Solution Manual

Collected Problem Solutions

Author Information

Amir Faridi

amirfaridi2002@gmail.com

August 24, 2025

Version 1.0

Contents

Part 1

Domain Theoretic Foundations of Functional Programming

Chapter 1

1-Introduction

Chapter 2

2-PCF and its Operational Semantics

Problem 2.1 (page 14)

Show that the σ with $\Gamma \vdash M : \sigma$ is uniquely determined by Γ and M.

Solution

We prove this by induction on the structure.

- if $M \equiv x$ (variable), then it must be by the variable rule: $\Gamma', x : \sigma\Delta' \vdash x : \sigma$; thus σ must be unique by the definition of the context Γ. ($\Gamma \equiv x_1 : \sigma_1, ..., x_n : \sigma_n$, where x_i are pairwise distinct variables).
- if $M \equiv Z$ (zero), then it must be derived by the zero rule: $\Gamma \vdash Z : \mathbb{N}$; thus its type is unique.
- if $M \equiv (\lambda x : \sigma.M)$, then it must be derived by the abstraction rule: $\frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x:\sigma.M):\sigma\to\tau}$. By IH, M and x have unique types τ and σ, respectively. Thus, the type of the abstraction is uniquely determined as $\sigma \to \tau$.
- if $M \equiv (M(N))$, then by the application rule, we would have $\frac{\Gamma \vdash M:\sigma\to\tau \;\; \Gamma \vdash N:\sigma}{\Gamma \vdash M(N):\tau}$. By IH, M and N have unique types $\sigma \to \tau$ and σ, respectively. Thus, the type of the application $M(N)$ is uniquely determined as τ.
- Same goes for the other cases (*succ*, *pred*, Y_σ, and ifz).

Problem 2.2 (Page 16, Lemma 2.1.)

The evaluation relation $\Downarrow$ is deterministic, i.e. whenever $M \Downarrow V$ and $M \Downarrow W$ then $V \equiv W$

Solution

We prove this by induction on the structure of the derivation.
Base cases.

- By the rules of the BigStep semantics for PCF, the lemma for the following base cases is trivial:
  - $M \equiv x$, then $x \Downarrow x$. So V and W can only be x; thus, $V \equiv W \equiv x$.
  - $M \equiv \lambda x : \sigma.M$, then $\lambda x : \sigma.M \Downarrow \lambda x : \sigma.M$.
  - $M \equiv \underline{0}$, then $\underline{0} \Downarrow \underline{0}$.

Inductive Steps.

- If $M \equiv succ(M)$, then it must be derived by the rule $\frac{M \Downarrow \underline{n}}{succ(M) \Downarrow \underline{n+1}}$. Then we would have $V \equiv \underline{n+1}$ and $W \equiv \underline{m+1}$ since the successor rule is the only way to derive $succ(M)$. By IH, we know that $\underline{n} = \underline{m}$, thus $\underline{n+1} = \underline{m+1}$, and hence $V = W$.
- If $M \equiv M(N)$. The derivation for $M(N)$ must be of the form $\frac{M \Downarrow \lambda x:\sigma.E \quad E[N/x] \Downarrow V}{M(N) \Downarrow V}$. A second derivation for $M(N)$ must use the same rule. i.e., $\frac{M \Downarrow \lambda x:\sigma'.E' \quad E'[N/x] \Downarrow W}{M(N) \Downarrow W}$. But then by IH, we would have $\lambda x : \sigma.E \equiv \lambda x : \sigma'.E'$. So $\sigma \equiv \sigma'$ and $E = E'$. Now, we have $E[N/x] \Downarrow V$ and $E[N/x] \Downarrow W$. By the IH on the sub-derivation for $E[N/x]$, we conclude $V \equiv W$.
- If $M \equiv pred(M)$, then the rules are $\frac{M \Downarrow \underline{0}}{pred(M) \Downarrow \underline{0}}$ and $\frac{M \Downarrow \underline{n+1}}{pred(M) \Downarrow \underline{n}}$. For the derivation $pred(M) \Downarrow V$, we must have a sub-derivation for $M \Downarrow \underline{x}$ for some numeral $\underline{x}$. Similarly, for $pred(M) \Downarrow W$, we must have a sub-derivation for $M \Downarrow \underline{y}$ for some numeral $\underline{y}$. By the IH on the sub-derivation for M, we can conclude that $\underline{x} \equiv \underline{y} \equiv k$.

  Let's examine k. If $k \equiv \underline{0}$, then both derivations must be $\frac{M \Downarrow \underline{0}}{pred(M) \Downarrow \underline{0}}$. Thus, $V \equiv W \equiv \underline{0}$. Same argument is valid for the case $k \equiv \underline{n+1}$.
- The other cases (Y_σ, and both cases of ifz) can be proved likewise.

Problem 2.3 (Page 16, Theorem 2.2, Subject Reduction)

If $\Gamma \vdash M : \sigma$ and $M \Downarrow V$ then $\Gamma \vdash V : \sigma$.

Solution

We prove this by induction on the structure of the derivation of $M \Downarrow V$.
Base cases.

- If $M \equiv x$, then if $\Gamma \vdash x : \sigma$, then the goal is trivial since $x \Downarrow x$.

  Same goes for the other base cases ($\lambda x : \sigma.M$, $\underline{0}$).

Inductive Steps.

- If $M \equiv succ(M)$, then the only evaluation rule is $\frac{M \Downarrow \underline{n}}{succ(M) \Downarrow \underline{n+1}}$; so $V \equiv \underline{n+1}$. We are given $\Gamma \vdash succ(M) : \sigma$ and by the typing rule for $succ$, we have $\Gamma \vdash M : nat$.

  Now, from $M \Downarrow \underline{n}$ and $\Gamma \vdash M : nat$, by IH, $\Gamma \vdash \underline{n} : nat$; thus, by the typing rule for $succ$, we have $\Gamma \vdash \underline{n+1} : nat$.

- If $M \equiv M(N)$, the evaluation rule is $\frac{M \Downarrow \lambda x{:}\sigma.E \quad E[N/x] \Downarrow V}{M(N) \Downarrow V}$. We are given $\Gamma \vdash M(N) : \sigma$ and by the typing rule for application, we have $\Gamma \vdash M : \sigma \to \tau$ and $\Gamma \vdash N : \sigma$. Now, we have $M \Downarrow \lambda x : \sigma.E$ and $\Gamma \vdash M : \sigma \to \tau$ and by IH, we have $\Gamma \vdash \lambda x : \sigma.E : \sigma \to \tau$. The typing rule for abstraction is $\frac{\Gamma, x{:}\sigma \vdash E : \tau}{\Gamma \vdash \lambda x{:}\sigma.E : \sigma \to \tau}$, so we have $\Gamma, x : \sigma \vdash E : \tau$. Using the substitution lemma, we can conclude that $\Gamma \vdash E[N/x] : \tau$. Now, from $E[N/x] \Downarrow V$ and $\Gamma \vdash E[N/x] : \tau$, by IH, we have $\Gamma \vdash V : \tau$.

- Same goes for the other cases (both cases of $pred$, Y_σ, and both cases of ifz).

Problem 2.4 (Page 17)

$M \Downarrow V$ iff $M \rhd^* V$.

Solution

In order to prove this, we prove the following two lemmas:

(a). if $M \Downarrow V$ then $M \rhd^* V$
(b). if $M \rhd N$ then for all values V, if $N \Downarrow V$, then $M \Downarrow V$.

Then, applying **(b)** iteratively, it follows that:

(c). if $M \rhd^* N$ and $N \Downarrow V$, then $M \Downarrow V$.
and by **(a)** and **(c)**, we can conclude that $M \Downarrow V$ iff $M \rhd^* V$.
Proof of (a). We prove this by induction on the structure of the derivation of $M \Downarrow V$.
Base cases.

- All the base cases ($M \equiv x$, $M \equiv \lambda x : \sigma.M$, and $M \equiv \underline{0}$) are trivial by the reflexivity of $\rhd^*$.

Inductive Steps.

