Information in Computer Science

J. Michael Dunn

Introduction

Before we address the topic of "Information in Computer Science," it is obvious that we will have to say something about what is meant by "information." I expect this to be a common preoccupation of the many authors contributing to this volume. Though perhaps not as obvious, it turns out equally important that we should be clear what is meant by "Computer Science." Incidentally, we will take for granted the meaning of the word "in" and never say "It depends upon what the meaning of the word 'in' is."

Before continuing I want to tip my hat to Luciano Floridi for his insight in establishing the very concept of the "philosophy of information." His books Florida (1999, 2004) and various papers really legitimated this as a major area in philosophy and not just a minor topic. My minor disagreement with him below should not be seen as undercutting my fundamental respect. I also want to thank the editors of this *Handbook on the Philosophy of Information,* Pieter Adriaans and Johan van Benthem, for their leadership in fostering this area, and particularly Pieter Adriaans for his helpful comments on this chapter.

What is Information?

Plato defined knowledge as "justified true belief." I like to think of information, at least as a first approximation and what is left from knowledge when you subtract, justification, truth, and belief. It is as it were an idle thought. Anyone who has searched for information on the Web does not have to have this concept drummed home. So much of what we find on the Web has no truth or justification, and one would have to be a fool to believe it. It is something like a Fregean "thought," i.e., the "content" of a belief that is equally shared by a doubt, a concern, a wish, etc. It might be helpful to say that it is what philosophers call a "proposition," but that term itself would need explanation.

Some people believe information must be true. Floridi (2003) has claimed this, and Fetzer (2004) has responded. Floridi's point has more to do with technical considerations than natural language considerations, most notably to deal with a semantic paradox from Carnap and Bar-Hillel, that on the standard technical definition of information a contradiction contains the maximum amount of information (see below).

Fetzer gives several examples from ordinary life about false information, or "misinformation," e.g., giving wrong directions to Hyde Park. He does give Flordi a conceivable defense of his position, saying that "Floridi might want to defend his position by claiming that false information is to information as artificial flowers are to flowers." I have heard a similar defense in a story of the "Information Booth" in a railway station and how it would be misnamed if it gave out false information. But note that I said "false information" in a very natural way. I think it is part of the pragmatics of the word "information" that when one asks for information, one expects to get true information, but it is not part of the semantics, the literal meaning of the term. If there is a booth in the train

station advertising "food," one expects to get edible, safe food, not rotten or poisoned food. But rotten food is still food.

It is common in some circles (I have certainly done it) to make a distinction between data and information, and information and knowledge. Data is what is produced by instruments, but does not become information until it is somehow recorded -- typically these days in a computer. And information does not become knowledge until it at least meets Plato's tests. (There is a large philosophical literature starting with Gettier that says these are not enough.) One of the most important of these in the context of "the information flood" is that it somehow become "believed" (internalized, mastered) by humans. I believe that information technology, viewed as an augmentation of the human condition, raises new issues about the meaning of knowledge, or perhaps some extended sense of it. What happens when I have large, distributed databases, or even the Web with search engines like Google? It seems to me that changes in some fundamental sense the old fashioned paradigm of the expert who has internalized not just the belief but also all of the justification for that belief.

It is equally common to conflate these terms. Thus a so-called data base might be better called an "information base," and "knowledge representation" is better called "information representation."

Once we discuss what Computer Science is we shall return to the concept of information, and see how it has been defined in a more technical setting.

What is "Computer Science"?

It turns out that there are philosophical or at least conceptual issues arising in about just what is included under the label "Computer Science."

In North America we find at various universities that the academic units that house at least some computer scientists have many variations in their name and structure. Variants include Computing Science, Computer Engineering, Computer and Information Science, Library and Information Science, Information Science, (Computer/Management) Information Systems, Informatics, Information Technology (IT), Information and Communications Technology (ICT). This keeps you on your toes in getting the names right at various universities – you also have to be conscious of the difference between a department and a school (or college) and you also have to keep singulars and plurals straight. E.g. University of California at Irvine has the new Brett School of Computing and Information Sciences, and within it there is a Department of Informatics. Penn State University has a new School of Information Technologies and Science. Indiana University has a new School of Informatics, and at Bloomington Computer Science is a department within the school. And if we look at Europe we find the use of the word "Informatics" very frequently used as more or less synonymous with "Computer Science," especially in Germany and France. In Holland, Scandinavia, and perhaps Italy, it seems to have a somewhat broader meaning, and in the U.K broader yet (with the University of Edinburgh being perhaps the very broadest so as to include information processing systems, whether artificial or natural – this last including what is more commonly called Cognitive Science).

