambda\alpha, \beta : * . \, x : \mathtt{nat} \to \alpha. \, y : \alpha \to \mathtt{nat} \to \beta. \lambda z : \mathtt{nat}. \, y \, (x \, z) \, z$$
$$: \Pi\alpha, \beta : * .(\mathtt{nat} \to \alpha) \to (\alpha \to \mathtt{nat} \to \beta) \to \mathtt{nat} \to \beta \qquad \textbf{16 T2-Abst}$$

■

Problem

(3.6 b) Find inhabitant of type

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

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

Solution.

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

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

Proof.

1.	$\delta : *$	**Bound**
2.	$x : (\alpha \to \gamma) \to \delta$	**Bound**
3.	$y : \alpha \to \beta$	**Bound**
4.	$z : \beta \to \gamma$	**Bound**
5.	$u : \alpha$	**Bound**
6.	$y \, u : \beta$	*,* **T-App**
7.	$z \, (y \, u) : \gamma$	*,6 **T-App**
8.	$\lambda u : \alpha. \, z \, (y \, u) : \alpha \to \gamma$	**7 T-Abst**
9.	$x \, (\lambda u : \alpha. \, z \, (y \, u)) : \delta$	**8 T-Abst**
10.	$\lambda z : \beta \to \gamma. \, x \, (\lambda u : \alpha. \, z \, (y \, u)) : (\beta \to \gamma) \to \delta$	**9 T-Abst**
	$\lambda y : \alpha \to \beta.\lambda z : \beta \to \gamma. \, x \, (\lambda u : \alpha. \, z \, (y \, u))$	
11.	$: (\alpha \to \beta) \to (\beta \to \gamma) \to \delta$	**10 T-Abst**
	$\lambda x : (\alpha \to \gamma) \to \delta.\lambda y : \alpha \to \beta.\lambda z : \beta \to \gamma. \, x \, (\lambda u : \alpha. \, z \, (y \, u))$	
12.	$: ((\alpha \to \gamma) \to \beta) \to (\alpha \to \beta) \to (\beta \to \gamma) \to \delta$	**11 T-Abst**
13.		

$$\lambda\delta : * .\lambda x : (\alpha \to \gamma) \to \delta.\lambda y : \alpha \to \beta.\lambda z : \beta \to \gamma. \, x \, (\lambda u : \alpha. \, z \, (y \, u))$$
$$: \Pi\delta : * .((\alpha \to \gamma) \to \delta) \to (\alpha \to \beta) \to (\beta \to \gamma) \to \delta \qquad \textbf{12 T2-Abst}$$

■

Problem

(3.6 c) Find inhabitant of type

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

In the empty context

Solution.

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

Proof.

1.	α, β, γ	**Bound**
2.	$f : \alpha \to (\beta \to \alpha) \to \gamma$	**Bound**
3.	$x : \alpha$	**Bound**
4.	$f\ x : (\beta \to \alpha) \to \gamma$	***,* T-App**
5.	$u : \beta$	**Bound**
6.	$x : \alpha$	**T-Var**
7.	$\lambda u : \beta.\ x : \beta \to \alpha$	**6 T-Abst**
8.	$f\ x\ (\lambda u : \beta.\ x) : \gamma$	**4,7 T-App**
9.	$\lambda x : \alpha.\ f\ x\ (\lambda u : \beta.\ x) : \alpha \to \gamma$	**8 T-Abst**
10.	$\lambda f : \alpha \to (\beta \to \alpha) \to \gamma.\lambda x : \alpha.\ f\ x\ (\lambda u : \beta.\ x)$ $: (\alpha \to (\beta \to \alpha) \to \gamma) \to \alpha \to \gamma$	**9 T-Abst**
11.		

$$\lambda\alpha, \beta, \gamma : * .\lambda f : \alpha \to (\beta \to \alpha) \to \gamma.\lambda x : \alpha.\ f\ x\ (\lambda u : \beta.\ x)$$
$$: \Pi\alpha, \beta, \gamma : * .(\alpha \to (\beta \to \alpha) \to \gamma) \to \alpha \to \gamma \qquad \textbf{10 T2-Abst}$$

■

Problem

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

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

Typing M.

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

∎

Typing $x\ M\ \beta$.

1. $M : \alpha$ **T-Var**
2. $x : \alpha \to \Pi\alpha : *.\alpha$ **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

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

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 $K \in \Lambda_{\mathbb{T}2}$.

