A Guide to Understanding **'COMPUTATION THEORY'**

Prerequisites:	Part IA courses	**Discrete Mathematics**
		Regular languages and Finite Automata
	Diploma course	**Mathematics for Computation Theory**

This course is about the computational power of various classes of abstract machine. The classes we consider all operate sequentially, apply a finite program to a finite input data structure, and compute with potentially unlimited memory until a data dependent stopping rule is triggered: if the program terminates the result is presented as a finite output data structure. It turns out that within these rather wide constraints the various models that we consider possess equivalent power, and that the class of such computations can be presented in terms of the mathematical theory of recursive functions. The aim of the course is to answer the question "**can** this problem be solved on a machine?".

The course splits naturally into three sections. The first − really up to and including the proof of the Unsolvability of the Halting Problem − is intended to develop intuition about what can and what can't be done within a particular model of computation, the *register machine* as introduced by Minsky and John Conway. The middle section shows that the results are more general than might have been supposed: register machines are proved equivalent to an earlier "mechanical" model, the *Turing machine*, and canonical forms are derived for both; we next show that these models are also equivalent to Church's *recursive functions*. The final section of the course presents some rather more difficult results, essentially exploring *semi-recursive* phenomena − decision problems that cannot be resolved, but for which an answer can be checked. Progress through the pages of the lecture notes is linear to a first approximation, so the first section should take about 5 lectures, the second 4, and the final somewhat hairy mathematical section only 3.

Most of the standard textbooks listed below include exercises, and a number are also given at various points in the lecture notes. The course syllabus has been stable over a considerable period, so all of the questions in the Tripos papers available through the Computer Laboratory Web site are directly relevant (with the exception of 1995, when the treatment was rather more mathematical). If you want to understand the course you **must** tackle some problems: if you want to have fun as well, then designing a few programs for register or Turing machines would actually be a useful way to spend your time.

A. Course Reading

Because there are many different formalisms it is possible to treat the subject in many different ways. At one extreme, the subject is part of mathematics: Recursive Function Theory, a branch of Mathematical Logic. This must not be forgotten, and the central results (such as the *Unsolvability of the Halting Problem*) are theorems. Some books are so rigorous that the computational intuition is lost. Others eschew precise arguments to such an extent that no sceptic could be convinced by their claims. I believe that the right approach for a serious course in a Computer Science department lies in between these two extremes, and all the books below occupy this middle ground.

To my way of thinking none of these books is as good as Brookshear, see section D. The book by Hopcroft and Ullman was one of several recommended for the Michaelmas Term course "Mathematics for Computation Theory" given to Part 2 General and Diploma students: it covers the ground for the present course on Computation Theory also, and I've nothing against it (incidentally, a new edition is promised for 2001, but I've not yet seen any clear evidence of it). However, as a single text to cover both of the courses I believe that Davis, Sigal and Weyuker is better, and the book by Sudkamp is at least as good. If you can find a copy Rayward−Smith is good value, though the treatment is based on Turing machines, which gives it a different flavour from the lecture notes (a good thing?).

The recent book by Neil Jones looks good to me, but it isn't really suitable for first course reading except by the most confident, since it assumes a certain amount of sophistication about the meaning of programs. It steers a skilful middle course between theory and practice, relating theoretical results to practical techniques such as partial evaluation. It is also relevant to the course on Complexity (how long **must** it take to solve this problem?).

The first four books given below also cover the course on Complexity, and all in addition contain useful material about Regular Algebra.

* M D Davis, R Sigal and E J Weyuker, **"Computability, Complexity and Languages"**, Academic Press (2nd edition 1994), £43.95. ISBN 012206382-1
* J E Hopcroft and J D Ullman, **"Introduction to Automata Theory, Languages and Computation"**, Addison−Wesley (1979), £32.73. ISBN 020102988-X
* Thomas A Sudkamp, **"Languages and Machines"** (especially chapters 9, 11−13), Addison−Wesley (2nd edition 1995), £38.99. ISBN 020182136-2
* Neil D Jones, **"Computability and Complexity from a Programming Perspective"**, MIT Press (1997), £41.50. ISBN 026210064-9

V J Rayward−Smith, **"A First Course on Computability"**, McGraw-Hill (1995?), £21.99. ISBN 063201307-9. (may no longer be in print)

Prices are obtained via the Internet and may not be representative

B. Logic and Formal Arithmetic

These two books are included for background, and both cover a lot of difficult material in a very approachable way. The collection edited by Hintikka includes a reprint of a 1957 article by Hartley Rogers that is the best explanation I know of what the course is all about. There is an article by Abraham Robinson about non−standard analysis as well as important papers by Post, Smullyan and Kreisel. I've included it here rather than in section D because it offers a very specific perspective on the course. Crossley's book is short and readable, and offers a uniform presentation of many of the key ideas of mathematical logic. At the Dover reprint price this book is the great bargain in the list!

Gödel's Incompleteness Theorem is particularly relevant, not only because the result has implications for computer systems that perform integer arithmetic, but more importantly because the proof technique developed by Gödel is directly analogous to that required to show that the Halting Problem is undecidable (see the paper by Smullyan).

J Hintikka (ed.), **"The Philosophy of Mathematics"**, Oxford University Press (1969), out of print.

