Curriculum Committee
Minutes

January 22, 2007

Attendees: Tom Doeppner, Chad Jenkins, Franco Preparata, Steve Reiss, John Savage (Chair)

Absent: Claire Kenyon

Guest: Philip Klein

1. Approval of Minutes
   The minutes of December 22 were approved.

2. Next Meeting
   We agreed that our meetings will be on Monday afternoons at 3pm, although Steve not be available on Monday, January 29.

3. CS019 Programming with Data Structures and Algorithms
   The committee discussed and approved a new course, CS019 Programming with Data Structures and Algorithms, submitted by Steve Reiss.
   The course is designed to be a one-semester version of CS015 and CS016 for students with prior programming background. It would have as prerequisites CS004 OR a score of 4 or 5 on the CS AP test OR permission of instructor.
   Steve expects matriculating students to be able to write a 100-line program.
   The course will cover data structures including stacks, lists, queues, trees, heaps, and graphs; algorithmic methods such as divide and conquer and dynamic programming; and basic analysis techniques. The course also covers object-oriented programming in Java, basic object-oriented design, and software development techniques.
The course will have six two-week programming assignments. Each will require the use of one (or more) of the data structures and algorithms that are being taught. The emphasis will be on using the built-in data structure library of Java rather than implementing data structures from scratch. Each assignment will also illustrate some aspect of programming such as interfaces and packages, user interfaces, design patterns, or testing. For each assignment, each student will have a design review. For all but the last assignment, students will present their code in a code walkthrough. There will also be a final exam and a midterm.

For students who have a programming background in something other than Java, there will be a series of evening lectures the first week that covers basic Java syntax and the relationship of their language (C, Matlab, C++) to Java. Two books will be used, one a Java reference that is readable (different ones might be appropriate based on the student’s background) and one a data structures text.

The topics to be covered are the following. Steve discussed the algorithmic topics with Claire, who teaches CS157, and she approved of them.

- Collections: Arrays, Lists, Stacks, Queues
- Searching: Maps: hashing, trees
- Sorting: Quicksort, mergesort, topological sort
- Graphs: representations, shortest path, depth-first and breadth-first search
- Dynamic programming
- Divide and conquer
- Asymptotic analysis (big-O; NP-completeness)
- Other data structures as appropriate to the assignments
- Heaps and priority queues
- Object-oriented design
  - Interfaces    Callbacks
  - Recursion    User-interface design and swing
  - Testing      Design patterns
  - Code style   Exceptions
  - Programming tools Simple Multithreading (swing-based)

In order to keep the faculty informed of important instructional initiatives, Steve will present the course at a faculty meeting.
4. CS295 Computational Models of the Neor cortex

Tom Dean plans to teach a CS295 special topics course with the title *Computational Models of the Neor cortex* and the following abstract.

This course addresses the problem of modeling the primate perceptual neocortex using probabilistic graphical models, including Bayesian and Markov networks, and extensions to model time and change such as hidden Markov models and dynamic Bayesian networks. The emphasis is on problems of learning, inference, and attention. Sources include the literature in computational and cognitive neuroscience, machine learning, and other fields that bear on how biological and engineered systems make sense of the world. Prerequisites: basic probability theory, algorithms, and statistics.

The committee approves the course but encourages Tom to seek approval for a regular course if he expects to teach it more than once. This might be a special topics course in this area, if the content will change with time, or a regular course, if it will not. Our approval is not contingent on his seeking approval of a regular course but we do strongly encourage him to consider such an action.

5. CS25/26 Pathways: Some Mathematical Methods for Computer Science

Philip described a proposed new two-semester course (called CS23/24 in previous minutes) developed with Tom Dean that would present essential material from CS22 and CS51 but would add additional mathematical material. It is their intention to teach this course on an experimental basis to an audience second-year or older students (although advanced first-year students would be allowed in) starting this fall. A draft catalogue description follows.

An application-oriented introduction to some of the mathematical methods arising in computer science. The goals of the course include: developing facility with reading and writing proofs; becoming familiar with the use of graphs, discrete probability theory, and linear algebra in modeling; understanding the theoretical limits of computation; becoming acquainted with some computational methods such as linear programming. This two-course sequence satisfies the CS22 requirement and the Computer Science concentration’s introductory theory requirement.

