Subrata Dasgupta

COMPUTER SCIENCE

A Very Short Introduction

OXFORD

Subrata Dasgupta

COMPUTER SCIENCE

A Very Short Introduction

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP,
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries

First edition published in 2016

Impression: 1

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data
Data available

Library of Congress Control Number: 2015950971

ISBN 978–0–19–873346–1

Printed in Great Britain by
Ashford Colour Press Ltd, Gosport, Hampshire

Contents

Preface

The 1960s were tumultuous times, socially and culturally. But tucked away amidst the folds of the Cold War, civil rights activism, anti-war demonstrations, the feminist movement, student revolt, flower-power, sit-ins, and left-radical insurrections—almost unnoticed—a new science came into being on university campuses in the West and even, albeit more tentatively, in some regions of the non-Western world.

This science was centred on a new kind of machine: the electronic digital computer. The technology surrounding this machine was called by a variety of names, most commonly, ‘automatic computation’, ‘automatic computing’, or ‘information processing’. In the English-speaking world this science was most widely called *computer science*, while in Europe it came to be labelled ‘informatique’, or ‘informatik’.

The *technological idea* of automatic computation—designing and building real machines that would compute with minimal human intervention—can at least be traced back to the obsessive dreams of the English mathematician and intellectual gadfly Charles Babbage in the early 19th century, if not further back. The *mathematical concept* of computing was first studied in the late 1930s by the logicians Alan Turing in England and Alonzo Church in the United States. But the impetus for a proper *empirical* *science* of computing had to wait until the invention, design, and implementation of the electronic digital computer in the 1940s, just after the end of the Second World War. Even then, there was a gestation period. An autonomous science with a name and an identity of its own only emerged in the 1960s when universities began offering undergraduate and graduate degrees in computer science, and the first generation of formally trained *computer scientists* emerged from the campuses.

Since the advent of the electronic digital computer in 1946, the spectacular growth of the technologies associated with this machine (nowadays called generically 'information technology' or 'IT') and the related cultural and social transformation (expressed in such terms as 'information age', 'information revolution', 'information society') is visible for all to see and experience. Indeed, we are practically engulfed by this techno-social milieu. The *science*—the intellectual discipline—underlying the technology, however, is less visible and certainly less known or understood outside the professional computer science community. Yet computer science surely stands alongside the likes of molecular biology and cognitive science as being amongst the most consequential new sciences of the post-Second World War era. Moreover, there is a certain strangeness to computer science that compels attention and sets it apart from all other sciences.

My intent in this book is to offer the intellectually curious reader seriously interested in scientific ideas and principles the basis for an understanding of the fundamental nature of computer science; to enrich, if you will, the public understanding of this strange, historically unique, highly consequential, and still new, science. Put simply, this book strives to answer in direct, immediate, and concise fashion the question: *What is computer science?*

Before we proceed, some terminological clarity is in order. In this book I will use the word *computing* as a verb to denote a certain kind of activity; *computation* is used as a noun to signify the outcome of computing; *computational* is used as an adjective; *computer* is a noun which will refer to a device, artefact, or system that does computing; *artefact* refers to anything made by humans (or, sometimes, animals); and a *computational artefact* is any artefact that participates in computational work.

Finally, a caveat must be stated. This book begins by accepting the proposition that computer science is indeed a science; that is, it manifests the broad attributes associated with the concept of science, notably, that it entails the systematic blend of empirical, conceptual, mathematical and logical, quantitative and qualitative modes of inquiry into the nature of a certain kind of phenomena. Questioning this assumption is an exercise in the philosophy of science that is beyond the scope of this book. The abiding issue of interest here is the *nature* of computer science *qua* science and, especially, its distinct and distinguishing character.

Acknowledgements

I thank Latha Menon, my editor at OUP for her support and sage advice on this project from its very onset. Her comments on the penultimate version of the book were especially insightful.

Jenny Nugee always responded readily with editorial help and suggestions at various stages of this work. I thank her.

Four anonymous readers of two different drafts of the manuscript offered invaluable suggestions and comments which I took seriously. I am most grateful to them and wish I could acknowledge them by name.

My thanks to Elman Bashar for preparing the illustrations.

Portions of this material were used in an upper-level undergraduate course on 'Computational Thinking' which I have taught on several occasions to non-computer science majors. Their responses have been most helpful in shaping and sharpening the text.

Finally, as always, my thanks to members of my family. In their different ways each continues to provide the sustenance that makes living the life of the mind worthwhile.

List of illustrations

Chapter 1
The 'stuff' of computing

What is computer science? A, now classic, answer was offered in 1967 by three eminent early contributors to the discipline, Alan Perlis, Allen Newell, and Herbert Simon, who stated, quite simply, that computer science is the study of computers and their associated phenomena.