The exact boundaries of "computer science" are clearly difficult to define, but I take it from the above discussion that it can include the study of computing, not just computing machines, and it can also include the study of information, at least in digital form.

Computers as Information ("Data") Processors

"Computers" store and process digital information, which can be thought of as a series of bits (1, 0) or a series of switch settings (on, off) or a series of voltages (high, low). The future may replace the current electronics with "spintronics" where the bits can be represented on single electrons by "spin up," "spin down."

The numbers 1 and 0 are abstract, but the numerals "1" and "0", switch settings, and voltages are not. These are implementations of the abstract bits, and they are conventional in nature. Not only might there be different implementations, but (they could have been reversed for example. They need to be distinguishable (discrete). In a real computer, a switch is not just in the two states: closed (on) or open (off). It can instead be in the process of closing (or opening). And a voltage can be somewhere in the middle of the process say of dropping from high to low. It depends on when the measurement is taken.

This helps emphasize the arbitrariness of picking out certain parts of the physical world as implementations of the two bits. Computers may be digital, but the world that they are a part of is not (except at the quantum level, at least when a measurement is made). In practice computers avoid the intermediate values by settling on ranges of "fault tolerance" of the strength of the voltages. Also a clock counts the number of cycles per second and only measures the voltages at approximately their maximum and minimum. This is why we talk of computers being so-and-so numbers of cycles per second. This relates to the refresh rate of the central processing unit (CPU chip).

Information in a purely technical sense is a string of bits. But in a more intuitive/practical sense information as stored in a computer is a derivative notion depending on the encoding (encoder?) and the program (programmer?) and the interpreter *(*user?). Must this derive from human intelligence, or can there be "real" AI that does not depend on the parenthetical items?

Where is the "Computing" in a Computer?

Before the advent of the modern computer in the form of a machine, there were people who did complicated series of mathematical calculations, and these people were actually called computers. During WWII many of these were women doing ballistics calculations. Indeed, the first electronic computers were developed at the end of WWII to do calculations for ballistics, the atomic bomb, code-breaking, etc. This all led as we know successively to main-frame computers, mini computers, workstations, and most familiar to most of us now, personal computers.

Consider the common uses of "personal computers": word processing, e-mail, calendar, notes, address book, games, digital photos/video, CD/DVD player, etc. And we now have new applications such as VOIP (Voice Over IP). Spreadsheets are the exception in "office

suites" – they actually are used in computing in something like the original meaning of the word.

Other "computers" that are not used for computation in any usual sense include: digital cameras, digital thermostats, cell phones, PDA's, digital music editors, slot machines trains, planes and automobiles, etc. Pervasive or ubiquitous computing envisages computers, or at least network nodes, in more or less everything that we used in our daily lives. With RFID tags this can even include the clothes we purchase, and wear. The digital revolution is with us!

History of the Concept of Information

Early History. I defer to Pieter Adriaans' chapter in this Handbook as a kind of division of labor, and also reference a earlier work of my own on the history of the concept of information, Dunn (2001).

The part of this last that I need to go over quickly is the development of what has been called a "UCLA Proposition." I believe the term originated with Alan Anderson, but the concept originated in the work of Boole, with his dual interpretation of what is now called "Boolean algebra." Famously, the elements of a Boolean algebra can be interpreted as either classes (operated upon by relative complement, intersection, union) or as propositions (operated upon by negation, conjunction, disjunction). Boole was well aware of this and he termed the first his primary interpretation and the second his secondary interpretation. Boole connected these: a proposition can be regarded as the set of "times" in which it is true (Dipert, 1978). If we take "times" metaphorically (as something like occasions, cases, states, or possible words, we find the beginning of a thread that weaves through Carnap, Montague, Kripke), where propositions are viewed respectively as sets of "state-descriptions," "indices," "possible worlds." Carnap and Bar-Hillel has actually two concepts: "information," i.e., the set of state-descriptions that make a proposition true, and "content" the set of state descriptions that make a proposition false. Of course classically, one is just the set-theoretic complement of the other, and so one can choose to work with either one. This may be an over-simplifying assumption as we will see later..

