Spring 2007

University of Wisconsin-Milwaukee
Department of Electrical Engineering and Computer Science

Guidelines for Ph.D. Qualifying Exam
Computer Science

The Ph.D. qualifying exam is composed of two half-day written exams, each lasting four hours. The exam is closed books and closed notes.

Day 1: Theory and Hardware.

There are two main areas for examination on day 1: Theory and Hardware. Four questions per area is given; a student is expected to answer any three of them. Thus, a student is expected to answer a total of six questions.

Day 2: Software and Systems.

There are four areas for examination on day 2: Artificial Intelligence; Computer Networks and Operating Systems; Software Engineering; Programming Languages and Compilers. Two questions from each area is given. A student is expected to answer any six of these eight questions.

If you have any questions about the exam or about the material you should know, please discuss them with your advisor. Exams from previous semesters are available at the department's office.

Theory

Following is a list of topics to be covered in three main areas, as well as suggested reading. The reading lists are based mainly upon texts that have been recently used by instructors. There are several other texts available to cover these topics that a student may prefer to consult. It may be that not all of the topics may be found in any single one textbook.

Discrete Mathematics

Topics include: logic, sets, functions and relations, modular arithmetic, mathematical induction, combinatorics, graphs, discrete probability, recurrence relations.
[Most of these topics are usually covered in CS 317 (Discrete Information Structures).]

Suggested text:
K. Rosen, *Discrete Mathematics and its Applications*, 6th ed., McGraw-Hill, Chapters 1-9.

Automata and Formal Languages

Topics include: finite automata, pushdown automata, Turing machines and variants, regular languages, context-free languages, context-sensitive languages, recursive languages, recursively enumerable languages, regular expressions, various classes of grammars and normal forms, Chomsky hierarchy, reducibility, decidability.
[Most of these topics are usually covered in CS 417 (Introduction to the Theory of Computation).]

Suggested texts:
M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, Chapters 1-5.
J. Hopcroft, R. Motwani and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed., Addison-Wesley, Chapters 1-9.
P. Linz, *An Introduction to Formal Languages and Automata*, 4th ed., Jones and Bartlett, Chapters 1-13.

Data Structures and Algorithms

Topics include: asymptotic notation, recurrence relations, lists, stacks, queues, trees (including binary search trees), priority queues, heaps, hashing, sorting, structures/algorithms for disjoint sets, graph algorithms, algorithm design techniques (greedy algorithms, dynamic programming, recursion, divide-and-conquer, etc.), selection, network flows, worst-case and amortized analysis, NP-completeness, approximation algorithms, randomized algorithms.
[Most of these topics are usually covered in CS 535 (Data Structures and Algorithms) and CS 704 (Analysis of Algorithms).]

Suggested texts:
M. A. Weiss, *Data Structures and Algorithm Analysis*, Benjamin/Cummings, Chapters 1-10.
T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms*, 2nd ed., McGraw-Hill, Chapters 1-26, 34, 35.

Hardware

Following is a list of topics to be covered in three main areas, as well as suggested reading. The reading lists are based mainly upon texts that have been recently used by instructors. There are several other texts available to cover these topics that a student may prefer to consult.

Digital Logic

Topics include: Binary number systems, Boolean Algebra, Logic circuit minimization, Combinational circuits design and analysis, Synchronous Sequential circuits design and analysis, Registers, Counters, Memory, and Programmable Logic Arrays.
[Most of these topics are usually covered in Digital Logic Course EE 354.]

Suggested text:
Morris Mano, *Digital Design*, 3rd ed., Prentice Hall, Chapters 1-7.

Computer Architecture

Topics include: Data representation, Register transfer language (Hardware Description Language Basic computer organization, Assembly language programming, Microprogrammed control, Central processing unit, Pipelining and vector processing, Computer arithmetic and ALU, Input/Output organization, Memory organization, Parallel computer architecture.
[Most of these topics are usually covered in Computer Architecture Course CS 458.]

Suggested texts:
M. Morris Mano, *Computer Systems Architecture*, Prentice Hall, Chapters 3-13.
William Stallings, *Computer Organization and Architecture*, 5th ed., Prentice Hall, Chapters 1-16.

Switching Theory

Topics include: Boolean Algebra, Shannon's expansion theorems and canonical expressions, Minimal and reducible sum of products, Symmetrical and Threshold functions, Fault detection in combinational logic circuits; Sequential circuits and machines including linear sequential circuits.
[Most of these topics are usually covered in Switching Theory and Automata Theory Course CS 751.]

Suggested text:
Zvi Kohavi, *Introduction to Switching and Automata Theory*, McGraw-Hill, Chapters 6-10.

Artificial Intelligence

The following is a list of topics to be covered: state space search, heuristic (informed) search, adversarial search, knowledge representation, propositional logic, first-order logic, deductive inference, resolution, planning, natural language, phrase structure grammars, network grammars, chart parsing, and programming in Lisp.

Material from the following chapters of the books below forms the syllabus of the AI part of the examination.

[Most of these topics are usually covered in CS 422 (Introduction to Artificial Intelligence) and CS 710 (Artificial Intelligence). Some of these topics are also covered in CS 722 (Artificial Intelligence Planning Techniques) and in CS 423 (Introduction to Natural Language Processing) and CS 723 (Natural Language Processing).]