This is a quite straightforward response and I think most computer scientists would accept it as a rough and ready working definition. It centres on the computer itself, and certainly there would be no computer science without the computer. But both computer scientists and the curious layperson may wish to understand more precisely the two key terms in this definition: 'computers' and their 'associated phenomena'.

An automaton called 'computer'

The computer is an *automaton*. In the past this word, coined in the 17th century (plural, 'automata') meant any artefact which, largely driven by its own source of motive power, performed certain repetitive patterns of movement and actions without external influences. Sometimes, these actions imitated those of humans and animals. Ingenious mechanical automata have been devised since pre-Christian antiquity, largely for the amusement of the wealthy, but some were of a very practical nature as, for example, the water clock said to be invented in the 1st century CE by the engineer Hero of Alexandria. The mechanical weight-driven clock invented in 15th-century Italy is a highly successful and lasting descendant of this type of automaton. In the Industrial Revolution of the 18th century, the operation of a pump to remove water from mines motivated by the 'atmospheric' steam engine invented by Thomas Newcomen (in 1713), and later improved by James Watt (in 1765) and others, was another instance of a practical automaton.

Thus, mechanical automata that perform physical actions of one sort or another have a venerable pedigree. Automata that mimic cognitive actions are of far more recent vintage. A notable example is the 'tortoise' robot *Machina Speculatrix* invented by British neurophysiologist W. Grey Walter in the late 1940s to early 1950s. But the automatic electronic digital computer, developed in the second half of the 1940s, marked the birth process of an entirely new genus of automata; for the computer was an artefact designed to simulate and imitate certain kinds of human *thought* processes.

The idea of computing as a way of imitating human thinking—of the computer as a 'thinking machine'—is a profoundly interesting, disturbing, and controversial notion which I will address later in the book, for it is the root of a branch of computer science called *artificial intelligence* (AI). But many computer scientists prefer to be less anthropocentric about their discipline. Some even deny that computing has any similarity at all to autonomous human thinking. Writing in the 1840s, the remarkable English mathematician Ada Augustus, the Countess of Lovelace, an associate of Charles Babbage (see Preface) pointed out that the machine Babbage had conceived (called the Analytical Engine, the first incarnation of what a century later became the modern general purpose digital computer), had no 'pretensions' to initiating tasks on its own. It could only do what it was ordered to do by humans. This sentiment is often repeated by modern sceptics of AI, such as Sir Maurice Wilkes, one of the pioneers of the electronic computer. Writing at the end of the 20th century and echoing Lovelace, he insisted that computers only did what 'they had been written to do'.

So what *is* it that computers *do* which sets them apart from every other kind of artefact, including other sorts of automata? And what makes computer science so distinctive as a scientific discipline?

For the purpose of this chapter, I will treat the computer as a 'black box'. That is, we will more or less ignore the internal structure and workings of computers; those will come later. For the present we will think of the computer as a generic kind of automaton, and consider only *what* it does, not *how* it does what it does.

Computing as information processing

Every discipline that aspires to be 'scientific' is constrained by the fundamental *stuff* it is concerned with. The stuff of physics comprises matter, force, energy, and motion; that of chemistry is atoms and molecules; the stuff of genetics is the gene; and that of civil engineering comprises the forces that keep a physical structure in equilibrium.

A widely held view amongst computer scientists is that the fundamental stuff of computer science is *information*. Thus, the computer is the means by which information is automatically retrieved from the 'environment', stored, processed, or transformed, and released back into the environment. This is why an alternative term for computing is *information processing*; why in Europe computer science is called 'informatique' or 'informatik'; and why the 'United Nations' of computing is called the International Federation for Information Processing (IFIP).

The problem is that despite the founding of IFIP in 1960 (thus giving official international blessing to the concept of information processing), there remains, to this day, a great deal of misunderstanding about what information *is*. It is, as Maurice Wilkes once remarked, an *elusive* thing.

'Meaningless' information

One significant reason for this is the unfortunate fact that the word 'information' was appropriated by communication engineers to mean something very different from its everyday meaning. We usually think of information as telling us something *about* the world. In ordinary language, information is *meaningful*. The statement 'The average winter temperature in country X is 5 degrees Celsius' tells us something about the climate in country X; it gives us information about X. In contrast, in the branch of communication engineering called 'information theory', largely created by American electrical engineer Claude Shannon in 1948, information is simply a commodity transmitted across communication channels such as telegraph wires and telephone lines. In information theory, information is devoid of meaning. The unit of information in information theory is called the *bit* (short for 'binary digit') and a bit has only two values, usually denoted as '0' and '1'. However, in this age of personal computers and laptops, people are more familiar with the concept of the *byte*. One byte consists of eight bits. Since each bit can have one of two values, a byte of information can have 2^8 (= 256) possible values ranging from 00000000 to 11111111. What bits (or bytes) *mean* is of no concern in this sense of 'information'.