- If $M \equiv succ(M)$, then the rule is $\frac{M \Downarrow \underline{n}}{succ(M) \Downarrow \underline{n+1}}$; thus, $V \equiv \underline{n+1}$. By IH on $M \Downarrow \underline{n}$, we have $M \rhd^* \underline{n}$. Now, there are two cases.
  - If $M \equiv \underline{n}$, then $succ(M) \equiv succ(\underline{n}) \equiv \underline{n+1}$. So $M \rhd^* V$.
  - If $M \rhd M_1 \rhd \ldots \rhd \underline{n}$, then by the congruence rule for $succ$, we would have $succ(M) \rhd succ(M_1) \rhd \ldots \rhd succ(\underline{n}) \equiv \underline{n+1}$. So, $succ(M) \rhd^* \underline{n+1} \equiv V$.
- If $M \equiv M(N)$, then the rule is $\frac{M \Downarrow \lambda x{:}\sigma.E \quad E[N/x] \Downarrow V}{M(N) \Downarrow V}$. Then, by IH on $M \Downarrow \lambda x : \sigma.E$, we have $M \rhd^* \lambda x : \sigma.E$, and on $E[N/x] \Downarrow V$, we have $E[N/x] \rhd^* V$.

If $M \equiv \lambda x:\sigma.E$, then $M(N) \equiv (\lambda x:\sigma.E)(N)$. By small-step rule $(\lambda x:\sigma.E)(N) \rhd E[N/x]$. Since $E[N'/x] \rhd^* V$, we have $M(N) \rhd E[N/x] \rhd^* V$, so $M(N) \rhd^* V$.

If $M \rhd M_1 \rhd \cdots \rhd \lambda x:\sigma.E$, then by the congruence rule for application $M(N) \rhd M_1(N) \rhd \cdots \rhd (\lambda x:\sigma.E)(N)$. Then $(\lambda x:\sigma.E)(N) \rhd E[N/x]$, and $E[N/x] \rhd^* V$. Then we would have: $M(N) \rhd^* (\lambda x:\sigma.E)(N) \rhd E[N/x] \rhd^* V$. So $M(N) \rhd^* V$.

- The other cases are fairly similar to these.

Proof of (b). We prove this by induction on the structure of the derivation of $M \rhd V$.
Base cases.

- Assume $M \equiv (\lambda x:\sigma.E)(A)$ and $N \equiv E[A/x]$ ($M \rhd N$). Assume also $N \Downarrow V$, i.e., $E[A/x] \Downarrow V$. We know $\lambda x:\sigma.E \Downarrow \lambda x:\sigma.E$. By the rule: $\frac{\lambda x:\sigma.E \Downarrow \lambda x:\sigma.E \quad E[A/x] \Downarrow V}{(\lambda x:\sigma.E)(A) \Downarrow V}$. So $M \Downarrow V$.
- Assume $M \equiv Y_\sigma(E)$ and $N \equiv E(Y_\sigma(E))$. Assume also $N \Downarrow V$, i.e., $E(Y_\sigma(E)) \Downarrow V$. By the fixpoint rule: $\frac{E(Y_\sigma(E)) \Downarrow V}{Y_\sigma(E) \Downarrow V}$. So $M \Downarrow V$.
- Assume $M \equiv pred(\underline{0})$ and $N \equiv \underline{0}$. Assume also $N \Downarrow V$, i.e., $\underline{0} \Downarrow V$. This means $V \equiv \underline{0}$. We need to show $pred(\underline{0}) \Downarrow \underline{0}$. This holds by the rule $\frac{\underline{0} \Downarrow \underline{0}}{pred(\underline{0}) \Downarrow \underline{0}}$.
- Assume $M \equiv pred(\underline{k+1})$ and $N \equiv \underline{k}$. Assume also $N \Downarrow V$, i.e., $\underline{k} \Downarrow V$. This means $V \equiv \underline{k}$. We need to show $pred(\underline{k+1}) \Downarrow \underline{k}$. This holds by the rule $\frac{\underline{k+1} \Downarrow \underline{k+1}}{pred(\underline{k+1}) \Downarrow \underline{k}}$.
- Assume $M \equiv ifz(\underline{0}, E_1, E_2)$ and $N \equiv E_1$. Assume also $N \Downarrow V$, i.e., $E_1 \Downarrow V$. We need to show $ifz(\underline{0}, E_1, E_2) \Downarrow V$. This holds by the rule $\frac{\underline{0} \Downarrow \underline{0} \quad E_1 \Downarrow V}{ifz(\underline{0}, E_1, E_2) \Downarrow V}$.
- Assume $M \equiv ifz(\underline{k+1}, E_1, E_2)$ and $N \equiv E_2$. Assume also $N \Downarrow V$, i.e., $E_2 \Downarrow V$. We need to show $ifz(\underline{k+1}, E_1, E_2) \Downarrow V$. This holds by the rule $\frac{\underline{k+1} \Downarrow \underline{k+1} \quad E_2 \Downarrow V}{ifz(\underline{k+1}, E_1, E_2) \Downarrow V}$.

Inductive Steps.

- $\frac{M_1 \rhd M_2}{succ(M_1) \rhd succ(M_2)}$: $M = succ(M_1), N = succ(M_2)$, where $M_1 \rhd M_2$. Assume $N \Downarrow V$, i.e., $succ(M_2) \Downarrow V$. This implies $V \equiv \underline{k+1}$ and $M_2 \Downarrow \underline{k}$ for some k. By IH on the sub-derivation $M_1 \rhd M_2$: since $M_2 \Downarrow \underline{k}$, it follows that $M_1 \Downarrow \underline{k}$. Then, by the rule $\frac{M_1 \Downarrow \underline{k}}{succ(M_1) \Downarrow \underline{k+1}}$. So $M \equiv succ(M_1) \Downarrow \underline{k+1}$. Since $V \equiv \underline{k+1}$, we have $M \Downarrow V$.
- $\frac{M_1 \rhd M_2}{M_1(A) \rhd M_2(A)}$: $M = M_1(A)$, $N = M_2(A)$. Assume $M_2(A) \Downarrow V$. This means $M_2 \Downarrow \lambda x.E$ and $E[A/x] \Downarrow V$. By IH on $M_1 \rhd M_2$, since $M_2 \Downarrow \lambda x.E$, then $M_1 \Downarrow \lambda x.E$. Thus, by the rule, $M_1(A) \Downarrow V$.
- $\frac{M_1 \rhd M_2}{pred(M_1) \rhd pred(M_2)}$: $M = pred(M_1)$, $N = pred(M_2)$. Assume $pred(M_2) \Downarrow V$. This means $M_2 \Downarrow \underline{k}$ and V is $\underline{0}$ (if $k = 0$) or $\underline{k-1}$ (if $k > 0$). By IH on $M_1 \rhd M_2$, since $M_2 \Downarrow \underline{k}$, then $M_1 \Downarrow \underline{k}$. Thus, by the rule for $pred$, $pred(M_1) \Downarrow V$.
- $\frac{M_1 \rhd M_2}{ifz(M_1,N_1,N_2) \rhd ifz(M_2,N_1,N_2)}$: $M = ifz(M_1,N_1,N_2)$, $N = ifz(M_2,N_1,N_2)$. Assume $ifz(M_2,N_1,N_2) \Downarrow V$. This means $M_2 \Downarrow \underline{k}$ and either $N_1 \Downarrow V$ (if $k = 0$) or $N_2 \Downarrow V$ (if $k > 0$). By IH on $M_1 \rhd M_2$, since $M_2 \Downarrow \underline{k}$, then $M_1 \Downarrow \underline{k}$. Thus, by the rule for ifz, $ifz(M_1,N_1,N_2) \Downarrow V$.

Problem 2.5 (Page 19)

Show that the applicative relation $\underset{\sim}{\subset}_\sigma$ is a preorder on Prg_σ, i.e. that $\underset{\sim}{\subset}_\sigma$ is reflexive and transitive.

