

SYLLABIS

TEXT BOOKS:

- I) "Introduction to Languages and the Theory of Computation,"
by John C. Martin, 3rd edition,
Parts IV, V and VI .
- II) "Introduction to Algorithms",
by Thomas H. Cormen,
Charles E. Leiserson,
Ronald L. Rivest,
Chapters 36 and 37.

I) is text book for CIS 130 and II) for CIS 101. The table of contents of those parts of the books constitutes a detailed syllabus for our course.

OTHER GOOD BOOKS:

- "Formal Languages, Automata, and Complexity," by Glenn Brookshear.
"Automata and Computability," by Dexter C. Kozen.
"Elements of the Theory of Computations," by Harry R. Lewis and Christos H. Papadimitriou.
"The Theory of Computation," by Bernard M. Moret.

MORE ADVANCED BOOKS THAN OURS:

- "The Language of Machines", by Robert W. Floyd and Richard Biegel
"Models of Computation", by John E. Savage "Computability Theory", by S. Barry Cooper
"Introduction to Automata Theory, Languages, and Computation", by John E. Hopcroft and
Jeffrey D. Ullman.

A BOOK ABOUT THE PHILOSOPHICAL IMPLICATION OF THE MATERIAL OF THE CLASS

- "Computability - Computable Functions, Logic, and the Foundation of Mathematics",
by Richard L. Epstein and Walter A. Carnielli.
"Machines and Thought," edited by Peter Milican and Andy Clark.
"Games of Life" by Karl Sigmund (seems to be out of print).

The above and related texts of various levels will be put on reserve in the Science Library (if there is serious interest).

The main portion of the class will be spent on Turing Machines (ca 4 weeks) and on determining the class of functions they can compute:

Turing's Thesis:

"The computational power of Turing Machines is as great as the power of any possible computational device."

Church's Thesis:

"The class of 'partial recursive' functions is the class of all (partial) computable functions."

We will also give problems and functions that are provable not computable by Turing machines. Modulo the above these problems and functions are not computable by any computational device. In the second half of the course we will mainly study the time complexity of computational problems.