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. $\alpha, \beta : *$ **Bound**
2. $x : \alpha$ **Bound**
3. $y : \beta$ **Bound**
4. $x : \alpha$ **T-Var**
5. $\lambda y : \beta.\, x : \beta \to \alpha$ **4 T-Abst**
6. $\lambda x : \alpha.\lambda y : \beta.\, x : \alpha \to \beta \to \alpha$ **5 T-Abst**

7. $\lambda\alpha, \beta : * .\lambda x : \alpha.\lambda y : \beta.\, x : \Pi\alpha, \beta : * .\alpha \to \beta \to \alpha$ **5 T2-Abst**

■

Second Form.

1. $\alpha : *$ **Bound**
2. $\quad x : \alpha$ **Bound**
3. $\qquad \beta : *$ **Bound**
4. $\qquad\quad y : \beta$ **Bound**
5. $\qquad\qquad x : \alpha$ **T-Var**
6. $\qquad\quad \lambda y : \beta.\, x : \beta \to \alpha$ **5 T-Abst**
7. $\qquad \lambda\beta : * .\lambda y : \beta.\, x : \Pi\beta : * .\beta \to \alpha$ **6 T2-Abst**
8. $\quad \lambda x : \alpha.\lambda\beta : * .\lambda y : \beta.\, x : \alpha \to (\Pi\beta : * .\beta \to \alpha)$ **7 T-Abst**
9. $\lambda\alpha : * .\lambda x : \alpha.\lambda\beta : * .\lambda y : \beta.\, x : \Pi\alpha : * .\alpha \to (\Pi\beta : * .\beta \to \alpha)$ **8 T2-Abst**

■

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.

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

Proof.

1. $\alpha, \beta, \gamma : *$ **Bound**
2. $\quad x : \alpha \to \beta \to \gamma$ **Bound**
3. $\qquad y : \alpha \to \beta$ **Bound**
4. $\qquad\quad z : \alpha$ **Bound**
5. $\qquad\qquad x\, z : \beta \to \gamma$ ***,* T-App**
6. $\qquad\qquad y\, x : \beta$ ***,* T-App**
7. $\qquad\qquad x\, z\, (y\, x) : \gamma$ **5,6 T-App**
8. $\qquad\quad \lambda z : \alpha.\, x\, z\, (y\, x) : \alpha \to \gamma$ **7 T-Abst**
9. $\qquad \lambda y : \alpha \to \beta.\lambda z : \alpha.\, x\, z\, (y\, x) : (\alpha \to \beta) \to \alpha \to \gamma$ **8 T-Abst**

10.
$$\lambda x : \alpha \to \beta \to \gamma.\lambda y : \alpha \to \beta.\lambda z : \alpha.\, x\, z\, (y\, x)$$
$$: (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$$
9 T-Abst

11.

$$\lambda \alpha, \beta, \gamma : *\,.\lambda x : \alpha \to \beta \to \gamma.\lambda y : \alpha \to \beta.\lambda z : \alpha.\, x\, z\, (y\, x)$$
$$: \Pi \alpha, \beta, \gamma : *\,.(\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$$
10 T2-Abst

■

Problem

(3.10 a) Consider the term

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

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. $x : \Pi \alpha : *\,.\alpha \to \alpha$ **Bound**
2. $x\,(\sigma \to \sigma) : (\sigma \to \sigma) \to (\sigma \to \sigma)$ ***,* T2-App**
3. $x\,\sigma : \sigma \to \sigma$ ***,* T2-App**
4. $x\,(\sigma \to \sigma)(x\,\sigma) : \sigma \to \sigma$ **2,3 T-App**
5.

$$\lambda x : \Pi \alpha : *\,.\alpha \to \alpha.\, x\,(\sigma \to \sigma)(x\,\sigma)$$
$$: (\Pi \alpha : *\,.\alpha \to \alpha) \to \sigma \to \sigma$$
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.

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

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

Proof.

1. $M : (\Pi\alpha : * \,.\alpha \to \alpha) \to \sigma \to \sigma$ **T-Var**
2. $M\,\sigma : (\sigma \to \sigma) \to \sigma \to \sigma$ **1,* T2-App**
3. $y : \sigma$ **Bound**
4. $\big|\; y : \sigma$ **T-Var**
5.   have $N := \lambda y : \sigma.\, y : \sigma \to \sigma$ **4 T-Abst**
6. $M\,\sigma\,N : \sigma \to \sigma$ **2,5 T-Abst**

∎

Problem

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

$$\lambda x : \bot .\, x \,(\bot \to \bot \to \bot)(x\,(\bot \to \bot)\,x)(x\,(\bot \to \bot \to \bot)\,x\,x)$$

Solution.