Solution

-Relfexivity. We need to show that for any closed PCF term M of type σ, $M \sqsubseteq_\sigma M$.
Base Case. For $M \in Prg_{nat}, M \sqsubseteq_{nat} M$ means that $\forall n \in \mathbb{N}, M \Downarrow \underline{n} \Rightarrow M \Downarrow \underline{n}$. This is trivially true.
Inductive Case. For $M \in Prg_{\sigma\to\tau}, M \sqsubseteq_{\sigma\to\tau} M$ means that $\forall P \in Prg_\sigma, M(P) \sqsubseteq_\tau M(P)$, which holds by IH.
-Transitivity. We need to show that for any closed PCF terms M, N, K of type σ, if $M \sqsubseteq_\sigma N$ and $N \sqsubseteq_\sigma K$, then $M \sqsubseteq_\sigma K$.
Base Case. For $M, N, K \in Prg_{nat}$, assume $M \sqsubseteq_{nat} N$ and $N \sqsubseteq_{nat} K$. Then, by definition, we have the followings:

- $\forall n \in \mathbb{N}, M \Downarrow \underline{n} \Rightarrow N \Downarrow \underline{n}$
- $\forall n \in \mathbb{N}, N \Downarrow \underline{n} \Rightarrow K \Downarrow \underline{n}$

Thus, if $M \Downarrow \underline{n}$ then $K \Downarrow \underline{n}$, which means $M \sqsubseteq_{nat} K$.
Inductive Case. For $M, N, K \in Prg_{\sigma\to\tau}$, assume $M \sqsubseteq_{\sigma\to\tau} N$ and $N \sqsubseteq_{\sigma\to\tau} K$. Then, by definition, we have the followings:

- $\forall P \in Prg_\sigma, M(P) \sqsubseteq_\tau N(P)$
- $\forall P \in Prg_\sigma, N(P) \sqsubseteq_\tau K(P)$

Thus, we would have $\forall P \in Prg_\sigma, M(P) \sqsubseteq K(P)$.

Chapter 3

3-The Scott Model of PCF

Problem 3.1 (Page 26)

Show that (Scott) continuous functions between predomains are always monotonic.

Solution

Let $f : (A, \sqsubseteq_A) \to (B, \sqsubseteq_B)$ be a Scott continuous function between predomains. For any $x, y \in A$ with $x \leq y$, we have that $X = \{x, y\}$ is a directed subset of A. The supremum of the set X is obviously y. Thus $\bigsqcup X = y$, and since f is continuous, $f(\bigsqcup X) = f(y) = \bigsqcup f(\{x, y\})$. Hence, $f(x) \leq f(y)$.

Problem 3.2 (Page 26, Theorem 3.3)

Let $(A_i | i \in I)$ be a family of predomains. Then their product $\prod_{i \in I} A_i$ is a predomain under the componentwise ordering, and the prodjections $\pi_i : \prod_{i \in I} A_i \to A_i$ are Scott continuous. If, moreover, all A_i are domains then so is their product $\prod_{i \in I} A_i$.

Solution

Let $D = \prod_{i \in I} A_i = \{f : I \to \bigcup_{i \in I} A_i | \forall i \in I, f(i) \in A_i\}$. We need to show that $(D, \sqsubseteq_D)$ is a poset, and every directed subset of D has a least upper bound. Note that the order $\sqsubseteq_D$ is defined as follows:

$$(d_i)_{i \in I} \sqsubseteq_D (d'_i)_{i \in I} \quad iff \quad \forall i \in I, d(i) \sqsubseteq_{A_i} d'(i)$$

We now show that $(D, \sqsubseteq_D)$ forms a poset.

- **Reflexivity:** For any $(d_i)_{i \in I} \in D, d_i \sqsubseteq_{A_i} d_i, \forall i \in I$ since each A_i is a poset. Thus $(d_i)_{i \in I} \sqsubseteq_D (d_i)_{i \in I}$
- **Transitivity:** Assume $(d_i)_{i \in I} \sqsubseteq_D (d'_i)_{i \in I}$ and $(d'_i)_{i \in I} \sqsubseteq_D (d''_i)_{i \in I}$. And by each A_i being

transitive, it follows immediately that $d_i \sqsubseteq_{A_i} d_i''$ for all $i \in I$. Therefore, $(d_i)_{i\in I} \sqsubseteq_D (d_i'')_{i\in I}$.

- **Antisymmetry:** Similar to the previous case, it follows immediately from the fact that each A_i is antisymmetric.

Now, suppose that $X \subseteq D = \prod_{i\in I} A_i$ is a directed subset. define $X_i = \{\pi_i(x) | x \in X\}$, that is, the projection of X to A_i. X_i is directed since X is directed. Moreover, X_i has a least upper bound $\bigsqcup X_i \in A_i$. Define $z \in D$ with $z_i = \bigsqcup X_i$ for each $i \in I$. By construction, it is obvious that z is the least upper bound of X in D. Thus, D is a predomain.

Problem 3.3 (Page 27)

Prove that the evaluation map $ev : [A_1 \to A_2] \times A_1 \to A_2$ with $ev(f,a) = f(a)$ is continuous in each argument.

Solution

For the first argument, fix $a \in A_1$ and let $F \subseteq [A_1 \to A_2]$ be a directed set of continuous functions. By Theorem 3.5, we have $\bigsqcup F(a) = g(a) = \bigsqcup_{f\in F} f(a)$. Thus, $ev(\bigsqcup F, a) = g(a) = \bigsqcup_{f\in F} f(a) = \bigsqcup\{ev(f,a) | f \in F\}$.
Now, for the second argument, fix $f \in [A_1 \to A_2]$ and let $X \subseteq A_1$ be a directed set. Because f is continuous, we have $f(\bigsqcup X) = \bigsqcup\{f(x) | x \in X\}$. Thus,

$$ev(f, \bigsqcup X) = f(\bigsqcup X) = \bigsqcup\{f(x) | x \in X\} = \bigsqcup\{ev(f,x) | x \in X\}.$$

Hence, ev is continuous in both arguments and by Lemma 3.4, it is jointly continuous.

Problem 3.4 (Page 30)

Prove that $\Psi : [[D \to D] \to D] \times [D \to D] \to D : (F, f) \mapsto f(F(f))$ is continuous in each argument.

Solution

For the first argument, fix $f \in [D \to D]$ and let $\mathscr{F} \subseteq [[D \to D] \to D]$ be a directed set. We know that $(\bigsqcup \mathscr{F})(f) = \bigsqcup\{F(f) | F \in \mathscr{F}\}$. Thus,

$$\underbrace{f((\bigsqcup \mathscr{F})(f))}_{\Psi(\bigsqcup \mathscr{F}, f)} = f(\bigsqcup\{F(f) | F \in \mathscr{F}\}) = \bigsqcup\{f(F(f)) | F \in \mathscr{F}\}$$

For the second argument, fix $F \in [[D \to D] \to D]$ and let $X \subseteq [D \to D]$ be a directed set. We

know that $F(\bigsqcup X) = \bigsqcup\{F(f)|f \in X\}$. Thus,

$$f(F(\bigsqcup X)) = f(\bigsqcup\{F(f)|f \in X\}) = \bigsqcup\{f(F(f))|f \in X\}$$

Hence, Ψ is continuous in both arguments and by Lemma 3.4, it is jointly continuous.

Problem 3.5 (Page 33)

(β-equality). If, $\Gamma, x : \sigma \vdash M : \tau$ and $\Gamma \vdash N : \sigma$ then

$$[\![\Gamma \vdash (\lambda x : \sigma.M)(N)]\!] = [\![\Gamma \vdash M[N/x]]\!]$$

Solution

$[\![\Gamma \vdash M[N/x]]\!](\vec{d}) = [\![\Gamma, x : \sigma \vdash M]\!](\vec{d}, [\![\Gamma \vdash N]\!](\vec{d}))$ by Lemma 3.15.
Now, for the other side,

$$[\![\Gamma \vdash (\lambda x : \sigma.M)(N)]\!](\vec{d}) = ev([\![\Gamma \vdash \lambda x : \sigma.M]\!](\vec{d}), [\![\Gamma \vdash N]\!](\vec{d}))$$

$$= [\![\Gamma \vdash \lambda x : \sigma.M]\!](\vec{d})([\![\Gamma \vdash N]\!](\vec{d})) = [\![\Gamma, x : \sigma \vdash M]\!](\vec{d}, [\![\Gamma \vdash N]\!](\vec{d}))$$

Problem 3.6 (Page 34)

(η-equality). If, $\Gamma \vdash M : \sigma \to \tau$ then

$$[\![\Gamma \vdash \lambda x : \sigma.M(x)]\!] = [\![\Gamma \vdash M]\!]$$

for $x \notin Var(\Gamma)$.