Carnap and Bar-Hillel also suggested a numeric measure could be given by counting the number of states, and also suggested another numeric measure:

$Cnt\#(A) = 1 - prob(A)$.

Computer science represents information as a string of zeros and ones ("bits"). Given a set I of indices set, any set $A \subseteq I$ can be understood as an indexed set of bits: 1 if i is in, 0 if i is out. This is an abstract version of the Carnap and Bar-Hillel notion of information: view the members of I as being abstractions of state-descriptions, and view the indexing functions as characteristic functions.

Classical Information Theory

Shannon Information. While logicians were busy with various variations on the theme we have labeled "UCLA propositions," Claude Shannon (1948) was independently developing

the quantitative counterpart to a UCLA proposition (actually to its complement). Shannon suggested that we measure that information in a message as roughly the inverse of probability; formally log to the base 2 ($\log_2$) of the inverse of the probability.

Frequently the messages have meaning: that is they are referred to or correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages.

The intuitive idea behind Shannon's measure is that the more surprising a message is, the more information it conveys. If I tell you that the sun will rise tomorrow, this is very unsurprising. But if I say that it won't, this is very surprising indeed, and in some intuitive sense more informative.

This corresponds to the quantitative measure of content proposed by Carnap and Bar-Hillel in that rather than the more probable getting the highest measure, it is the least probable. Carnap and Bar-Hillel used the arithmetic inverse of addition, which is subtraction (from 1). Alternatively, Shannon chose in effect to use the multiplicative inverse, which is division (by 1). So the multiplicative inverse of n is 1/n. Both have the effect of inverting a high number to a low number (and vice versa) so as to make the more surprising be the less informative.

Shannon's definition corresponds elegantly with the notion of information as a string of bits:

$\text{Inf}(s) = \log_2 \text{inv Prob}(s) = \text{Length}(s)$.

Thus the information in a binary string is just the length of the string.

Remember that log to the base 2 ($\log_2$) is the inverse of the corresponding power of 2, i.e., the following are equivalent:

$x = 2^y$,
$y = \log_2 x$.

Thus:

$\log_2$ of 2 = 1 since $2^1 = 2$,
$\log_2$ of 4 = 2 since $2^2 = 4$,
$\log_2$ of 8 = 3 since $2^3 = 8$,
$\log_2$ of 16 = 3 since $2^4 = 16$, etc.

Formally the Shannon information measure of an event is given by:

$\text{info}(E) = \log_2 [1/\text{prob}(E)]$.

Considering a few examples helps. Let us suppose that someone in the next room is tossing a fair coin. Let us decide that 1 means heads, and 0 means tails. This incidentally indicates a very important aspect of information, which Shannon emphasized. All information is relative to an initial encoding.

If they do just one toss, and I, standing in the doorway, yell out "1," I have given you a certain amount of information. How much? Well since prob = ½, info = $\log_2$(inverse ½) = $\log_2(2) = 1$.

What if there are two tosses, and I yell "1 0"? This time we figure that since prob = ¼, then info = $\log_2$(inverse 1/4) = $\log_2$ (4) = 2.

And with three tosses, and "101"? This time we figure that since prob = 1/8, then info = $\log_2$(inverse 1/8) = $\log_2$ (8) = 3.

Not only is this pleasant mathematically, but as Shannon noted it has a common sense appeal as well.

If I have 2 books and buy a 3rd (as an idealization let's assume that all are of equal length), from an intuitive point I view I have not doubled the amount of information I possess, rather added just one more book (increased it by half). Shannon's formula above accords nicely with this intuition.

Imagine the economic and practical consequences of pricing books based on Shannon's formula if "$\log_2$" were not thrown in. Before Amazon sells you a book, they would first have to send out an appraiser to find out how many books you already have, or perhaps more feasibly check how many books you have bought from them.