In computing, information processing in this meaningless sense is certainly relevant since (as we will see) a physical computer, made out of electronic circuits, magnetic and electromechanical devices, and the like (collectively dubbed 'hardware'), stores, processes, and communicates information as multiples of bits and bytes. In fact, one of the ways in which the capacity and performance of a computational artefact is specified is in terms of bits and bytes. For example, I may buy a laptop with 6 gigabytes of internal memory and 500 gigabytes of external memory ('hardrive'), (where 1 gigabyte = 10^9 bytes); or we may speak of a computer network transmitting information at the rate of 100 megabits/second (where 1 megabit = 10^3 bits).

'Meaningful' (or semantic) information

But the physical computer is (as we will see in Chapter 2) only one kind of computational artefact. Meaningless information is just one kind of information the computer scientist is interested in. The other, more significant (and arguably more interesting), kind is information that has meaning: *semantic* information. Such information connects to the 'real world'—and in this sense corresponds to the everyday use of the word. For example, when I access the Internet through my personal computer, information processing certainly occurs at the physical or 'meaningless' level: bits are transmitted from some remote computer ('server') through the network to my machine. But I am seeking information that is about something, say the biography of a certain person. The resulting text that I read on my screen means something to me. At this level, the computational artefact I am interacting with is a semantic information processing system.

Such information can, of course, be almost anything about the physical, social, or cultural environment, about the past, about thoughts and ideas of other people as expressed by them publicly, and even about one's own thoughts if they happen to be recorded or stored somewhere. What such meaningful information shares with meaningless information, as computer scientist Paul Rosenbloom has noted, is that it must be expressed in some physical medium such as electrical signals, magnetic states, or marks on paper; and that it resolves uncertainty.

Is information *knowledge*?

But consider an item of semantic information such as the biography of an individual. On reading it, I can surely claim to possess *knowledge* about that individual. And this points to the second source of confusion about the concept of information in ordinary language: the conflation of information with knowledge.

The poet T.S. Eliot had no doubt about their distinction. In his play *The Rock* (1934) he famously asked:

> Where is the wisdom we have lost in knowledge?
> Where is the knowledge we have lost in information?

Eliot was clearly implying a hierarchy: that wisdom is superior to knowledge, and knowledge to information.

Computer scientists generally avoid talking about wisdom as being beyond the scope of their purview. But they have also remained somewhat uneasy about distinguishing knowledge from information, at least in some contexts. For example, in AI, a subfield of computer science, a long-standing problem of interest has been knowledge *representation*—how to represent knowledge about the world in computer memory. Another kind of problem they study is how to *make inferences* from a body of knowledge. The kinds of things AI researchers recognize as knowledge include facts ('All men are mortal'), theories ('Evolution by natural selection'), laws ('Every action has an equal and opposite reaction'), beliefs ('There is a God'), rules ('Always come to a dead stop at a stop sign'), and procedures ('how to make seafood gumbo'), etc. But in what way such entities constitute knowledge and not information remains largely unsaid. AI researchers may well claim that what they do, in their branch of computer science, is knowledge processing rather than information processing; but they seem to fall shy of explaining why their concern is knowledge and not information.

In another specialty known as 'data mining' the concern is 'knowledge discovery' from large volumes of data. Some data mining researchers characterize knowledge as 'interesting' and 'useful' patterns or regularities hidden in large databases. They distinguish knowledge discovery from information retrieval (another kind of computing activity) in that the latter is concerned with retrieving 'useful' information from a database on the basis of some query, whereas the former identifies knowledge that is more than just 'useful' information, or more than patterns of regularity: such information must be 'interesting' in some significant sense to become knowledge. Like T.S. Eliot, data mining researchers rate knowledge as superior to information. At any rate, knowledge processing is what data mining is about rather than information retrieval.

Luciano Floridi, a philosopher of computing, offered the following view of the information/knowledge nexus. Information and knowledge bear a 'family resemblance'. They are both meaningful entities but they differ in that information elements are isolated like bricks whereas knowledge relates information elements to one another so that one can produce new inferences by way of the relationships.