Solution

$$\forall \vec{d} \in [\![\Gamma]\!], d' \in D_\sigma, [\![\Gamma \vdash \lambda x : \sigma.M(x)]\!](\vec{d})(d') = [\![\Gamma, x : \sigma \vdash M(x)]\!](\vec{d}, d')$$

$$= ev([\![\Gamma, x : \sigma \vdash M]\!](\vec{d}, d'), \underbrace{[\![\Gamma, x : \sigma \vdash x]\!](\vec{d}, d')}_{\pi_x(\vec{d}, d') = d'}) =^\star [\![\Gamma \vdash M]\!](\vec{d})(d')$$

For the last equation ($\star$), because $x \notin Var(\Gamma)$ and $\Gamma \vdash M$, we have

$$[\![\Gamma, x : \sigma \vdash M]\!](\vec{d}, d') = [\![\Gamma \vdash M]\!](\vec{d})$$

.

Chapter 4

4-Computational Adequacy

Chapter 5

5-Milner's Context Lemma

Problem 5.1 (Page 44)

Prove that $\leq_\sigma$ is closed under suprema of directed sets. That is, if $X \subseteq D_\sigma \times D_\sigma$ is directed and $X \subseteq \leq_\sigma$, meaning that for every $(x,y) \in X, x \leq_\sigma y$, then $\bigsqcup X \in \leq_\sigma$.

Solution

Let $X = \{(x_k, y_k) | k \in K\}$ be a directed subset of $D_\sigma \times D_\sigma$ such that for all $k \in K, (x_k, y_k) \in \leq_\sigma$. This means for each $k \in K$, we have $\forall P \in Prg_\sigma, y_k R_\sigma P \implies x_k R_\sigma P (\star)$.
Let $x = \bigsqcup_{k \in K} x_k$ and $y = \bigsqcup_{k \in K} y_k$ (Note that $\bigsqcup X = (x,y)$). We need to show $x \leq_\sigma y$. That is, $\forall P \in Prg_\sigma y R_\sigma P \implies x R_\sigma P$.
Let $P \in Prg_\sigma$ be an arbitrary closed PCF term and that $y R_\sigma P$. Now, for each $k \in K$, we have $y_k \sqsubseteq y$ and by Lemma 4.2(1), we get $y_k R_\sigma P$ for all $k \in K$. Using $(\star)$, we have $x_k R_\sigma P$ for each $k \in K$.
By Lemma 4.2(2), $R_\sigma P$ is closed under directed suprema and $\{x_k | k \in K\}$ is a directed subset of D_σ whose elements are all in $R_\sigma P$. So, their suprema x must also be in $R_\sigma P$, meaning that $x R_\sigma P$.

Chapter 6

6-The Full Abstraction Problem

Chapter 7

7-Logical Relations

Problem 7.1 (Page 52)

(Theorem 7.2 When M is Variable) Let R be a logical relation of arity W on the Scott model of PCF. Then for λ-terms $x_1 : \sigma_1, ..., x_n : \sigma_n \vdash x_k : \sigma_k$ for some k and $d_j \in R_{\sigma_j}$ for $j = 1, ..., n$ it holds that

$$\underline{\lambda} i \in W.[\![x_1 : \sigma_1, ..., x_n : \sigma_n \vdash x_k]\!](\vec{d}(i)) \in R_{\sigma_k}$$

Solution

By premise, $d_k \in R_{\sigma_k}$. By definition, the goal reduces to $d_k \in R_{\sigma_k}$!

Part 2

Type Theory and Formal Proof

Chapter 8

1-Untyped Lambda Calculus

Problem 8.1 (1.1)

Apply Notation 1.3.10 on the follwoing λ-terms. SO, remove parentheses and combine *lambda*-abstractions:

(a) $(\lambda x.(((xz)y)(xx)))$

(b) $((\lambda x.(\lambda y.(\lambda z.(z((xy)z)))))(\lambda u.u))$

Solution

(a) $(\lambda x.(((xz)y)(xx))) = \lambda x.((xz)y)(xx) = \lambda x.(xz)y(xx) = \lambda x.(xz)y(xx) = \lambda x.xzy(xx)$

(b) Final answer: $\lambda xyz.z((xy)z)(\lambda u.u)$

Problem 8.2 (1.2)

Find out for each of the following λ-terms whether it is α-equivalent, or not, to $\lambda x.x(\lambda x.x)$:

(a) $\lambda y.y(\lambda x.x)$

(b) $\lambda y.y(\lambda x.y)$

(c) $\lambda y.y(\lambda y.x)$

Solution

The original term is $\lambda x.x(\lambda x.x) =_\alpha \lambda a.a(\lambda b.b)$. With this clarification on the original term, it is obvious that it is α-equivalent to only the first term.

Problem 8.3 (1.3)

Use the definition of $=_\alpha$ to prove that $\lambda x.x(\lambda z.y) =_\alpha \lambda z.z(\lambda z.y)$, in spite of the fact that z occurs as a binding variable in $x(\lambda z.y)$.

Solution

$\lambda z.z(\lambda z.y) =_\alpha \lambda x.(z(\lambda z.y))^{z\to x}$, because $x \notin FV(z.(\lambda z.y))$ and x is not a binding variable in $z(\lambda z.y)$.
Since $\lambda x.(z(\lambda z.y))^{z\to x} \equiv \lambda x.x(\lambda z.y)$, by symmetry of $=_\alpha$, it follows that $\lambda x.x(\lambda z.y) =_\alpha \lambda z.z(\lambda z.y)$.

Problem 8.4 (1.4)

Consider the following λ-term:

$$U := \underbrace{(\lambda z.zxz)}_{A}\underbrace{((\lambda y.xy)x)}_{B}$$

(a) Give a list of all subterms of U

(b) Draw the tree representation of U

(c) Find the set of all free variables of U by a calculation, as in Examples 1.4.2.

(d) Find out which of the following λ-terms are α-equivalent to U and give motivation why; also check which of them satisfies the Barendregt convention

i. $(\lambda y.yxy)((\lambda z.xz)x)$,
ii. $(\lambda x.xyx)((\lambda z.yz)y)$,
iii. $(\lambda y.yxy)((\lambda y.xy)x)$,
iv. $(\lambda v.(vx)v)((\lambda u.uv)x)$.

Solution

(a) $sub(U) = sub(A) \cup sub(B)$ where
$sub(A) = \{\lambda z.zxz, zxz, z, x, z, zx\}$ and
$sub(B) = \{(\lambda y.xy)x, \lambda y.xy, x, x, y, xy\}$

(b) Easy!

(c) $FV(U) = FV(A) \cup FV(B) = (FV(zxz)\backslash\{z\}) \cup ((FV(xy)\backslash\{y\}) \cup \{x\}) = \{x\}$

(d) *i.* Obviously, this is α-equivalent to the term U and it satisfies the Barendregt conven-

tion.

ii. The free variables x in B has changed to y. Hence, it is not equivalent to U. However, it satisfies the Barendregt convention.

iii. Yes, they are are equivalent since only the bound variable z is change to y in A. However, it does not satisfy the Barendregt convention for obvious reasons.

iv. The free variable x in B has changed to v, making them not to be equivalent. It also does not satisfy the Barendregt convention since v occurs as both bound and free variable.

Problem 8.5 (1.5)

Give the results of the following substitutions:

(*a*) $(\lambda x \cdot y(\lambda y \cdot xy))[y := \lambda z \cdot zx]$,

(*b*) $((xyz)[x := y])[y := z]$,

(*c*) $((\lambda x \cdot xyz)[x := y])[y := z]$,

(*d*) $(\lambda y \cdot yyx)[x := yz]$.

Solution

(*a*) First of all, $\lambda x.y(\lambda y.xy) =_\alpha \lambda a.y(\lambda b.ab)$. Hence, the result of the substitution would be $(\lambda a.(\lambda z.zx)(\lambda b.ab))$.

(*b*) $((xyz)[x := y])[y := z] = (yyz)[y := z] = (zzz)$

(*c*) $(\lambda x.xyz) =_\alpha (\lambda a.ayz)$, and the substitution would be:

$$((\lambda a.ayz)[x := y])[y := z] = (\lambda a.ayz)[y := z] = (\lambda a.azz)$$

(*d*) $(\lambda y.yyx) =_\alpha (\lambda a.aax) \Rightarrow (\lambda a.aax)[x := yz] = (\lambda a.aayz)$

Problem 8.6 (1.10)

We define the λ-terms `zero`, `one`, `two` (the first three so-called Church numerals), and the λ-terms `add` and `mult` (which mimic addition and multiplication of Church numerals) by:

$$\begin{aligned} \texttt{zero} &:= \lambda fx \cdot x, \\ \texttt{one} &:= \lambda fx \cdot fx, \\ \texttt{two} &:= \lambda fx \cdot f(fx), \\ \texttt{add} &:= \lambda mnfx \cdot mf(nfx), \\ \texttt{mult} &:= \lambda mnfx \cdot m(nf)x. \end{aligned}$$

(a) Show that `add one one` $\to_\beta$ `two`.

