Syllabus for Part I of the Ph.D. Qualifying Exam in Computer Science Revised (12/5/13)

Part I of the Ph.D. qualifying exam is for four hours and consists of three areas: Hardware, Software, and Theory. There will be a total of 12 questions, four questions from each area. A student is required to answer any nine questions with at least two questions from each area. The exam is closed-book and closed-notes. The use of electronic devices is not allowed.

This document describes the list of major topics from which the questions for each area will be chosen. Each major topic is accompanied by (i) suggested text(s) and (ii) a Computer Science course that covers the topic in sufficient depth.
Hardware

The hardware portion of the qualifying exam covers the topics of digital logic, computer organization, and computer architecture.

Digital Logic

Topics: binary number system, boolean algebra, logic circuit minimization, combinational circuits design and analysis, synchronous sequential circuits design and analysis, registers, counters, memory, programmable logic arrays, and hardware description languages Verilog or VHDL.


Course: CompSci 315 Intro to Computer Organization and Assembly Language

Computer Architecture

Topics: register transfer language, basic computer organization and design, instruction set design, control unit design, arithmetic and logical (ALU) design, pipelining and vector processing, input/output organization, memory organization.


Course: CompSci 458 Computer Architecture
Software

The software portion of the qualifying exam covers programming data structures and algorithms and also covers operating systems. For both of these areas, we give a list of topics from which the qualifying examination question for the area will be chosen. We also list a suggested textbook that covers the topics in sufficient depth. Each area also is addressed by a course that covers the topics as well.

Programming Data Structures and Algorithms

Topics: objects, recursion, exceptions, container ADTs, dynamic arrays, iterators, linked lists, generic classes, stacks, queues, binary search trees, graphs (depth-first and breadth-first search), hashing, sorting (insertion sort, merge sort).


Course: CompSci 351: Data Structures and Algorithms

Operating Systems

Topics: process management (processes, threads, scheduling, context switching, process creation and termination, critical sections, deadlocks), memory management (memory hierarchy, address translation, memory-related services, memory layouts, memory allocation, overallocation).


Course: CompSci 537: Introduction to Operating Systems
Theory

The theory portion of the exams covers the topics of discrete mathematics, automata and formal languages, and algorithm design and analysis.

Discrete Mathematics
Topics: logic, sets, functions, proofs, mathematical induction, counting, binomial coefficients and identities, discrete probability, basic graph theory.


Course: CompSci 317: Discrete Information Structures

Algorithm Design and Analysis
Topics: asymptotic notation, solving recurrence relations, stacks, queues, vectors, lists, trees, priority queues and heaps, hashing, binary search trees (including AVL trees and red black trees), sorting and selection, structures/algorithms for disjoint sets, algorithm design techniques (the greedy method, divide and conquer, recursion, dynamic programming), graph algorithms (including minimum spanning tree algorithms and shortest path algorithms).

Suggested Texts:

Course: CompSci 535: Algorithm Design and Analysis