Exam of Theoretical Computer Science. Year 2003–2004. A. Pettorossi.

1. **Decidability and Partial Recursive Functions**: • r.e. sets and recursive sets. • Semidecidability of the Halting Problem of the Turing Machine. • Partial Recursive Functions. (PE3: Chapter 3: Section: 1, 2, 3, 4. Chapter 4: Section: 1. All proofs are optional.)

2. **Predicate Calculus and Logic Programming**: • Syntax and semantics: Classical Presentation. • Operational and Denotational Semantics of Definite Logic Programs. (PC&LP: Sections: 1, 2.1, 3, 4, 8, 9.3, 9.4. All proofs are optional.)

3. **Well-founded Recursion Theorem** (WI Section 10.4 + HO) (No proof)

4. **Language IMP**: • Operational Semantics (WI Sections: 2.1–2.4). • Denotational Semantics (WI Section 5.2. Use the functional form, as indicated in HO). • Axiomatic Semantics (WI Sections: 6.1–6.4). • Rule induction (WI Sections: 4.1–4.3). • Soundness of Axiomatic Semantics (WI Section 6.5). • Weakest Preconditions, Expressiveness Theorem, and Relative Completeness Theorem. (WI Sections: 7.2 and 7.4. PE1 Section 8.3). (No proofs)

5. **Language REC**: • Operational Semantics and Denotational Semantics in call-by-value and in call-by-name. • Equivalence of semantics. • The Factorial example in the two languages. (WI Sections 9.1, 9.2, 9.3, Theorem 9.8, Sections 9.5, 9.6, and Theorem 9.15. No proofs)

6. **Complete partial orders**. • Products, function space, curry, apply, lifting, sum, and Cond. • Metalanguage for denotational semantics. • Fixpoint of continuous functions and Kleene Theorem. • Scott fixpoint induction, Park induction, and Bekić Theorem. (HO + WI Sections 8.1, 8.2. Basic definitions and properties of Sections 8.3, 8.4, 10.1, 10.2, 10.3. Proof of Kleene's Theorem. No proofs of all other theorems.)

7. **Higher order languages**: • The Eager language, the Lazy-1 language (see Section 11.7), and Lazy-2 language (see Section 11.10): • Operational Semantics and Denotational Semantics. • The Factorial Example in the three languages. • The β rule and the η rule in Eager, Lazy-1, and Lazy-2. (HO + WI Sections 11.1, 11.2, and 11.3. Section 11.4: only what is necessary for Corollary 11.15. Sections 11.5, 11.6, and 11.7. Section 11.8: only what is necessary for Corollary 11.24. Basic concepts of 11.10. No proofs)
Optional: Fixpoint operators in Eager, Lazy-1, and Lazy-2 (WI: Section 11.9). *Optional*: Adequacy and full-abstraction (WI: Parts of Section 11.10).

8. **Non-determinism and Concurrency**. • Owicki-Gries rules (HO). • Milner's CCS. • Axiomatization of finite state processes. • Hennessy-Milner logic. • Modal μ-calculus. • Local model checking. • Correctness proof of the Alternating Bit Protocol. (HO + WI: Sections 14.1, 14.4–14.8. No proofs).
Optional: Dijkstra's guarded commands (WI: Section 14.2). *Optional*: Bisimulation equivalence and Bisimulation congruence for CCS (WI: Exercise 14.9).

9. **Project**: A Prolog program for the operational semantics of IMP.
Optional: A Prolog program for the operational semantics of REC. *Optional*: A simple local model checker in Prolog. Proof of a modal formula of a simple communication protocol similar to the one indicated in WI Exercise 14.22.

References.

- PE1, PE2, PE3, and PE4: Red books.
- PC&LP: A. Pettorossi and M. Proietti: Predicate Calculus and Logic Programming, Aracne 2002.
- WI: Winskel's book.
- HO: Handouts.