(b) Prove that `add one one` $\neq_\beta$ `mult one zero`.

Solution

(a) `add one one` $= (\lambda mnfx.mf(nfx))$ `one one` $\to_\beta$ $(\lambda nfx.$ `one` $f(nfx))$`one` $\to_\beta$ $\lambda fx.$`one`$f($`one`$fx) = \lambda fx.(\lambda fx.fx)f($`one`$fx) \to_\beta \lambda fx.f($`one`$fx) \to_\beta \lambda fx.f(fx) =$ `two`.

(b) We already know that `add one one` $\to_\beta$ `two`

$$\texttt{mult one zero} = (\lambda mnfx.m(nf)x)\texttt{one zero} \to_\beta \lambda fx.\texttt{one}(\texttt{zero}f)x$$

$$= \lambda fx.(\lambda fx.fx)(\lambda fx.x)fx \to_\beta \lambda fx.(\lambda fx.fx)(\lambda x.x)x \to_\beta \lambda fx.x = \texttt{zero}$$

Problem 8.7 (1.11)

The *successor* is the function mapping natural number n to $n+1$. It is represented in λ-calculus by $suc := \lambda mfx.f(mfx)$. Check the following for the Church numerals defined in the previous exercise:

(a) $suc\ zero =_\beta one$,

(b) $suc\ one =_\beta two$.

Solution

(a) `suc zero` $= (\lambda mfx.f(mfx))(\lambda fx.x) \to_\beta \lambda fx.f((\lambda fx.x)(fx)) \to_\beta \lambda fx.f(x) =$ `one`

(b) `suc one` $= (\lambda mfx.f(mfx))(\lambda fx.fx) \to_\beta \lambda fx.f((\lambda fx.fx)fx) \to_\beta \lambda fx.f(fx) =$ `two`

Problem 8.8 (1.12)

We define the λ-terms *true* and *false* (the *booleans*) and *not* (resembling the logical $\neg$-operator) by:

$$true := \lambda xy.x \quad \text{(so it happens that } true \equiv \mathsf{K}\text{)},$$

$$false := \lambda xy.y \quad \text{(and } false \equiv zero\text{)}.$$

$$not := \lambda z.z \ \ false \ \ true$$

Show that $not \ \ not \ \ p =_\beta p$ for all terms p, in each of the two cases:

(a) $p \twoheadrightarrow_\beta true$

(b) $p \twoheadrightarrow_\beta false$

Solution

It is sufficient to show that $not \ \ true \twoheadrightarrow_\beta false$ and $not \ \ false \twoheadrightarrow_\beta true$.

$$not \ \ true = (\lambda z.z(\lambda ab.b)(\lambda ab.a))(\lambda ab.a) \to_\beta (\lambda ab.a)(\lambda ab.b)(\lambda ab.a) \to_\beta \lambda ab.b = false$$

similarly, $not \ \ false \twoheadrightarrow_\beta true$.

Problem 8.9 (1.16)

Let M be a $\lambda - term$ with the following properties:

- M has a β-normal form.
- There exists a reduction path $M \equiv M_0 \to_\beta M_1 \to_\beta \ldots$ of infinite length.

(a) Prove that every M_i has a β-normal form.

(b) Give an example of a λ-term with the mentioned two properties and show that it satisfies these properties.

Solution

(a) Since M has a β-normal form, there is an L in β-nf such that $M =_\beta L$. By *CR*, there is an N such that $M \twoheadrightarrow_\beta N$ and $L \twoheadrightarrow_\beta N$. The latter and Lemma 1.9.2 imply that $L \equiv N$, hence $M \twoheadrightarrow_\beta L$. From $M \twoheadrightarrow_\beta M_i$ and $M \twoheadrightarrow_\beta L$ follows $M_i =_\beta L$. So, M_i has a β-nf, since L is in β-nf.

(b) On the one hand, $(\lambda u.v)\Omega$ has a β-nf. On the other hand, $(\lambda u.v)\Omega \to_\beta (\lambda u.v)\Omega \to_\beta \ldots$

Problem 8.10 (1.17)

Prove the following: if MN is strongly normalising, then both M and N are strongly normalising.

Solution

If M had an infinite reduction path, let's say $M \to_\beta M_1 \to_\beta \ldots$, then we would have the following reduction path: $MN \to_\beta M_1N \to_\beta M_2N \to_\beta \ldots$, but this is a contradiction since MN is strongly normalising. Same goes for the case if N had an infinite reduction path.

Problem 8.11 (1.18)

Let L, M and N be λ-terms such that $L =_\beta M$ and $L \twoheadrightarrow_\beta N$. Moreover, let N be in β-normal form. Prove that also $M \twoheadrightarrow_\beta N$.

Solution

Since $M =_\beta L$, then by corollary 1.9.9, there is an X such that $M \twoheadrightarrow_\beta X$ and $L \twoheadrightarrow_\beta X$. By Church-Rosser Theorem, since $L \twoheadrightarrow X$ and $L \twoheadleftarrow N$, there is some Y such that $N \twoheadrightarrow_\beta Y$ and $X \twoheadrightarrow_\beta Y$, but N is in $\beta - nf$ and hence, $N \equiv Y$. Now, we have $M \twoheadrightarrow_\beta X \twoheadrightarrow_\beta N$ and by the transitivity of $\twoheadrightarrow_\beta$, we have $M \twoheadrightarrow_\beta N$.

Chapter 9

2-Simply Typed Lambda Calculus

Problem 9.1 (2.10)

Prove that the following pre-typed λ-terms are legal, by giving derivations in (shortened) flag notation.

(a) $xz(yz)$,

(b) $\lambda x : (\alpha \to \beta) \to \beta \cdot x(yz)$,

(c) $\lambda y : \alpha \cdot \lambda z : \beta \to \gamma \cdot z(xyy)$,

(d) $\lambda x : \alpha \to \beta \cdot y(xz)z$.

Solution

I'll be using derivation tree instead of flag notation.

(a) Just simply take the following context, and apply them together!

$$\Gamma = \{x : \alpha \to \beta \to \gamma, y : \alpha \to \beta, z : \alpha\}$$

(b) Consider the following context, and apply them together before abstracting x.

$$\Gamma = \{y : \gamma \to (\alpha \to \beta), z : \gamma\}$$

(c) Consider the following context:

$$\Gamma = \{x : \alpha \to \alpha \to \beta\}$$

$$\dfrac{\dfrac{\Gamma, y : \alpha, z : \beta \to \gamma \vdash z : \beta \to \gamma \qquad \Gamma, y : \alpha, z : \beta \to \gamma \vdash xyy : \beta}{\Gamma, y : \alpha, z : (\beta \to \gamma) \vdash z(xyy) : \gamma}\ \text{(App. - three times)}}{\Gamma \vdash \lambda y : \alpha \cdot \lambda z : \beta \to \gamma \cdot z(xyy) : \alpha \to (\beta \to \gamma) \to \gamma}\ \text{(Abs. - twice)}$$

(d) Simply take the following context and apply the rules: $\Gamma = \{y : \beta \to \alpha \to \gamma, z : \alpha\}$

Problem 9.2 (2.13)

Find a term of type τ in context Γ, with:

(a) $\tau \equiv (\alpha \to \beta) \to \alpha \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \gamma$,

(b) $\tau \equiv \alpha \to (\alpha \to \beta) \to \gamma, \Gamma \equiv x : \alpha \to \beta \to \alpha \to \gamma$,

(c) $\tau \equiv (\alpha \to \gamma) \to (\beta \to \alpha) \to \gamma, \Gamma \equiv x : (\beta \to \gamma) \to \gamma$.

Give appropriate derivations.

Solution

In each case, giving derivation is fairy easy. So I'll settle for giving the term.

