Principles of Programming Languages

Module M11: Summary

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

April 06, 2022

Table of Contents

Principles of Programming Languages: Summary

- λ-Calculus: Syntax
- Functional Programming Languages
  - Functional: Lisp, Scheme, ML & Haskell
  - Multi-Paradigm: λ in Python & C++
- λ-Calculus: Semantics
- λ-Calculus: Typed
- Type Systems
- Denotational Semantics
  - Definition
  - Relationship with Operational and Axiomatic Semantics
  - Semantics of Imperative Languages

Module 01: Course Information

[1] Module 01: Course Information
[2] Module 02: Syntax of λ Calculus
[3] Module 03: Semantics of λ Calculus
[4] Module 04: Typed λ Calculus
[5] Module 05: λ in Functional Programming Languages
[6] Module 06: λ in C++
[7] Module 07: Type Systems
[8] Module 08: Denotational Semantics
[9] Module 09: Imperative Languages

Refer: Syllabus of Principles of Programming Languages

Module 02: Syntax of λ Calculus

- Relations
- Functions
  - Compositions
  - Currying
- λ Calculus
  - Concept of λ
- λ Syntax
  - λ Expressions
    - Notation
  - Example
    - Simple
    - Composition
    - Boolean
    - Numerals
    - Recursion
    - Curried Functions
    - Higher Order Functions

Module 03: Functional Programming

- Functional Programming
- Lisp
- Scheme
- Haskell
- Python

Module 04: Semantics of λ Calculus

- Free and Bound Variables
- Substitution
- Reduction
  - α-Reduction
  - β-Reduction
  - η-Reduction
  - δ-Reduction
- Order of Evaluation
  - Normal and Applicative Order

Module 05: Typed λ Calculus

- $\Lambda^{\rightarrow}$
  - Type Expression
  - Pre-Expression & Expression
  - Type-checking Rules
    - Examples
- $\Lambda^{\rightarrow}_{rr}$
  - Types
    - Tuple Type
    - Record Type
    - Sum Type
    - Reference Type
    - Array Type
  - Type Expression
  - Pre-Expression
  - Type-checking Rules
    - Derived Rules

Module 06: λ in C++

- Functors
  - Callable Entities
  - Function Pointers
    - Replace Switch / IF
    - Statements
    - Late Binding
    - Virtual Function
    - Callback
    - Issues
  - Basic Functors
    - Elementary Example
    - Examples from STL
- λ in C++
  - λ Expression
  - Closure Object
  - Examples
    - Factorial
    - Fibonacci
    - Pipeline
  - Curry Function
- More on λ in C++

Module 07: Type Systems

- Type Systems
  - Type & Type Error
  - Type Safety
  - Type Checking
  - Type Inference
- Type Inference
  - `add x = 2 + x`
  - `apply (f, x)`
  - Inference Algorithm
    - Unification
- Examples
  - `sum`
  - `length`
  - `append`
  - Homework
- Type Deduction
  - Polymorphism
    - Ad-hoc
    - Parametric
    - Subtype
  - C++11,...

Module 08: Denotational Semantics

- Styles
- Syntax
- Domains
  - Domains
    - Product
    - Sum
  - Rat
- Algebra
  - Nat, Tr
  - String
  - Unit
  - Product Dom
  - Sum Dom
  - Lists
  - Function
  - Arrays
  - Lifted Domain
  - Recursive Function
- Denotational Definitions
  - Binary
  - Calculator

Module 09: Imperative Languages

- Imperative Languages
  - Lifted Domains
- Language + Assignment
- Programs are Functions
- Interactive File Editor
- Dynamically Typed Language (with IO)
- Recursive Definitions
- Language with
  - Contexts
  - Block Structured Language
  - Applicative Language
- Summary