Solution Manual

Collected Problem Solutions

Author Information

Amir Faridi

amirfaridi2002@gmail.com

July 15, 2025

Version 1.0

Contents

1 Domain Theoretic Foundations of Functional Programming 2

- 1 Introduction 3
- 2 PCF and its Operational Semantics 4
- 3 The Scott Model of PCF 5
- 4 Computational Adequacy 6
- 5 Milner's Context Lemma 7
- 6 The Full Abstraction Problem 8
- 7 Logical Relations 9

2 Type Theory and Formal Proof 10

Part 1

Domain Theoretic Foundations of Functional Programming

Chapter 1

Introduction

Chapter 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. ($\Gamma \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n$, 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 $\frac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash M(N):\tau}$. 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).

Chapter 3

The Scott Model of PCF

Chapter 4

Computational Adequacy

Chapter 5

Milner’s Context Lemma

Chapter 6

The Full Abstraction Problem

Chapter 7

Logical Relations

Part 2

Type Theory and Formal Proof