(a) $\Gamma \vdash \lambda f : (\alpha \to \beta) \cdot \lambda a : \alpha \cdot xa(fa) : \tau$

(b) $\Gamma \vdash \lambda a : \alpha . \lambda f : (\alpha \to \beta) . (xa)(fa)(a) : \tau$

(c) $\Gamma \vdash \lambda f : (\alpha \to \gamma) . \lambda g : (\beta \to \alpha) . x(\lambda b : \beta . f(gb)) : \tau$

Problem 9.3 (2.14)

Find an inhabitant of the type $\alpha \to \beta \to \gamma$ in the following context:

$$\Gamma \equiv x : (\gamma \to \beta) \to \alpha \to \gamma.$$

Give an appropriate derivation.
(Hint: if τ is inhabited, then also $\sigma \to \tau$ is inhabited.)

Solution

I only provide the term. Derivation is straight forward. $\lambda a : \alpha . \lambda b : \beta . x(\lambda c : \gamma . b)a : \alpha \to \beta \to \gamma$

Problem 9.4 (2.15)

Give the (var)- and $(appl)$-cases of the proof of Lemma 2.10.5 (1) (the 'Thinning Lemma').

Solution

1. **Var:** Assume we have a judgment derived by the (var) rule, that is, $\Gamma' \vdash x : \sigma$. This means that $x : \sigma \in \Gamma'$ and obviously we would have $x : \sigma \in \Gamma''$. Thus $\Gamma'' \vdash x : \sigma$.
2. **App:** Assume we have a judgment derived by the (app) rule, that is, $\Gamma' \vdash M : \sigma \to \tau$ and $\Gamma' \vdash N : \sigma$ are the premises, and $\Gamma' \vdash MN : \tau$ is the conclusion. By IH, we have $\Gamma'' \vdash M : \sigma \to \tau$ and $\Gamma'' \vdash N : \sigma$, and by the (app) rule, we have: $\Gamma'' \vdash MN : \tau$.

Problem 9.5 (2.16)

Prove Lemma 2.10.8 (the 'Subterm Lemma').

Solution

The proof is by induction on the structure of the derivation of the judgment $\Gamma \vdash M : \sigma$. We need to show that for every subterm L of M, there exists a context Γ' and a type ρ such that $\Gamma' \vdash L : \rho$.

- **Var:** If $M \equiv x$, the only subterm is x itself.
- **App:** If we have $\Gamma \vdash PQ : \sigma$, then the non-trivial subterms would be P and Q. The derivation is $\frac{\Gamma \vdash P:\rho\to\sigma \quad \Gamma \vdash Q:\rho}{\Gamma \vdash PQ:\sigma}$. By IH, all subterms of P and Q, are legal, and hence, proves the lemma for this case.
- **Abs:** Similarly one can prove for this case.

Problem 9.6 (2.17)

Prove Lemma 2.10.9 (the 'Uniqueness of Types Lemma').
(Hint: use Lemma 2.10.7 (the 'Generation Lemma')).

Solution

- **Var:** Assume $x : \sigma \in \Gamma$ and $x : \tau \in \Gamma$ (By the Generation lemma). By the definition of context (a list of declarations with different subjects), we must have $x : \sigma \equiv x : \tau$, and hence, $\sigma \equiv \tau$.
- **App. and Abs:** It is straightforward to assume two types for the derivations by Generation lemma, and then use IH to conclude the equivalences of the types!

Problem 9.7 (2.18)

Prove the Compatibility cases in the proof of Lemma 2.11.5.

Solution

- **App. Left:** $L \equiv MK$ and let $L \to_\beta L' \equiv M'K$ since $M \to'_\beta$. The judgments are $\Gamma \vdash MK : \rho$. By Generation lemma, we would have $\Gamma \vdash M : \sigma \to \rho$ and $\Gamma \vdash K : \sigma$. In this case, the premise of our reduction is $M \to_\beta M'$, and by IH, subject reduction holds for this premise. Therefore, $\Gamma \vdash M' : \sigma \to \rho$. Now, by applying the (app.) rule, we would have $\Gamma \vdash M'K : \rho$
- **App. Right:** Similar to the previous case.
- **Abstraction:** Here we have $L \equiv \lambda x : \tau.M \to_\beta L' \equiv \lambda x : \tau.M'$ because $M \to_\beta M'$. By use of IH and (abs.) rule, this case will be solved easily.

Chapter 10

3-Second Order Typed Lambda Calculus

Problem 10.1 (3.1)

How many $\lambda 2$-contexts are there consisting of the four declarations $\alpha : *$, $\beta : *$, $f : \alpha \to \beta$, $x : \alpha$?

Solution

All possible valid permutations:

1. $\alpha : \star, \beta : \star, f : \alpha \to \beta, x : \alpha$
2. $\beta : \star, \alpha : \star, f : \alpha \to \beta, x : \alpha$
3. $\alpha : \star, \beta : \star, x : \alpha, f : \alpha \to \beta$
4. $\beta : \star, \alpha : \star, x : \alpha, f : \alpha \to \beta$
5. $\alpha : \star, x : \alpha, \beta : \star, f : \alpha \to \beta$

Problem 10.2 (3.2)

Give a full (i.e. not-shortened) derivation in $\lambda 2$ to show that the following term is legal; use the flag format. (Cf. Example 3.1.1 (3).)

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

Solution

For convinience, I'll just write the steps down instead of using flag notation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma\vdash f:\alpha\to\beta \qquad \Gamma\vdash x:\alpha}{\Gamma\vdash fx:\beta}\,(\text{App.}) \qquad \Gamma\vdash g:\beta\to\gamma}{\alpha\beta\gamma:\star,f:\alpha\to\beta,g:\beta\to\gamma,x:\alpha\vdash g(fx):(\alpha\to\beta)\to(\beta\to\gamma)\to\alpha\to\gamma}\,(\text{App.})}{\alpha\beta\gamma:\star\vdash\lambda f:\alpha\to\beta,g:\beta\to\gamma,x:\alpha\cdot g(fx):(\alpha\to\beta)\to(\beta\to\gamma)\to\alpha\to\gamma}\,(\text{3Abs.})}{\vdash\lambda\alpha\beta\gamma:\star\cdot\lambda f:\alpha\to\beta,g:\beta\to\gamma,x:\alpha\cdot g(fx):\Pi\alpha\beta\gamma:\star\cdot(\alpha\to\beta)\to(\beta\to\gamma)\to\alpha\to\gamma}$$

Problem 10.3 (3.5)

Let $\bot\equiv\Pi\alpha:*.\alpha$ and $\Gamma\equiv\beta:*,x:\bot$.

(*a*) Prove that $\bot$ is legal.

(*b*) Find an inhabitant of β in context Γ.

(*c*) Give three not β-convertible inhabitants of $\beta\to\beta$ in context Γ, each in β-normal form.

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

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

Solution

(*a*) $\alpha:\star\vdash\alpha:\star\Longrightarrow\vdash\bot:\star$

(*b*) Obviously we have: $x\beta:\beta$.

(*c*) $\Gamma\vdash id\equiv\lambda y:\beta.y:\beta\to\beta$ $\quad\Gamma\vdash cons\equiv\lambda y:\beta.x\beta:\beta\to\beta$ and $\Gamma\vdash M\equiv\lambda y:\beta.x(\beta\to\beta)y:\beta\to\beta$ are three non-β-convertible inhabitants of $\beta\to\beta$ in Γ and all of them are in β-nf.

(*d*) $\lambda f:\beta\to\beta\to\beta.f(x\beta)(x\beta):(\beta\to\beta\to\beta)\to\beta$ since $(x\beta:\beta)$. Also, $x((\beta\to\beta\to\beta)\to\beta):(\beta\to\beta\to\beta)\to\beta$.

Problem 10.4 (3.14)

We may also introduce the polymorphic booleans in $\lambda 2$:

$$Bool \equiv \Pi\alpha : * \cdot \alpha \to \alpha \to \alpha,$$

$$True \equiv \lambda\alpha : * \cdot \lambda x, y : \alpha \cdot x,$$

$$False \equiv \lambda\alpha : * \cdot \lambda x, y : \alpha \cdot y.$$

Construct a $\lambda 2$-term $Neg : Bool \to Bool$ such that $Neg\ True =_\beta False$ and $Neg\ False =_\beta True$. Prove the correctness of your answer.