* J N Crossley, **"What is Mathematical Logic?"** (1972), Dover reprint, £4.76. ISBN 048626404-1

C. Background and Recreational Reading

There are lots of good books that give the flavour of the course, with a great variety in the way of mathematical prerequisites. Those that require very little include Roger Penrose's **"The Emperor's New Mind"** and various popular books by Raymond Smullyan (**"What is the name of this book?"** and **"The Lady or the Tiger"** are good fun but have a serious purpose − **"Forever Undecided"** explores the use of diagonal arguments and so is closely related to the course, whereas **"To mock a mockingbird"** is about combinatory logic as a basis for computation, so highly technical but a bit on the fringe). Galloway & Porter seem to turn up copies of the OUP edition of Smullyan's books from time to time in their sales, and it is worth looking out for them (not so long ago I bought a copy of "To mock a mockingbird" for £1). The works of Lewis Carroll are perhaps not directly relevant, but they share a certain attitude of mind.

Turing's World is a fancy Turing Machine simulator that runs on a Mac (and might have run on a PC, but there were technical difficulties and the authors eventually lost interest). There are quite a few interesting predefined machines, and it is easy to add new ones that you would like to explore (I built a machine for Ackermann's Function, for example). The textbook that accompanies the disc explains what is going on in a very coherent way.

David Hilbert in 1902 proposed a set of important problems that would challenge future mathematicians, and by and large they've given the world a run for its money. The Tenth Problem finally yielded in 1968 to a concerted attack, and the article by Martin Davis in the Scientific American explains exactly how the undecidability of the Halting Problem played a key role in the solution. There is a heavier treatment of the same result in the Springer graduate text by Barnes and Mack, 'An Algebraic Introduction to Mathematical Logic'. Possibly the nicest treatment of all is in Keith Devlin's book, which contains a professional but comprehensible introduction to a number of the many recent major advances in mathematics. It also includes a general section on incompleteness theorems and undecidability, as well as a discussion of complexity and NP−completeness.

The first reference below is in a class of its own. Harel presents a coherent high−level view of the theory of computer science, short on mathematical detail but full of all the right intuitions − it gives the background for the course on Complexity as well, and is generally a good thing.

* David Harel, **"Algorithmics − the spirit of computing"** (2nd edition 1992), Addison−Wesley, £31.99 (especially chapters 8 and 9). ISBN 020150401-4

* Jon Barwise and Jon Etchemendy, **"Turing's World 3.0 − an Introduction to Computability Theory"** (1993).
(Mac) CSLI Lecture Notes No 35, £17.95. ISBN 188152610-0

* Keith Devlin, **"Mathematics: the New Golden Age"** (2nd edition, 1998), Penguin £7.99. ISBN 014025865-5

Raymond Smullyan, **"What is the name of this book?"** Penguin Paperbacks (1988), £7.19. ISBN 014013511-1

Raymond Smullyan, **"The Lady or the Tiger? ; and Other Logical Puzzles"** Penguin Paperbacks (1983), £2.50. ISBN 014022478-5

* Raymond Smullyan, **"Forever Undecided − a puzzle guide to Gödel"** (1987), Oxford Paperbacks (2000), £7.99. ISBN 019280141-4

Raymond Smullyan, **"To mock a mockingbird"** (1985), Oxford Paperbacks (2000), £7.99. ISBN 019280142-2

M Davis, **"Hilbert's Tenth Problem"**, Scientific American, (November 1973).

D. Golden Oldies!

The books by Bird and Minsky are both out of print, but should be available in libraries: both have a lot going for them. Bird covers the elements of the theory of complete partial orders as well, not relevant to this course but nice to have: the basic model of computation is the register machine, which matches the notes well. Minsky is the nearest to splashy of any of the books recommended, but the feeling of the subject comes across really well, and there are lots of fun examples (yes, you **can** program a Turing machine!).

Formal Languages are closely related to their mechanical recognisers, and Kleene's Theorem about the power of finite−state machines is just one of many important results that link languages to automata. Kurki−Suonio is the only book I know (other than Markov's, which is unapproachable) to tie in Markov Algorithms, which formed the theoretical basis for a number of practical programming languages in the 1960's, including SNOBOL.

Brookshear is quite recent, and is strongly motivated by relating the content of the course to practical applications in computing. It covers the inversion of partial recursive functions more clearly than any other text. It was originally issued as a student text (hardback for the price of paper) at about £23. In 1999 it had risen to £48.99, and not surprisingly there were few takers. If you can find a copy at a reasonable price it can be strongly recommended. There are no plans to reissue it, as far as I know.

Hermes does the mathematics beautifully: there is good coverage of Ackermann's function (illuminating); the basic model is the Turing machine, with precise detail that never dominates. This is a classic.

R Bird, **"Programs and Machines"**, Wiley (1976), out of print.

M Minsky, **"Computation: Finite and Infinite Machines"**, Prentice−Hall (1967), out of print.

R Kurki−Suonio, **"Computability and Formal Languages"**, Auerbach (1971), out of print.

J Glenn Brookshear, **"Theory of Computation: Languages, Automata and Complexity"**, Benjamin−Cummings (1989). ISBN 080530143-7 out of print.

H Hermes, **"Enumerability, Decidability, Computability"**, Springer (1965), out of print.

J K M Moody

January 2001