Proof. The type $\bot \to \bot$ is closed and well formed. Therefore, the term is legal.

1. $\bot : * \equiv \Pi\alpha : * \,.\alpha$ **T-Form**
2. $x : \bot$ **Bound**
3. $\big|\; x\,(\bot \to \bot \to \bot) : \bot \to \bot \to \bot$ ***,* T2-App**
4. $x\,(\bot \to \bot) : \bot \to \bot$ ***,* T2-App**
5. $x\,(\bot \to \bot)\,x : \bot$ **4,* T-App**
6. $x\,(\bot \to \bot \to \bot)(x\,(\bot \to \bot)\,x) : \bot \to \bot$ **3,5 T-App**
7. $x\,(\bot \to \bot \to \bot)\,x : \bot \to \bot$ **3,* T-App**
8. $x\,(\bot \to \bot \to \bot)\,x\,x : \bot$ **7,* T-App**
9. $x\,(\bot \to \bot \to \bot)(x\,(\bot \to \bot)\,x)(x\,(\bot \to \bot \to \bot)\,x\,x) : \bot$ **6,8 T-App**
10.

$\lambda x : \bot .\, x\,(\bot \to \bot \to \bot)(x\,(\bot \to \bot)\,x)$
    $(x\,(\bot \to \bot \to \bot)\,x\,x) : \bot \to \bot$ **9 T-Abst**

∎

(3.12) Given the Polymorphic Church Numerals:

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

$$\overline{0} \equiv \lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, x : \mathtt{nat}$$

$$\overline{1} \equiv \lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, f \, x : \mathtt{nat}$$

$$\overline{2} \equiv \lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, f \, (f \, x) : \mathtt{nat}$$

$$\mathrm{succ} \equiv \lambda n : \mathtt{nat} . \lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, (n \, \beta \, f \, x)$$

Prove that

$$\mathrm{succ} \ \overline{0} \underset{\beta}{=} \overline{1}$$

$$\mathrm{succ} \ \overline{1} \underset{\beta}{=} \overline{2}$$

Solution.

$$\mathrm{succ} \ \overline{0}$$

$$\equiv (\lambda n : \mathtt{nat} . \lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, (n \, \beta \, f \, x))(\lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, x)$$

$$\underset{\beta}{\to} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, ((\lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, x)\beta \, f \, x))$$

$$\underset{\beta_{\mathrm{T2}}}{\to} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, ((\lambda f : \beta \to \beta . \lambda x : \beta . \, x) \, f \, x))$$

$$\underset{\beta}{\twoheadrightarrow} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, x) \overset{\beta \to \alpha}{\underset{\alpha_{\mathrm{T2}}}{\equiv}} \overline{1}$$

$$\mathrm{succ} \ \overline{1}$$

$$\equiv (\lambda n : \mathtt{nat} . \lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, (n \, \beta \, f \, x))(\lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, f \, x)$$

$$\underset{\beta}{\to} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, ((\lambda\alpha : * . \lambda f : \alpha \to \alpha . \lambda x : \alpha . \, f \, x)\beta \, f \, x))$$

$$\underset{\beta_{\mathrm{T2}}}{\to} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, ((\lambda f : \beta \to \beta . \lambda x : \beta . \, f \, x) \, f \, x))$$

$$\underset{\beta}{\twoheadrightarrow} (\lambda\beta : * . \lambda f : \beta \to \beta . \lambda x : \beta . \, f \, (f \, x)) \overset{\beta \to \alpha}{\underset{\alpha_{\mathrm{T2}}}{\equiv}} \overline{2}$$

Problem

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

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

Show that

$$\text{add}\ \overline{1}\ \ \overline{1} \underset{\beta}{=} \overline{2}$$

Solution.

$$\text{add}\ \overline{1}\ \ \overline{1}$$

$$\equiv (\lambda m, n : \texttt{nat}.\lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\ m\ \alpha\ f\ (n\ \alpha\ f\ x))\overline{1}\ \ \overline{1}$$

$$\underset{\beta}{\twoheadrightarrow} \lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\overline{1}\alpha\ f\ (\overline{1}\alpha\ f\ x)$$

$$\equiv \lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\overline{1}\alpha\ f\ ((\lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\ f\ x)\alpha\ f\ x)$$

$$\underset{\beta}{\twoheadrightarrow} \lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\overline{1}\alpha\ f\ (f\ x)$$

$$\equiv \lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.(\lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\ f\ x)\alpha\ f\ (f\ x)$$

$$\underset{\beta}{\twoheadrightarrow} \lambda\alpha : *.\lambda f : \alpha \rightarrow \alpha.\lambda x : \alpha.\ f\ (f\ x) \underset{\alpha}{\equiv} \overline{2}$$

Problem

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

Solution.