To take an example: suppose, while driving, I hear on my car radio that physicists in Geneva have detected a fundamental particle called the Higgs boson. This new fact ('The Higgs boson exists') is certainly a piece of new information for me. I may even think that I have acquired some new knowledge. But this would be an illusion unless I can connect this information with other related items of information about fundamental particles and cosmology. Nor would I be able to judge the significance of this information. Physicists possess an integrated web of facts, theories, laws, etc., about subatomic particles, and about the structure of the universe that enable them to assimilate this new fact and grasp its significance or consequences. They possess the knowledge to do this, while I have merely acquired a new piece of information.

Is information *data*?

In mentioning 'data mining', I have introduced another term of great relevance: *data*. And here is yet another source of ambiguity in our making sense of the information concept, especially in the computer science community.

This ambiguity, indeed confusion, was remarked upon by the computer scientist Donald Knuth as far back as 1966, a time when computer science, emerging as a scientific discipline in its own right, was demanding the invention of new concepts and clarification of old ones. Knuth noted that in science there appeared to be some confusion concerning the terms 'information' and 'data'. When a scientist executes an experiment involving measurement, what is elicited might be any one of four entities: the 'true' values of that which is measured; the values that are actually obtained—approximations to the true values; a representation of the values; and the concepts the scientist teases out by analysing the measurements. The word 'data', Knuth asserted, most appropriately applies to the third of these entities. For Knuth, then, speaking as a computer scientist, data is the *representation* of information obtained by observation or measurement in some precise manner. So, in his view, information precedes data. In practice, the relationship between information and data is as murky as that between information and knowledge. Here, I can only cite a few of the diverse views of this relationship.

For Russell Ackoff, a prominent systems and management scientist, data constitute the outcome of observations; they are representations of objects and events. As for information, Ackoff imagined someone asking some questions *of* data which is then 'processed' (presumably by a human being or a machine) to afford answers, and this latter is information. So according to Ackoff, *contra* Knuth, data precedes information.

For Luciano Floridi, data also precedes information but in a different sense. Data exists, according to Floridi, only when there is an absence of uniformity between two states of a system. As he puts it, a datum (the infrequently used singular of 'data') exists whenever there are two variables, x and y such that $x \neq y$. So, for Floridi, data is a condition which itself has no meaning except that it signifies the presence of difference. When I am approaching a traffic light for instance, my observation of a red signal is a datum because it could have been otherwise: yellow or green.

Given this definition of data, Floridi then defines information as one or more data elements that are structured according to some rules, and are meaningful. To use the linguist's jargon, information is data when it possesses both syntax and semantics. Thus, my observation of the red traffic signal, a datum, becomes information because the meaning of the red light is that 'motorists must stop at the traffic light'. If I did not associate this action with the red light, the latter would remain only a datum.

As a final example, for AI researchers Jeffrey Shrager and Pat Langley, data do not result *from* observation; rather, observation *is* data; more precisely, what is observed is selectively recorded to qualify as data. Information does not figure in their scheme of things.

The programmer's point of view

These examples suffice to demonstrate the murkiness of the information/data connection from different perspectives. But let me return to Knuth. His definition of data reflects to a large extent, I think, the view of those computer scientists who specialize in another aspect of computer science, namely, computer programming—the techniques by which humans communicate a computational task to the computer (a topic I discuss later in this book). Even while paying lip service to the idea of computing as information processing, programmers and programming theorists do not generally reflect on 'information'; rather, they are more concerned with the Knuthian idea of data. More precisely, they concern themselves with data as the fundamental objects ('data objects') upon which computations are performed; and, thus, they are preoccupied with the classification of data objects ('data types'), the rules for representing complex data objects ('data structures'), and the rules for manipulating, processing, and transforming such data objects to produce new data objects. For such computer scientists it is data that matters, not information, not knowledge. To be more exact, programmers take for granted that there is information 'out there' in the 'real world'. But the interesting question for them is how to represent real world information in a form that is appropriate not only for automatic computing but also for human understanding. (Needless to say, other practitioners, such as historians, statisticians, and experimental scientists, do not usually regard data in this fashion.)

I will elaborate on this later in the book. But to give a very simple example of the programmer's view of data: in a university environment there will exist information in the registrar's office about its body of enrolled students: their names, dates of birth, home addresses, email addresses, names of parents or guardians, the subjects they are majoring in, the courses taken, the grades obtained, scholarships held, fees paid, and so on. The university administration needs a system that will organize this information in a systematic fashion (a 'database') such that, perhaps, information concerning any particular student can be accurately and speedily retrieved; new information about existing or new students can be inserted; the progress of individual students can be efficiently tracked; and statistics about the student population as a whole or some subset can be gathered. The programmer given the task of creating such a system is not concerned with the information per se, but rather, given the nature of the information, how to identify the basic data objects representing student information, construct data structures representing the data objects, and build a database so as to facilitate the computational tasks the university administration demands.