The "Paradox of the Monkeys"

Not everyone finds Shannon's definition intuitive. It is a kind of "paradox" that this means that the works of Shakespeare contain less information that a random rearrangement of their letters and punctuation marks. There is the well-known story, commonly attributed to Aldous Huxley, that a bunch of monkeys, typing randomly for a sufficiently long time, would eventually (through pure chance and statistics), type all of Shakespeare's works. Most of the time, the monkey will type complete nonsense, but occasionally it will type *Hamlet.* Of course *Hamlet* would appear over and over again out of this "mist" of typing (and so would Faulkner's *Sound and the Fury*, and Miss Manners' column from last weekend's newspaper, etc.) but the vast preponderance would be totally meaningless.

The supposed Paradox of the Monkeys is that the utter nonsense of most of this typing would contain more information than the small amount of time that Hamlet might appear. This is my name because of my way of putting it, but the point has been made by others. On one way of thinking about this, it would seem that when the monkey is typing scenes from *Hamlet*, the characters are more predictable than when typing total nonsense. We

know that spaces occur rather frequently, that most of the strings between spaces (words) occur in a reasonably small dictionary, that the words "Hamlet" and "Ophelia" occur with some relative frequency, etc. The Monkey Paradox put quickly is that complete nonsense would seem to carry more information than careful than the words of a great author.

I believe there are several answers, in stages, to this supposed "paradox" (and hence the quotes). First, let us suppose that the monkey is in fact typing completely at random. Then there is in fact no more likelihood that she will type the 18 character string "signifying nothing" than the string "gnihton gniyfingis" (the last is just a reversal of the first). This is just like the fact that if the monkey had typed 3 "a"s, this in no way increases the probability of the next character being "a" than does a gambler's being dealt 3 aces increase the probability that the next card will be an ace. In truly random sequences, patterns are in the mind of the beholder.

In a recent paper Dalkilic *et al* (2006) have devised a computer program that can recognize "authentic texts" as those that fall into "the sweet spot" between total randomness and total predictability.

Of course patterns can arise because of hidden causal influences. I am reminded of a story about Raymond Smullyan, a first-class magician as well as a first-class logician, who while teaching some elementary probability to a class, pulled out a deck of cards and proceeded to deal a Royal Flush. Raymond told the class that surprising as this might seem, this hand was just as likely as any other hand. And of course he repeated the procedure producing one Royal Flush after another until the class broke up with laughter.

Can it be that a predictable sequence can in fact provide significant information? Consider the following which is based on a true event. During the Second World War, the Allies had broken the German's Enigma code. But the Germans used certain special code names for ships, so it didn't matter if one had deciphered the code name – it was still a code name. But the Allies had good reasons to predict that a certain code name was that of a certain ship they knew. So they sent a message with low encryption mentioning something of interest about that ship. They listened to the German traffic, and, as predicted, that information was sent on mentioning that ship using its code name. This of course confirmed that the prediction was right. Thus even though it was say highly probable that the allies had the right code name, and hence highly probable that it would show up in the German transmission, it seems it still contained significant information when it did.

My reading of this is that what was significantly increased was not the actual information. The probability of the code name being the name of that specific ship was not significantly changed from a purely numerical point of view. Risk was reduced.

It is commonly recognized that the concept of risk involves both probability and the cost of the consequences, and the standard mathematization of "risk" used by insurance companies (and in fact anyone involved in so-called "risk assessment") involves a function in both of those variables. The standard notion is that the risk of an event is a product of the probability and the cost of the consequences, and those of us who are rational use that notion on a daily basis, with some very rough idea of both the probability and the cost of a given event.

I suggest that some similar composite notion is involved with what we might call the "significance" of a piece of information. I am not sure of how to best mathematicize it, but as at least a first approximation I suggest that we just multiply the inverse of the probability by the cost.

