Chapter 1

Introduction

1.1 Outline

1. Definition of a simple, typed lambda calculus

2. Context and definition of *effects* in programming languages

3. Examples of common effects

   (a) IO
   (b) mutable data, state
   (c) exception
   (d) nondeterminism
   (e) non-termination / non-totality
   (f) reader, writer, output
   (g) continuation

4. Explanation of how effects are handled in ML's style

   (a) call-by-value at runtime
   (b) examples of effects: IO, exceptions
   (c) effects are untyped
   (d) simple exception-handling system (type/catch)

5. Explain why this style is not purely functional

   (a) exposes *side-effects*, which are implicit effects not accessible by the programming language