Fully Abstract Models of Typed Lambda-Calculi

Robin Milner

June 24, 2025

1 Introduction

A denotational semantic definition of L consists of a semantic domain D of meanings, and a semantic interpretation $\mathcal{A} : L \to D$. We assume that we are mainly interested in the semantics of programs. Denote by $\mathcal{C}[\quad]$ a program context - that is, a program with a hole in it, to be filled by a pharase of some kind.

One desirable property of $\mathcal{A}$ is that for all phrases M and N we have $\mathcal{A}[\![\mathcal{C}[M]]\!] = \mathcal{A}[\![\mathcal{C}[N]]\!]$ whenever $\mathcal{A}[\![M]\!] = \mathcal{A}[\![N]\!]$.

This is not hard to achieve, particularly if $\mathcal{A}$ is given as a homomorphism. But it is unfortunate if for some M and N s.t. $\mathcal{A}[\![M]\!] \neq \mathcal{A}[\![N]\!]$ it nevertheless holds for *all* program contexts that $\mathcal{A}[\![\mathcal{C}[M]]\!] = \mathcal{A}[\![\mathcal{C}[N]]\!]$; it means that $\mathcal{A}$ distinguishes finely among nonprogram phrases.

The reason for describing this situation as 'over-generous' is that it typically arises when there are many objects in D which cannot be realized (i.e., denoted by a phrase). For example, $\mathcal{A}[\![M]\!]$ and $\mathcal{A}[\![N]\!]$ may be functions which only differ at an unrealizable argument, which can never be supplied to the functions in a program context.

So we wish to find D and $\mathcal{A}$ s.t.

$$M \sqsubseteq N \quad \text{iff} \quad \forall \mathcal{C}[\quad].M \sqsubseteq N$$

2 Problems

1. 1: ?

3 References