Solomonoff-Kolmogorov-Chaitin "Algorithmic Information"

We mention this just briefly to indicate that there are other possible mathematical definitions of information. Suppose that one has a very long sequence:

11.00100100001111110110101010001000100001011010001100001000110100 11

Perhaps it can be given by an even shorter algorithm that computes this sequence. In fact it can as is given by the hint of the decimal point in the third position – this is just pi in binary notation and expressed to 64 places. Roughly the algorithmic information of a sequence is the length of the shortest algorithm that generates the sequence. This is discussed at some length in the chapter of this *Handbook* by Pieter Adriaans, and is often known under the name "algorithmic complexity."

Von Neumann Duality

The so-called von Neumann model of a computer emphasizes that there are both static and dynamic aspects of a computer. This is the duality of programs and "data" (stored programs). A stored program is simply a series of bits (information) that can be taken as an input to a program (even itself, perhaps copying it first to make the process transparent). And in principle any series of bits can be called as a program. Yes, it is very likely that it will not execute and there will be a "syntax error" message – but this can be viewed as just an "identity program" that leaves the input unchanged.

The von Neumann model was actually anticipated by Turing's Universal Machine. Turing observed that all of what we now call Turing machines could be enumerated

$$M_0, M_1, M_2, \ldots$$

and by some clever reasoning he constructed a universal machine M that given an ordinary input n plus the input of the appropriate index i would compute

$$M(n, i) = M_i(n).$$

In effect the index i codes up the program that the machine M_i implements.
One can similarly argue that the von Neumann model was also anticipated by other early models of computation, e.g. by the Feys-Curry Combinatory Logic and by Church's Lambda Calculus. In both cases these were originally regarded as untyped so that a term could function as either a function or as an argument, and a term can even be applied to itself. A Turing machine M_i can also be applied to itself for i one can look at $M_i(i)$.

For some time it was puzzling how to give mathematical models of these systems because self-reference or self-application is notorious in its tendency to produce paradoxes. Think of the famous Russell paradox of the set of all sets that are not members of themselves.

But Dana Scott, working with Christopher Strachey, produced models of the Lambda Calculus and of Combinatory Logic. There were in fact two kinds of models, the "graph model" and a model based on "continuous lattices." The latter was extended by Gordon Plotkin and has become known as the "domain model," or as "denotational semantics."

There is another way of approaching the von Neumann duality that stems from the Routley-Meyer semantics for relevance logic. This semantics famously uses a ternary frame (U,R), where U is a non-empty set and R is a 3-placed relation on U. In the actual semantics for relevance logic there may be other features on the frame, such as a distinguished world or set of worlds, and an involution * to model negation. We overlook these for simplicity here. The trick we use is to view the ternary relation $R\alpha\beta\gamma$ as a binary relation $R_\alpha\beta\gamma$ indexed by α. This means that for $A \quad U$, A can simultaneously be viewed as a UCLA proposition (a set of states) and as a program, i.e., a set of actions (binary relations) on states. This allows for a semantics for both combinatory logic and relation algebras. In the first AB is interpreted as "apply the actions coded up by the states in A to the states in B." For the second we understand both of A and B to be sets of actions (relations) and we take AB to be the point-wise relative product of these relations. Details can be found in Dunn and Meyer (1997) and Dunn (2001c) (see also Dunn (2001b)). Comparisons should also be made to "arrow logic" developed by van Benthem and his collaborators, and also the "logic of information flow" developed by Barwise and Seligman.

Information Representation, Networks, and Distributed Information

I believe that the fundamental philosophical question regarding the representation of information is: how much "lies in the eye of the beholder?" This is meant metaphorically of course – replace the word "eye" with "interpretation" or even "mind." This can be nicely illustrated by an anecdote the late Australian philosopher Ian Hinkfus once told me. He was working for IBM in the design of an early computer and the ran our of nand gates in building their prototype. So they used nor gates instead and just reinterpreted the output.

There are many ways of representing information that are of direct or indirect interest to philosophers. Those that are the closest to familiar logics are the most obvious. I am thinking of PROLOG, FOL, RDF, etc.

Perhaps one of the most astounding recycling of philosophical notions has to do with the notion of "ontology," which has become very important in the area of knowledge representation. (Incidentally knowledge representation, or KR as it is familiarly called," is a good example of the tendency to use "knowledge" when what is really meant is "information.") Recently *Communications of the ACM* devoted an entire issue to the subject of ontology. One philosophical or at least conceptual issue is just what there is to "ontology" that goes beyond more familiar and mundane classification.