Philip and Tom have done a comparative assessment of the topics in CS22 and CS51 that they would include and exclude, which is shown below.
**CS22 topics included in CS25-26**
- Propositional logic
- Predicate logic
- Proof techniques, including proof by induction
- Sets and set theory, Cartesian products, relations
- Graphs, paths, trees
- Functions, cardinality
- Discrete probability theory: sample spaces, random variables, discrete distributions, moments, conditional probability, Bayes’ theorem, independence, tail bounds (Chebychev, Chernoff)

**CS22 topics not included in CS25-26**
- Recurrence relations, counting
- Continuous probability theory: Poisson distribution, normal distribution, central limit theorem

**CS51 topics included in CS25-26**
- Deterministic and nondeterministic finite automata, and their relation
- Regular expressions
- Connection between regular expressions and finite automata
- Deterministic and nondeterministic Turing machines
- Decidability
- Reducibility
- The classes P and NP
- NP-completeness and NP-complete problems

**CS51 topics not included in CS25-26**
- Pumping lemma (though we will prove a particular language is not regular)
- Push-down automata
- context-free languages
- CKY parsing algorithm for CFGs

**MA52 topics included in CS25-26**
- Matrices and matrix arithmetic
- Vector spaces
- Linear independence and span
- Dimension
- Projections
- Eigenvalues and eigenvectors

- Triangular and symmetric matrices
- Systems of linear equations
- Bases
- Inner products
- Least-squares approximation
MA52 topics not included in CS25-26
rotations Gaussian elimination
determinants cofactors
Cramer’s Rule Cauchy-Schwartz inequality
Gram-Schmidt diagonalization
kernel, image, isomorphism linear transformations

CS25-26 Topics that don’t fit into the above existing courses
Local search (deterministic and probabilistic)
Linear programming
Principal component analysis
Probabilistic automata
Connection between Matrices and Graphs
Markov Chains
Bayesian networks

Philip and Tom have prepared the following syllabus outline based on the above.

(a) Propositional logic, Predicate logic, and proof techniques, including proof by induction
(b) Sets and set theory, Cartesian products, relations, functions, cardinality
(c) Graphs and relation to relations (e.g. dags), paths, trees
(d) Discrete probability theory: sample spaces, random variables, discrete distributions,
(e) Deterministic and nondeterministic finite automata, and their relation
(f) Regular expressions, and their relation to finite automata
(g) Deterministic and nondeterministic Turing machines
(h) Decidability, and reducibility
(i) The classes P and NP
(j) NP-completeness and NP-complete problems
(k) matrices and matrix arithmetic
(l) Connection between matrices and graphs (including triangular and symmetric matrices)
(m) Markov Chains
(n) vector spaces
(o) systems of linear equations
(p) linear independence and span, bases, dimension, inner products, projections
(q) least-squares approximation
(r) eigenvalues and eigenvectors
(s) Principal component analysis
(t) Local search (deterministic and probabilistic)
(u) Linear programming
(v) Probabilistic automata
(w) Bayesian networks

In response to questions, Philip indicated that the material would be taught on the basis of themes, these being a) probability, b) computation, c) matrices, d) proofs, and e) graphs. A theme is the recurrent presentation of a topic in multiple contexts. The goal is to establish multiple connections between a topic and its applications.

Philip agrees that it is ambitious to teach the material proposed for this course. He understands that some of it might have to be forsaken so that the material that is presented is presented well.

Philip reported that other time constraints preclude spending a lot of time preparing on the teaching of this class. That is, he and Tom cannot spend the summer in preparation.

Out of fairness to students committee members strongly urged Philip and Tom to produce at least a rough syllabus before teaching the course. Committee would also like to see such a syllabus. Philip said that he would provide such a syllabus. Given the nature of the experiment, Philip agreed that the course should be offered to second-year or later students.

We discussed whether the course should be taught as a spring-fall course or a fall-spring course. Philip strongly favors the latter while many committee members favor the former. We decided that teaching an experimental course as a fall-spring course does not bind the course to being taught that way forever should it become a regular course.

The meeting concluded at 4pm without a final resolution of this matter.