Solution

Define:

$$Neg \equiv \lambda b : Bool.\lambda\alpha : *.\lambda x, y : \alpha.b\ \alpha\ y\ x : Bool \to Bool$$

We show that $Neg : Bool \to Bool$ and that it satisfies the required properties.

$$Neg\ True =_\beta False$$

$$\begin{aligned}
Neg\ True &= (\lambda b : Bool.\lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.b\ \alpha\ y\ x)\ True \\
&\to_\beta \lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.True\ \alpha\ y\ x \\
&= \lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.(\lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.x)\ \alpha\ y\ x \\
&\to_\beta \lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.(\lambda x : \alpha.\lambda y : \alpha.x)\ y\ x \\
&\to_\beta \lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.(\lambda y : \alpha.y)\ x \\
&\to_\beta \lambda\alpha : *.\lambda x : \alpha.\lambda y : \alpha.y \\
&= False
\end{aligned}$$

Similarly, it can be shown that $Neg\ False =_\beta True$.

Problem 10.5 (3.20)

Prove the Free Variables Lemma for $\lambda 2$ (cf. Lemma 3.6.4): if $\Gamma \vdash L : \sigma$, then $FV(L) \subseteq dom(\Gamma)$.

Solution

The proof is by induction on the structure of the derivation of the judgment $\Gamma \vdash L : \sigma$.

- **Var:** We have $\Gamma \vdash x : \sigma$. The only free variable here is x itself, and it must be in Γ.
- **App:** In this case, we have $\frac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash MN:\tau}$ (by generation lemma). Now, by IH, $FV(M) \subseteq dom(\Gamma)$ and $FV(N) \subseteq dom(\Gamma)$. Thus, $FV(MN) = FV(M) \cup FV(N) \subseteq dom(\Gamma)$.
- **Abs:** Similar for this case!
- **Form:** In this case, we have $\Gamma \vdash B : \star$. All free variables of B must have been declared in Γ by the rule itself.
- **App2:** The rule is $\frac{\Gamma \vdash M:\Pi\alpha:\star.A \quad \Gamma \vdash B:\star}{\Gamma \vdash MB:A[\alpha:=B]}$ by generation lemma. By IH, $FV(M) \subseteq dom(\Gamma)$. Also, by the (form.) fule's conditions on the other premise, all free type variables of B are in $dom(\Gamma)$. Thus the lemma holds.
- **Abs2:** Fairly similar!

Chapter 11

4-Types Dependent on Types

Problem 11.1 (4.1)

Give a diagram of the tree corresponding to the complete tree derivation of line (16) of Section 4.5.

Solution

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\beta:*,\ \alpha:*\vdash\alpha:*}\text{(Var.+sort)}\quad\dfrac{}{\beta:*,\ \alpha:*\vdash\alpha:*}\text{(Var.+sort)}}{\beta:*,\ \alpha:*\vdash\alpha\to\alpha:*}\text{(Form.)}}{\beta:*\vdash\lambda\alpha:*.\ \alpha\to\alpha:*\to*}\text{(Abs2)}\qquad\dfrac{}{\beta:*\vdash\beta:*}\text{(Var.+sort)}}{\beta:*\vdash(\lambda\alpha:*.\ \alpha\to\alpha)\,\beta:*}\text{(App2.)}$$

Problem 11.2 (4.2)

Give complete $\lambda\underline{\omega}$-derivations, first in tree format and then in flag format (not shortened), of the following judgements:

(a) $\emptyset\vdash(*\to*)\to*:\Box$,

(b) $\alpha:*,\beta:*\vdash(\alpha\to\beta)\to\alpha:*$.

Solution

$$\dfrac{\dfrac{\dfrac{}{\vdash*:\Box}\text{(sort.)}\quad\dfrac{}{\vdash*:\Box}\text{(sort.)}}{\vdash*\to*:\Box}\text{(form.)}\qquad\dfrac{}{\vdash*:\Box}\text{(sort.)}}{\vdash(*\to*)\to*:\Box}\text{(form.)}$$

$$\dfrac{\dfrac{\dfrac{}{\alpha:*,\beta:*\vdash\alpha:*}\text{(WVS.)}\quad\dfrac{}{\alpha:*,\beta:*\vdash\beta:*}\text{(Var.+sort)}}{\alpha:*,\beta:*\vdash\alpha\to\beta:*}\text{(form.)}\quad\dfrac{}{\alpha:*,\beta:*\vdash\alpha:*}\text{(WVS.)}}{\alpha:*,\beta:*\vdash(\alpha\to\beta)\to\alpha:*}\text{(form.)}$$

WVS: These branches can be completed easily by using (weak.), (var.), and (sort) rules.

Problem 11.3 (4.3)

(*a*) Give a complete (i.e. not shortened) $\lambda\omega$-derivation in flag format of $\alpha,\beta:*,x:\alpha,y:\alpha\to\beta\vdash yx:\beta$.

(*b*) Give a shortened $\lambda\omega$-derivation in flag format of $\alpha,\beta:*,x:\alpha,y:\alpha\to\beta,z:\beta\to\alpha\vdash z(yx):\alpha$.

Solution

$$\dfrac{\dfrac{\text{Similar To The Branch On The Right}}{\Gamma\vdash x:\alpha}\quad\dfrac{\dfrac{}{\Gamma_1\vdash\alpha\to\beta:\star}\text{(FWV.)}}{\Gamma\vdash y:\alpha\to\beta}\text{(Var.)}}{\alpha\beta:\star,x:\alpha,y:\alpha\to\beta\vdash yx:\beta}\text{(App.)}$$

(FWV.) Can be solved easily by the use of the (form.), (var.), and the (weak.) rules. Part (b) of the problem is also very similar and straightforward.

Problem 11.4 (4.5)

Give a shortened $\lambda\omega$-derivation in flag format of the following judgement:

$$\alpha:*,x:\alpha\vdash\lambda y:\alpha.x:(\lambda\beta:*.\beta\to\beta)\alpha.$$

Solution

(a) $\boxed{\alpha : *}$

(b) $\quad \boxed{x : \alpha}$

(c) $\quad\quad \boxed{y : \alpha}$

(1) $\quad\quad x : \alpha$ (weak) on (b)

(2) $\quad\quad \lambda y : \alpha . x : \alpha \to \alpha$ (abst) on (1)

(d) $\quad \boxed{\beta : *}$

(3) $\quad\quad \beta \to \beta : *$ (form) on (d) and (d)

(4) $\quad\quad \lambda \beta : * . \beta \to \beta : * \to *$ (abst) on (3)

(5) $\quad\quad (\lambda \beta : * . \beta \to \beta)\, \alpha : *$ (appl) on (4) and (a)

(6) $\quad \lambda y : \alpha . x : (\lambda \beta : * . \beta \to \beta)\, \alpha$ (conv) on (2) and (5)

Problem 11.5 (4.6)

(a) Prove that there are no Γ and N in $\lambda\omega$ such that $\Gamma \vdash \Box : N$ is derivable.

(b) Prove that there are no Γ, M and N in $\lambda\omega$ such that $\Gamma \vdash M \to \Box : N$ is derivable.

Solution

(a.)

(b.) Proof by induction on the structure of the derivatiov tree of the judgement $\Gamma \vdash M \to \Box : N$. The last step in the derivation can only have been (weak.), (form.), or (conv.).

- **(weak.):** First premiss must have been of the form $\Gamma' \vdash M \to \Box : N$. By induction this is not derivable.
- **(form.):** Second premiss must have been $\Gamma \vdash \Box : N$. This is not derivable by Exercise 4.6(a).
- **(conv.):** First premiss must have been $\Gamma \vdash M \to \Box : L$. By induction this is not derivable.

Thus, $\Gamma \vdash M \to \Box : N$ is not derivable.

Problem 11.6 (4.7)

(a) Give $\lambda\underline{\omega}$-definitions of the notions legal term, statement, $\lambda\underline{\omega}$-context and domain.

(b) Formulate the following theorems for $\lambda\underline{\omega}$: Free Variables Lemma and Thinning Lemma.