Information can be viewed in a quadrant. Along say the side we have Structured / Unstructured, and across the top we have Text/Multi-Media. The primary example of structured information is to be found in a traditional tabular database where one has tables say with the names of employees, their classification, date of hiring, salary, etc. Relational databases are much more flexible, but they are still relatively rigid. The primary example of unstructured information is to be found in the World Wide Web.
Once upon a time all information was in effect textual, but again the World Wide Web has brought multi-media to the fore. One can browse the Web and find pictures and music, not just text. Perhaps an example of a structured multi-media database would be the Apple i-Store, and an unstructured example would be Napster.

Unstructured information, the Web in particular, emphasizes the importance of search engines. There is an interesting, emerging distinction between memory (storage) and search, particularly when one emphasizes unstructured storage. Extenal storage and search can be seen as genuine extensions of human abilities, much like the first writing on a cave wall or on stone tablets.

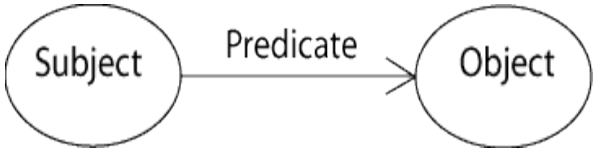
Another philosophical issue has to do with the meaning of negation. In traditional data bases one can assume the so-called "closed world assumption." Thus one does not list the non-employees in the database, and one can assume that if Joe Smith is not listed as an employee, then he is not an employee. This was made explicit in the programming logic PROLOG, based on a fragment of first order logic (Horn clauses), with an added "negation as failure."

Belnap (1977) introduced the idea of databases with inconsistent information. This was prescient, and the inconsistency of the Web demonstrates. It is worth pointing out that in a network the logic is partial (truth values are underdetermined) and with mirror sites, information may be contradictory (overdetermined). While I have given credit to Belnap for the explicit application to databases (and by extension to WWW), the idea of the 4-valued semantics to limit the effect of inconsistencies is implicit in my dissertation and explicit in Dunn (1976). See also Dunn (1985) for relevant history.

The familiar story of Napster and musical file sharing makes clear that there are ethical issues regarding shared information. We talk about "the information economy," but there seems to be no general consent that "information workers" should be paid for the information they produce. Intellectual Property (IP) raises a number of philosophical/conceptual/legal issues under the headings of Digital Rights Management (DRM) and Digital Content Management (DCM) – this last is newer and oriented towards music and video content. Incidentally, audio and video files raise the question again of just what information is. This is a realm in digital libraries where "metadata" is particularly important, especially for searches.

There are of course attempts to make the Web more structured. Languages such as HTML and particularly XML structure Websites. There is a whole planned architecture of the Web led by a group called the World Wide Web Consortium. It starts with the Rich Description Framework (RDF) which is essentially a very restricted form of positive first-order logic – no negation. RDF as a sublanguage of FOL has just conjunction, existential quantification, and binary predication.

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. This can be illustrated by a node and directed-arc diagram, in which each triple is represented as a node-arc-node link (hence the term "graph").



One significant feature from FOL that is missing from RDF is negation. Another is the restriction to binary predication. While many seeming ternary predications can be reduced to conjunctions of binary predications, this is not true of all. One wonders about adopting a "Pierceian" framework in which ternary predicates are the primitive, since it is much more plausible that all predication can be reduced to ternary predications (see Dunn and Meyer (1997)).

There are many issues raised by the digital storage and transmission of information, and by parallel computation on that information. The World Wide Web is just one example in this general framework, and perhaps the most extreme in terms of being the "Wild, Wild Web.." We cannot go into these here, but do want to mention philosophically important work by Dretske (1981), Barwise & Seligman (1997), Devlin (1997), etc. Some of this relates to Shannon's mathematical theory of communication, but there is much more to it. I refer the reader to the chapter of this *Handbook* written by J. Seligman.