S. Russell and P. Norvig, *Artificial Intelligence-A Modern Approach*, Prentice Hall, 2nd ed., 2003.
Paul Graham, ANSI Common Lisp, Prentice Hall, 1996, ISBN 0-13-370875-6.

Search

Russell and Norvig - Chapters 3, 4, and 6.

Knowledge Representation and Deduction

Russell and Norvig - Chapters 7,8 and 9.

Planning

Russell and Norvig - Chapter 11.

Learning

Russell and Norvig – Chapter 18.

Natural Language

Russell and Norvig - Chapter 22.

Lisp

Graham - Chapters 1,2,3,4,5,6,7,8 and 9.

Computer Networks and Operating Systems

Following is a list of topics to be covered in two main areas, as well as suggested reading. The reading lists are based mainly upon texts that have been recently used by instructors. There are several other texts available to cover these topics that a student may prefer to consult. It may be, that not all of the topics may be found in any single one textbook.

Computer Networks

Topics include: Applications and Layered Architectures, Data Communications, SONET, Cellular Wireless Networks, Peer-to-Peer Protocols & Data Link Layer, Local Area Networks & Wireless LAN, Packet Switching, Network Layer , Transport Layer, TCP/IP & Mobile IP, ATM Networks & Wireless ATM, Network Security, Multimedia Information and Networking, Network Management.
[Most of these topics are usually covered in Computer Networks Course CS 520.]

Suggested text:

Leon-Garcia and I. Widjaja, *Communication Networks Fundamental Concepts and Key Architectures* McGraw-Hill, Chapters 1-12.

Operating Systems

Topics include: Description of General Operating System Concepts, Disk Management Device Driver/Disk Controller Interaction, Scheduling Algorithms, RAIDs, Processes, Components of a Process, Implementing Processes, Threads: Lightweight Processes, Using Threads, CPU Scheduling, Process State Model, Scheduling Algorithms, Concurrency, Interprocess Communication, Named Pipe, Message Queue, Shared memory, Synchronization/Mutual exclusion, Spinlock, Atomic Instructions, Semaphore, Monitor, Deadlock, Conditions for deadlock, Solutions to Deadlock, Memory Management, Contiguous Allocation, Segmentation, Paging, Virtual Memory, Data Structure, Page Replacement Policies, File System, Components of a File, Disk Allocation/File Index, Directory Structure, Security and Protection, Distributed Computing, Message Passing, RPC, Distributed Memory Management.
[Most of these topics are usually covered in Operating Systems Course CS 537.]

Suggested text:

J. Nutt, *Operating Systems*, 2nd ed., Addison-Wesley , Chapters 1-18.

Software Engineering

Following is a list of topics to be covered: software and its nature; software quality; the software production process; principles of software engineering; software design, specification, and verification; tools and environments; software metrics; and software performance.
[Most of these topics are usually covered in CS 536 (Introduction to software Engineering).]

Suggested text:

C. Ghezzi, M. Jazayeri, D. Mandrioli, *Fundamentals of Software Engineering*, 1st ed., Prentice Hall Chapters 1-3, 5-7.

Programming Languages and Compilers

Following is a list of topics to be covered as well as suggested reading. The reading lists are based mainly upon texts that have been recently used by instructors. There are several other texts available to cover these topics that a student may prefer to consult.

Programining Languages

Detailed knowledge of particular languages is not tested, but students are assumed to be familiar with the concepts of object-oriented programming (as in Java), functional programming (as in ML), and logic programming (as in Prolog). Topics include: syntax (BNF and EBNF grammars, syntax diagrams, parse trees, precedence and associativity, ambiguity, phrase structure, lexical structure), language systems (compilers, assemblers, linkers, loaders, interpreters, intermediate-code compilation, delayed linking, dynamic compilation, binding times), types (static typing, dynamic typing, type inference, polymorphism), scoping (block scoping, namespaces, dynamic scoping), activation records (static allocation, runtime stacks, nested functions, garbage collection implications), memory management (stacks, heaps, fragmentation, compaction, garbage collection), object-oriented constructs (classes, prototypes, inheritance, encapsulation, polymorphism), exception handling, parameter-passing mechanisms (by value, by result, by value-result, by reference, macro expansion, by name, by need), logic programming concepts (unification, resolution, proof trees, backtracking), cost models (cons-cell lists, tail calls), natural semantics.
[Most of these topics are usually covered in CS 431 (Programming Languages Cocepts).]

Suggested text:

Adam Webber, *Modern Programming Languages: A Practical Introduction*, Franklin, Beedle and Associates, 2002

Compilers

Topics include: lexical analysis, syntactic analysis, semantic analysis, intermediate code generation, target code generation, compiler writing tools (lex, yacc, attribute grammars), top-down parsing, LR parsing, binding, scope, environment, overloading, name spaces, symbol tables, type checking, evaluation order, dispatch tables, calling sequences, stack frames, temporaries, register allocation, pipeline scheduling, optimization, basic blocks, data flow analysis, redundancy elimination, assembling, linking, (shared) libraries.
[Most of these topics are usually covered in CS 654 (Introduction to Compilers).]

Suggested text:

M. L. Scott, *Programming Language Pragmatics*, Morgan Kaufmann, 2000, Chapters 1-10, 13.