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

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

1. $m, n : \texttt{nat}$		**Bound**
2. $\alpha : *$		**Bound**
3. $f : \alpha \rightarrow \alpha$		**Bound**
4. $x : \alpha$		**Bound**
5. $m\ \alpha : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$		*,* **T2-App**
6. $n\ \alpha : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$		*,* **T2-App**
7. $n\ \alpha\ f : \alpha \rightarrow \alpha$		6,* **T-App**
8. $m\ \alpha(n\ \alpha\ f) : \alpha \rightarrow \alpha$		5,7 **T-App**

9.	$m\,\alpha(n\,\alpha\,f)\,x : \alpha$	**8,* T-App**
10.	$\lambda x : \alpha.\,m\,\alpha(n\,\alpha\,f)\,x : \alpha \to \alpha$	**9 T-Abst**
11.	$\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,m\,\alpha(n\,\alpha\,f)\,x : (\alpha \to \alpha) \to \alpha \to \alpha$	**10 T-Abst**
12.	$\lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,m\,\alpha(n\,\alpha\,f)\,x : \texttt{nat}$	**11 T2-Abst**
13.		

$$\lambda m, n : \texttt{nat}.\lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,m\,\alpha(n\,\alpha\,f)\,x$$
$$: \texttt{nat} \to \texttt{nat} \to \texttt{nat} \qquad \textbf{12 T-Abst}$$

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

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

It could be proven by induction that

$$\forall\,\overline{a} : \texttt{nat} \quad \overline{a} \equiv \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,f^a\,x$$

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

$$\twoheadrightarrow_\beta \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,\overline{n}\,\alpha(\overline{m}\,\alpha\,f)\,x$$

$$\twoheadrightarrow_\beta \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,\overline{n}\,\alpha(\lambda u : \alpha.\,f^m\,u)\,x$$

$$\twoheadrightarrow_\beta \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.(\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,f^n\,x)(\lambda u : \alpha.\,f^m\,u)\,x$$

$$\twoheadrightarrow_\beta \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.(\lambda u : \alpha.\,f^m\,u)^n\,x$$

By induction this can be further beta-reduced to

$$\twoheadrightarrow_\beta \lambda\alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\,f^{mn}\,x \equiv \overline{m\,n}$$

∎

Problem

(3.14) We present the Church-Encoded Boolean:

$$\texttt{bool} \in \mathbb{T}_2 := \Pi\alpha : *.\alpha \to \alpha \to \alpha$$
$$\text{true} \equiv \lambda\alpha : *.\lambda x, y : \alpha.\,x$$
$$\text{false} \equiv \lambda\alpha : *.\lambda x, y : \alpha.\,y$$

Construct a $\lambda 2$ term neg such that neg true $=_\beta$ false and neg false $=_\beta$ true.

Solution.

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

Neg True.

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

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

$$\xrightarrow[\beta]{} \lambda \alpha : *.\, \text{false } \alpha \xrightarrow[\eta]{} \text{false}$$

∎

Neg False.

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

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

$$\xrightarrow[\beta]{} \lambda \alpha : *.\, \text{true } \alpha \xrightarrow[\eta]{} \text{true}$$

∎

Problem

(3.15) Define

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

And reduce M true true, M true false, M false true, M false false, and decide which logical operator is represented by M.

Solution.

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

$$\xrightarrow[\beta]{} \lambda \beta : *.\lambda x, y : \beta.\, \text{true } \beta(\text{true } \beta\, x\, y)(\text{true } \beta\, y\, y)$$

$$\xrightarrow[\beta]{} \lambda \beta : *.\lambda x, y : \beta.(\text{true } \beta\, x\, y)$$

$$\xrightarrow[\beta]{} \lambda \beta : *.\lambda x, y : \beta.\, x \underset{\alpha}{\equiv} \text{true}$$

$$M \text{ true false} \equiv (\lambda u, v : \texttt{bool}.\lambda \beta : *.\lambda x, y : \beta.\, u\, \beta(v\, \beta\, x\, y)(v\, \beta\, y\, y))\text{ true false}$$

