Higher Order Computability

John Longley & Dag Normann

March 1, 2022

Contents

1 Theory of Computability Models

- $e\downarrow$ 'the value of e is defined'
- $e\uparrow$ 'the value of e is undefined'
- $e = e'$ 'the values of both e and e' are defined and they are equal'
- $e \simeq e'$ 'if either e or e' is defined then so is the other and they are equal'
- $e \succeq e'$ 'if e' is defined then so it e and they are equal'

if e is a mathematical expression possibly involving the variable x, we write $\Lambda x.e$ to mean the ordinary (possibly partial) function f defined by $f(x) \simeq e$

Finite sequences of length n starts from index 0.

1.1 Higher-Order Computability Models

1.1.1 Computability Models

Definition 1.1. A **computability model C** over a set T of **type names** consists of

- an indexed family $|\mathbf{C}| = \{\mathbf{C}(\tau) \mid \tau \in \mathsf{T}\}$ of sets, called the **datatypes** of $\mathbf{C}$
- for each $\sigma, \tau \in \mathsf{T}$, a set $\mathbf{C}[\sigma, \tau]$ of partial functions $f : \mathbf{C}(\sigma) \to \mathbf{C}(\tau)$, called the **operations** of $\mathbf{C}$

We shall use uppercase letters $A, B, C, ...$ to denote **occurrences** of sets within $|\mathbf{C}|$: that is, sets $\mathbf{C}(\tau)$ implicitly tagged with a type name τ. We shall write $\mathbf{C}[A, B]$ for $\mathbf{C}[\sigma, \tau]$ if $A = \mathbf{C}(\sigma)$ and $B = \mathbf{C}(\tau)$

In typical cases of interest, the operations of $\mathbf{C}$ will be 'computable' maps of some kind between datatypes