Solution

(a)

- **Legal Term:** A $\lambda\underline{\omega}$-term is considered legal if there exists a context Γ and a type ρ such that the judgement $\Gamma \vdash M : \rho$ is derivable.
- **Statement:** A statement is an expression of one of the following two forms: 1. $M : A$, or 2. $A : s$, where M and A are expressions, and $s \in \{\star, \Box\}$.
- **$\lambda\underline{\omega}$-context:** Is a list of declaration defined inductively as follows:
  - is a valid $\lambda\underline{\omega}$-context.
  - If Γ is a valid $\lambda\underline{\omega}$-context, α is a type variable not in the domain of Γ, then $\Gamma, \alpha : \star$ is a valid $\lambda\underline{\omega}$-context.
  - If Γ is a valid $\lambda\underline{\omega}$-context, ρ is a type such that all its free type variables are in the domain of Γ, and x is a term variable not in the domain of Γ, then $\Gamma, x : \rho$ is a valid $\lambda\underline{\omega}$-context.
- **Domain:** The domain of a contex Γ is the list of all variables subjects declared in Γ.

(b) **Free Variables Lemma:** If $\Gamma \vdash M : \sigma$, then $FV(M) \subseteq \Gamma$.
Thinning Lemma: If $\Gamma \vdash M : \sigma$ and $\Gamma \subseteq \Sigma$, then $\Sigma \vdash M : \sigma$.

Chapter 12

5-Types Dependent on Terms

Problem 12.1 (5.1)

Give a diagram of the tree corresponding to the complete tree derivation of line (18) of Section 5.3.

Problem 12.2 (5.6)

Prove that $(A \Rightarrow (A \Rightarrow B)) \Rightarrow (A \Rightarrow B)$ is a tautology, (first) in natural deduction and (second) by means of a shortened λP-derivation.

Problem 12.3 (5.7)

Prove that the following propositions are tautologies by giving shortened λP-derivations:

(a) $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$,

(b) $((A \Rightarrow B) \Rightarrow A) \Rightarrow ((A \Rightarrow B) \Rightarrow B)$,

(c) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$.

Problem 12.4 (5.10)

Consider the following context:

$$\Gamma \equiv S : *, P : S \to *, f : S \to S, g : S \to S,$$

$$u : \Pi x : S.(P(fx) \to P(gx)), v : \Pi x, y : S.((Px \to Py) \to P(fx))$$

(cf. Notation 3.4.2). Let $M \equiv \lambda x : S.v(fx)(gx)(ux)$.

(a) Make a guess at which type N may satisfy $\Gamma \vdash M : N$.

(*b*) Demonstrate that the proof object M does indeed code a proof of the proposition N you have guessed, by elaborating the λP-derivation.

Problem 12.5 (5.11)

Let S be a set, with Q and R relations on $S \times S$, and let f and g be functions from S to S. Assume that:

- $\forall_{x,y \in S}(Q(x, f(y)) \Rightarrow Q(g(x), y))$,
- $\forall_{x,y \in S}(Q(x, f(y)) \Rightarrow R(x, y))$,
- $\forall_{x \in S}(Q(x, f(f(x))))$.

Prove that $\forall_{x \in S}(R(g(g(x)), g(x)))$ by giving a context Γ and finding a term M such that:

$$\Gamma \vdash M : \Pi x : S.R(g(g(x)))(g(x)).$$

Give the corresponding (shortened) λP-derivation.

Chapter 13

6-The Calculus of Construction

Problem 13.1 (6.2)

Let $\Gamma \equiv S : *, P : S \to *, A : *$. Prove by means of a flag derivation that the following expression is inhabited in λC with respect to Γ:

$$(\Pi x : S.(A \to Px)) \to A \to \Pi y : S.Py.$$

(You may shorten the derivation, as explained in Section 4.5.)

Problem 13.2 (6.5)

Let $\mathscr{J}$ be the following judgement:

$$S : * \vdash \lambda Q : S \to S \to *, \lambda x : S, Qxx : (S \to S \to *) \to S \to *.$$

(a) Give a shortened derivation of $\mathscr{J}$ and determine the smallest subsystem to which $\mathscr{J}$ belongs.

(b) We may consider the variable Q in $\mathscr{J}$ as expressing a relation on set S. How could you describe the subexpression $\lambda x : S, Qxx$ in this setting? And what is then the interpretation of the judgement $\mathscr{J}$?

Problem 13.3 (6.8)

(a) Let $\Gamma \equiv S : *, P : S \to *$. Find an inhabitant of the following type N in context Γ, and prove your answer by means of a shortened derivation:

$$N \equiv [\Pi\alpha : *.((\Pi x : S.(Px \to \alpha)) \to \alpha)] \to [\Pi x : S.(Px \to \bot)] \to \bot.$$

(b) Which is the smallest system in the λ-cube in which your derivation may be executed?

(c) The expression $\Pi\alpha : *.((\Pi x : S.(Px \to \alpha)) \to \alpha$ may be considered as an encoding of $\exists_{x \in S}(P(x))$. (We shall show this in Section 7.5.) In Section 7.1 we make plausible that $A \to \bot$ may be considered as an encoding of the negation $\neg A$. With these things in mind, how can we interpret the content of the expression N? (See also Figure 5.2.)

Problem 13.4 (6.9)

Given $S : *, P : S \to *$ and $f : S \to S$, we define in λC the expression:

$$M \equiv \lambda x : S.\Pi Q : S \to *.(\Pi z : S.(Qz \to Q(fz))) \to Qx.$$

Give a term of type $\Pi a : S.(Ma \to M(fa))$ and a (shortened) derivation proving this.

Problem 13.5 (6.11)

Let $\Gamma \vdash M : N$ in λC and $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$.

(a) Prove that the $x_1, \ldots, x_n$ are distinct.

(b) Prove the Free Variables Lemma (Lemma 6.3.3) for λC.

(c) Prove that $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$, for $1 \leq i \leq n$.

Chapter 14

7-The Calculus of Construction

Problem 14.1 (7.1)

Verify that each of the following expressions is a tautology in constructive logic, (1) by giving a proof in natural deduction, and (2) by giving a corresponding derivation in λC. (For the natural deduction rules concerning $\Rightarrow$, $\bot$ and $\neg$, see Section 7.1.) You may employ the flag style for the derivations, as in the examples given in the present chapter.

(a) $B \Rightarrow (A \Rightarrow B)$,

(b) $\neg A \Rightarrow (A \Rightarrow B)$,

(c) $(A \Rightarrow \neg B) \Rightarrow ((A \Rightarrow B) \Rightarrow \neg A)$,

(d) $\neg(A \Rightarrow B) \Rightarrow \neg B$ (hint: use part (a)).

Problem 14.2 (7.2)

(a) Formulate the double negation law (DN) as an axiom in λC.

(b) Verify that the following expression is a tautology in classical logic, by giving a corresponding flag-style derivation in λC (use DN): $(\neg A \Rightarrow A) \Rightarrow A$.

Problem 14.3 (7.3)

Give λC-derivations proving that the following expressions are tautologies in classical logic (so you may use DN or ET):

(a) $(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$,

(b) $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$.

Problem 14.4 (7.9)

Verify that each of the following expressions is a tautology in constructive logic, (1) by giving a proof in first order natural deduction, and (2) by giving a flag-style derivation in λC:

(a) $\forall_{x\in S}(\neg P(x) \Rightarrow (P(x) \Rightarrow (Q(x) \wedge R(x)))$,

(b) $\forall_{x\in S}(P(x)) \Rightarrow \forall_{y\in S}(P(y) \vee Q(y))$.

Problem 14.5 (7.10)

As Exercise 7.9:

$$\forall_{x\in S}(P(x) \Rightarrow Q(x)) \Rightarrow$$
$$\forall_{y\in S}(P(y) \Rightarrow R(y)) \Rightarrow \forall_{z\in S}(P(z) \Rightarrow (Q(z) \wedge R(z))).$$

Problem 14.6 (7.14)

Let $\Gamma \equiv S : *, P : S \to *, Q : S \to *$. Consider the following λC-expression:

$$M \equiv \lambda u : (\exists x : S.(Px \wedge Qx)).\lambda \alpha : *.\lambda v : (\Pi x : S.(Px \to \alpha)).$$
$$u\alpha(\lambda y : S.\lambda w : (Py \wedge Qy).vy(w(Py)(\lambda s : Py.\lambda t : Qy.s))).$$

(a) Find a type N such that $\Gamma \vdash M : N$.

(b) Which logical tautology is expressed by N and proved by M?

(c) Give a derivation of $\Gamma \vdash M : N$.