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

$$\xrightarrow[\beta]{} \lambda \beta : *.\lambda x, y : \beta.(\text{false } \beta\, x\, y)$$

$$\xrightarrow[\beta]{} \lambda \beta : *.\lambda x, y : \beta.\, y \underset{\alpha}{\equiv} \text{false}$$

$$M \text{ false true} \equiv (\lambda u, v : \texttt{bool}.\lambda\beta : * .\lambda x, y : \beta.\ u\ \beta(v\ \beta\ x\ y)(v\ \beta\ y\ y))\ \text{false true}$$

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.\ \text{false}\ \beta(\text{true}\ \beta\ x\ y)(\text{true}\ \beta\ y\ y)$$

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.(\text{true}\ \beta\ y\ y)$$

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.\ y \equiv_{\alpha} \text{false}$$

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

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.\ \text{false}\ \beta(\text{false}\ \beta\ x\ y)(\text{false}\ \beta\ y\ y)$$

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.(\text{false}\ \beta\ y\ y)$$

$$\twoheadrightarrow_{\beta} \lambda\beta : * .\lambda x, y : \beta.\ y \equiv_{\alpha} \text{false}$$

Therefore M is equivalent to logical AND.

Problem

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

Solution.

$$\text{OR} \equiv \lambda u, v : \texttt{bool}.\lambda\beta : * .\lambda x, y : \beta.\ u\ \beta\ x\ (v\ \beta\ x\ y)$$

$$\text{XOR} \equiv \lambda u, v : \texttt{bool}.\lambda\beta : * .\lambda x, y : \beta.\ u\ \beta(v\ \beta\ y\ x)(v\ \beta\ x\ y)$$

$$\text{IMP} \equiv \lambda u, v : \texttt{bool}.\lambda\beta : * .\lambda x, y : \beta.\ u\ \beta(v\ \beta\ x\ y)\ x$$

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

Problem

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

Solution.

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

Proof.

$$\text{isZero } \overline{0} \equiv (\lambda n : \mathtt{nat}.\, n \; \mathtt{bool} \; (\lambda u : \mathtt{bool}.\, \text{false}) \; \text{true})\overline{0}$$

$$\underset{\beta}{\twoheadrightarrow} (\lambda \alpha : *.\lambda f : \alpha \to \alpha.\lambda x : \alpha.\, x) \; \mathtt{bool} \; (\lambda u : \mathtt{bool}.\, \text{false}) \; \text{true}$$

$$\underset{\beta}{\twoheadrightarrow} (\lambda f : \mathtt{bool} \to \mathtt{bool}.\lambda x : \mathtt{bool}.\, x)(\lambda u : \mathtt{bool}.\, \text{false}) \; \text{true}$$

$$\underset{\beta}{\twoheadrightarrow} \text{true}$$

∎

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

Problem

(3.18 a) Define type

$$\mathtt{tree} \equiv \Pi\alpha : * .(\mathtt{bool} \to \alpha) \to (\mathtt{bool} \to \alpha \to \alpha \to \alpha) \to \alpha$$

Then we construct an inhabitant

$$\lambda\alpha. *.\lambda u : \mathtt{bool} \to \alpha.\lambda v : \mathtt{bool} \to \alpha \to \alpha \to \alpha.\, M$$
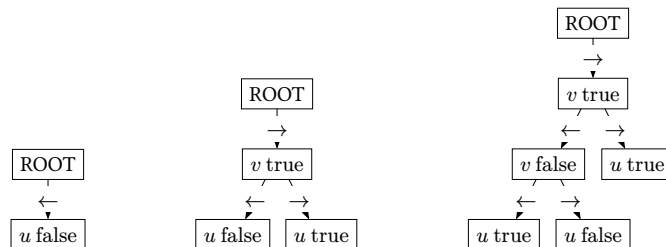
Is a node of a binary tree. Sketch graphs of trees where M is

$$u \text{ false}$$
$$v \text{ true } (u \text{ false})(u \text{ true})$$
$$v \text{ true } (u \text{ true})(v \text{ false } (u \text{ true})(u \text{ false}))$$

Solution. A binary tree is usually defined as this:

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

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

(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.

$$\text{leaf} := \lambda p : \mathtt{bool}.\lambda s, t : \mathtt{tree}.$$
$$\lambda \alpha.\lambda u . \; \mathtt{bool} \to \alpha.\lambda v : \mathtt{bool} \to \alpha \to \alpha \to \alpha.$$
$$v \, p \, (s \, \alpha \, u \, v)(t \, \alpha \, u \, v)$$

