Syllabus for Theoretical Foundations Component of PhD Qualifying Exam I (Theory)

May 1999 (edited 2001)

1 Description

The new Theory exam, Exam I in the Computer Science PhD written qualifier program, has two equally weighted parts, theoretical foundations and algorithmics. Students who entered under the previous system of exams, and who have completed just one of the Foundations or Algorithmics exams, have the option of doing just one of the two parts in half the total allotted time.

This theoretical foundations portion of the exam tests your knowledge of fundamental concepts in automata, computability, and elementary time complexity. The exam is written with the understanding that you had strong automata and computability courses in your undergraduate training or that you have taken the first-year graduate course in Theory of Computing at Indiana University.

The new exam will focus on automata and computability. It will continue to assume some knowledge of complexity. as presented in references [4]. References [3, 1] cover the main material of the exam. Each book covers most of the field. Below we have tried to indicate strong coverage of each topic in these books.

2 Topics

2.1 Automata

1. **The Chomsky hierarchy:** [1, ch8–9]
2. **Regular sets:** [2, ch1],[1, ch2–4],[4, ch1]
   (a) Equivalent descriptions (*e.g.* deterministic and nondeterministic Finite-state automata, Regular expressions)
   (b) Closure properties
   (c) Non-regular languages (Pumping lemma or Myhill-Nerode theorem)
   (d) Minimization
3. Deterministic context-free languages [1, §6.4]
   (a) deterministic push-down automata
   (b) closures/non-closures on boolean operations

4. Context-free languages [2, ch2],[1, ch5–7],[3, ch2],[4, ch2]
    (a) Equivalent descriptions (*e.g.* context-free grammars and push-down automata)
    (b) ambiguity
    (c) Chomsky normal form
    (d) Membership testing in cubic time on a deterministic RAM machine. (Cocke-Younger-Kasami or Earley algorithm)
    (e) Non-context-free languages (Pumping lemma)
    (f) Closure properties
    (g) Decidability of non-closures

2.2 **Computability** [2, ch3]

1. Turing machines [1, ch8],[3, ch4],[4, ch3]
2. Church's thesis [1, ch8],[3, ch5],[4, ch3]
3. Decidable (recursive) languages [1, ch8],[3, ch4],[4, ch3]
4. Recognizable (recursively enumerable) languages [1, ch8],[3, ch4],[4, ch3]
5. Unrecognizable languages, Halting problem, Unsolvability via a diagonal argument [1, ch9],[3, ch5],[4, ch4]
6. Reduction to prove other problems unsolvable [1, §8.1],[3, ch5],[4, ch5]

2.3 **Elementary Complexity** [1, ch10],[4, ch7]

1. The class P [3, ch6]
2. Satisfiability (SAT problem) [3, ch6]
3. The class NP [3, ch7]
4. NP-completeness [3, ch7]
5. Reduction within P and NP [3, ch6]
6. Log-space complexity [1, §11.2.3], [4, §8.1,8.4]

References

[1] John E. Hopcroft, Rajeev Montwani, and Jeffery D. Ullman. *Introduction to Automata Theory, Languages, and Computation,* 2nd ed. Boston: Addison–Wesley (2001).

[2] Dexter C. Kozen. *Automata and Computability.* New York: Springer (1997).

[3] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation,* 2nd ed. Upper Saddle River, NJ: Prentice–Hall (1998).

[4] Michael Sipser. *Introduction to the Theory of Computation.* Boston: PWS Publishing (1997).