Quantum Computation/Information

Our discussion here shall be brief because of limitations of space and time, and because currently the concepts of quantum computation and information a very speculative. Let me take the occasion to recommend an excellent book to the reader on this subject, Julian Brown (2001). Although this is a "popular" book it is extremely informative on both the detail and history of quantum computation, and includes a forward by on of the early proponents of quantum computing, David Deutsch. Deutsch's contributions go back to 1983, and were proceeded by Richard Feyman's suggestion in 1979.

In the classical model of a computer the most fundamental building block, the bit, can only exist in one of two distinct states, "0" or "1."

A "quantum bit" (qubit) can be in the classical "0" and "1" states, but it can also be in a superposition (coherent state) of both "0" and "1." This has to do with the well-known concept of "superposition" in which say an electron can be simultaneously in both the states "spin up" and "spin down" until measured, when it will then "decohere" into a single one of those state. Some of examples of qubits include the polarization state of a photon (i.e. parallel or perpendicular polarized to a given axis), an atomic two level system (e.g. hydrogen atom, with the electron in the ground state and the first excited state) and of course poor "Schroedinger's cat" (which is both dead and alive until a measurement is taken by looking at the cat to determine whether a radioactive atom decayed).

Consider a register of 3 classical bits: it would be possible to use this register to represent any one of the numbers from 0 to 7 at any one time. In a register of 3 qubits, the register can represent all the numbers from 0 to 7 simultaneously!

A processor that can use registers of qubits will in effect be able to perform calculations using all the possible values of the input registers simultaneously. This phenomenon is sometimes called quantum parallelism. There are thus conceivably more advantages to quantum computing that just the miniaturization given by "spintronics." In principle quantum computations can involve completely new algorithms on qubits that exploit the phenomenon of quantum parallelism. Perhaps the most famous of these is "Shor's algorithm," created by Peter Shor of AT&T Bell laboratories. This algorithm factors a large number into its prime factors. This task is classically so difficult that it forms the basis of RSA encryption, the standard method of encryption used today. This raises strongly the issues of quantum complexity theory and whether this might differ from classical complexity theory, and perhaps even that a quantum computer might be able to solve at least certain NP problems in polynomial time. But it is an open question whether factoring is classically an NP problem.

Another important quantum algorithm, though not as impressive in it seeming speed advantage is "Grover's algorithm" which can search an unsorted list of length n on average in time on the order $\sqrt{n}$ as opposed to the usual $n/2$ for the classical linear search algorithm which checks every element of a list until a match is found. Grover's algorithm starts by setting a quantum register to a superposition of all possible items in the search space. Grover's algorithm involves a sequence of simple quantum operations on the register's state. Grover describes these in terms of wave mechanics: "All the paths leading to the desired results interfere constructively, and the others ones interfere destructively and cancel each other out."

Some interpretations of quantum parallelism have used John Wheeler's "possible world" interpretation of quantum mechanics, which is of obvious philosophical interest.

It is interesting that regarding the potential for quantum computation to break RSA encryption, that one can "make lemonade out of lemons" by using certain quantum encryption devices (which in fact appear to be making faster practical headway than building quantum computers).

One question that I believe has not been sufficiently examined is the relationship between quantum computation and quantum logic. The latter was initially introduced by Birkhoff and von Neumann (1936) but had very different motivations and the concept of a "qubit" was not explicitly introduced. Dunn, Hagge, Moss, and Wang (2004) is a beginning.

Another philosophically interesting aspect of quantum computing is that it is reversible. No information is lost! This can also happen in a classical closed system, or it can be programmed with great overhead into a computation on a classical computer (one needs to keep somehow all of the previous steps). But the remarkable fact about quantum computing is that reversibility comes for free with no special attention.

Is the world ultimately digital because of quantum mechanics? Yes and no. Yes when it is measured, no when it is not. Quantum computing is a kind of hybrid between digital and analog computing. Perhaps it represents the best of both?

Rather than end on this high note, honesty compels me to mention the great practical difficulty with building a quantum computer due to the fact that coherent states are easily destroyed by small changes in their environment. As of the year 2005, it seems that the largest quantum computer ever built had a 6 qubit register: http://www.eetimes.com/showArticle.jhtml?articleID=174900229. For this reason it is important to develop fault tolerant quantum computing. One promising direction is "topological quantum computing," where the qubits are stored as "quantum knots." As all of us know who have tried to untangle a knot in our shoestring, knots are very resistant to even large changes in their environment. This was first proposed by Freedman, Kitaev, and Wang (2000), where the "knots" are braids in a 2-dimensional quasi-particle called an "anyon." The serious implementation issue is whether appropriate anyons can actually be found in nature. An excellent general article with references is Collins (2006).