Proof. We suppose

$$s \equiv \lambda \beta : * .\lambda u_s : \mathtt{bool} \to \beta.\lambda v_s : \mathtt{bool} \to \beta \to \beta \to \beta. \; S$$
$$t \equiv \lambda \gamma : * .\lambda u_t : \mathtt{bool} \to \gamma.\lambda v_t : \mathtt{bool} \to \gamma \to \gamma \to \gamma. \; T$$

We want

$$\text{leaf} \, p \, s \, t \underset{\beta}{=} \lambda \alpha : * .\lambda u : \mathtt{bool} \to \alpha.\lambda v : \mathtt{bool} \to \alpha \to \alpha \to \alpha. \, v \, p \, S \, T$$

For compactness we denote

$$\alpha : * \vdash \tau_{\text{mkleaf}} \equiv \mathtt{bool} \to \alpha$$
$$\alpha : * \vdash \tau_{\text{mknode}} \equiv \mathtt{bool} \to \alpha \to \alpha \to \alpha$$

By beta reduction we have

$$\text{leaf} \, p \, s \, t \equiv (\lambda p : \mathtt{bool}.\lambda s, t : \mathtt{tree}....) \, p \, s \, t$$
$$\underset{\beta}{\twoheadrightarrow} (\lambda \alpha : * .\lambda u .\tau_{\text{mkleaf}}.\lambda v : \tau_{\text{mknode}}. \, v \, p \, (s \, \alpha \, u \, v)(t \, \alpha \, u \, v))$$
$$\underset{\beta}{\twoheadrightarrow} (\lambda \alpha : * .\lambda u .\tau_{\text{mkleaf}}.\lambda v : \tau_{\text{mknode}}. \, v \, p \, (s \, \alpha \, u \, v)(t \, \alpha \, u \, v))$$
$$\underset{\beta}{\twoheadrightarrow} \lambda \alpha : * .\lambda u .\tau_{\text{mkleaf}}.\lambda v : \tau_{\text{mknode}}. \, v \, p \, S \, T$$

∎

(3.19) If $\Gamma \vdash L : \sigma$, then Γ 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 deducted $L : \sigma$.

Case 1 : T-Var. The premise is that Γ is a $\lambda 2$ context. ∎

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

$$\Gamma \vdash M : \tau \to \sigma \qquad \Gamma \vdash N : \tau$$

for some $\tau \in \mathbb{T}_2$. By the inductive hypothesis on any premise Γ 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 $\sigma \equiv \alpha \to \beta$. By the inductive hypothesis, $\Gamma, x : \alpha$. By the recursive definition of $\lambda 2$ contexts, for some valid context Δ, $\forall\, n \in \text{dom }\Delta$, n could not depende on statement declared after n in the context. Therefore, no statement in Γ could depend on $x : \alpha$. Therefore, Γ is a valid context. ∎

Case 4 : T-Form. The premise is that Γ 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

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

By the inductive hypothesis on the any premise Γ 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 Γ 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

$$\Gamma \vdash M : \tau \to \sigma \qquad \Gamma \vdash N : \tau$$

For some type τ. By the inductive hypothesis FV $M \subseteq \Gamma$ and FV $N \subseteq \text{dom } \Gamma$. Therefore 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 $\sigma \equiv \alpha \to \beta$. By the inductive hypothesis FV $M \subseteq \operatorname{dom} \Gamma \cup \{x\}$. Therefore

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

$\blacksquare$

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

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

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

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 FV $L = \text{FV } M$, thus FV $M = \text{FV } L \subseteq \Gamma$. $\blacksquare$

Problem

(3.21)   Give a recursive definition for FTV : $\mathbb{T}_2 \cup \Lambda_{\mathbb{T}2} \to \mathbb{V}_2$

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
FTV α	$\{\alpha\}$
FTV x	$\emptyset$
FTV $(A \to B)$	$\text{FTV}(A) \cup \text{FTV}(B)$
FTV $(\Pi\alpha : *.\, A)$	$\text{FTV}(A) \setminus \{\alpha\}$
FTV $(M\ N)$	$\text{FTV}(M) \cup \text{FTV}(N)$
FTV $(\lambda x : A.\, M)$	$\text{FTV}(A) \cup \text{FTV}(M)$
FTV $(M\ B)$	$\text{FV}(M) \cup \text{FTV}(B)$
FTV $(\lambda\alpha : *.\, M)$	$\text{FTV}(M) \setminus \{\alpha\}$

—

Completed Dec 16 10:11 pm.