Learning checklist

Part I

Lecture Notes pointers

Lecture 1: Introduction

- ☐ Understand the basic ideas of the *denotational approach* to the semantics of programming languages. [Slides 1–2]
- ☐ Understand the notion of *compositional semantics*. [Slides 3–4]
- ☐ Understand the need for supporting *fixed point operators*. [§1.1]

Lecture 2: Least fixed points

- ☐ Be able to give the definition and examples/non-examples of *partial order*, *cpo*, and *domain*. [§2.1]
- ☐ Be able to give the definition of *lubs* of chains, and use the definition as a proof principle.
  Be able to state and prove the *basic properties of lubs* of chains. [§2.1] [Slide 16]
- ☐ Be able to give the definition and examples/non-examples of *monotone*, *continuous*, and *strict* functions. [§2.1]
- ☐ Be able to give the definition of *least pre-fixed point*, and use the definition as a proof principle.
  Be able to prove that least pre-fixed points are *fixed-points*. [§2.2]
- ☐ Be able to state and prove *Tarski's fixed point theorem*. [§2.2]

Lecture 3: Constructions on domains

- ☐ Be able to give the definition of the *product* of domains, *function* domains, and *flat* domains. [§§3.1–3.3]
- ☐ Be able to give the definition and establish the *continuity* of the various functions (projections, pairings, evaluation, currying, composition, fixed point operator) associated to the above constructions. [§§3.1–3.3] [Slide 37]

Lecture 4: Scott induction[1]

- ☐ Be able to give the definition of the concept of *admissible subset* of a domain. [§4.1]
- ☐ Be able to state, prove the soundness of, and apply *Scott's induction principle*. [§§4.1–4.2]
- ☐ Be able to build *admissible subsets*, justifying the constructions. [§4.3]

[1] You can safely skip pages 32–34 of the lecture notes.

Learning checklist
Part II

Lecture Notes pointers

Lecture 5: PCF

- ☐ Understand the syntax, typing, and operational semantics of *PCF*. [§5.1–5.4]
- ☐ Be able to define *partial recursive functions* in PCF. [§5.3]
- ☐ Be able to give the definition of the notion of *contextual equivalence* in PCF. [§5.5]

Lecture 6: Denotational semantics of PCF

- ☐ Be able to give the definition of the *denotational semantics of PCF*. [§6.1–6.2]
- ☐ Be able to *use* the denotational semantics of PCF to prove contextual equivalence in PCF. [Slide 30]
- ☐ Be able to state the *compositionality properties* of the denotational semantics of PCF. [§6.3]
- ☐ Be able to state and prove the *soundness* of the denotational semantics of PCF. [§6.4] [Slide 27]

Lecture 7: Relating denotational and operational semantics

- ☐ Be able to state the *adequacy* property of the denotational semantics of PCF. [Slide 27]
- ☐ Be able to give the definition of the notion of *contextual preorder* in PCF, and to state and prove its extensionality properties. [§7.3]

Lecture 8: Full abstraction

- ☐ Understand the concept of *full abstraction* in PCF, and the reason for which it fails. [§8.1]
- ☐ Be able to give the definition of the operational and denotational semantics of PCF with parallel or. [§8.2]