Complex Systems, Modeling, Simulation, Virtual Reality

Computers have become more and more used in modeling and simulation, and can be used to model complex systems in a way that often produces unexpected outcomes. In biology it has been common place for sometime to distinguish between experiments *in vitro* (in the glass, "test tube") and *in vivo* (in the living organism). Now a new type of experiment has arisen given computer modeling: "*in silico" (in silicon,* or in the computer). This raises certain issues in scientific methodology and statistics

A related issue is "virtual reality," where computers can simulate the real world, or wildly imaginary worlds. The "CAVE" was developed at the University of Illinois in 1992. CAVE is an acronym for "Computer Assisted Virtual Environment," and the name was cleverly chosen as a take off on Plato's metaphor of The Cave in his *Republic*. In the *Republic* denizens of a cave see shadows reflected on a wall, and mistake them for the real things they represent. Plato of course wanted to say that the "real things" we see in everyday life are but poor reflections of their ideal forms. But there is another way to interpret this and that is that at least under certain conditions people cannot tell illusion from reality.

This reminds us of Descartes' Evil Demon who might deceive him (or us) into believing that we are sitting in front of a fire, etc. when in fact we are not. It is by now a commonplace skeptical argument, and is sometimes gotten at through talking about a "brain in a vat." The movie *The Matrix* is a more current example and is based on the premises that (most) people are in fact encased as cells in a large power-generating "matrix" and are deluded into thinking that they live ordinary lives by complicated computer programs manipulating their brains. Bostrom (2003) with his "simulation argument" is an extreme but well argued version of this. Heim (2001) is a good source of philosophical issues, both old and new raised by so-called "virtual reality."

Artificial Intelligence

Artificial intelligence (AI) might be viewed as a special case of simulation, and of course raises many interesting philosophical issues. There has also been a significant role for logic and philosophy in AI ever since John McCarthy was one of the three legendary founders of AI.

One issue that was raised by Alan Turing (and now called the Turing Test) has to do with the nature of intelligence and what would count as a computer showing intelligence. The Turing Test famously has to do with whether having a "conversation" with a computer under suitably disguised circumstances would allow you to determine that it is a machine and not a human. Another important issue is whether a machine can be conscious. This issue has been so widely discussed by philosophers that I just mention it here.

Nick Bostrom, Ray Kurzweil, and Bill Joy have independently been concerned with machines becoming more intelligent than humans. "Concerned" is perhaps the wrong term to use for Bostrom and Kurzweil. Bostrom co-founded (with David Pearce) the World Transhumanist Association (WTA) http://www.transhumanism.org/, "an interdisciplinary approach to understanding and evaluating the possibilities for overcoming biological limitations through technological progress," and Kurzweil (1999) has written with striking optimism about the time when machines will outrun humankind in intelligence and perhaps even in "spirituality" -- quite in opposition to the view of say the *Terminator* films. Bostrom and Kurzweil in various of their writing also suggest that human brains may in effect be uploaded into powerful computers thus extending human capabilities and even experiences by a kind of virtual reality, giving to my mind the title *The Matrix Revisited* a new meaning.

Ethical Issues

By a natural transition this takes us to ethical issues relating to information and computer science. It is difficult not to be a bit of a luddite when contemplating issues such as the above. The original Luddites were English textile workers in the early 19th century who objected to the introduction of power looms, which they felt threatened their jobs. It is ironic that Charles Babbage, the inventor of the "analytical engine" which foreshadowed the modern digital computer, was heavily influenced by the use of punched cards for programming the weaving in a Jacquard loom. Needless to say there is often the perception today that information technology threatens various individual's jobs. What started off as an issue about the "digital divide" between the information have's and have not's, has now often become an issue about off-shoring, finding the cheapest labor that still has the relevant IT knowledge. As it was famously put by Nandan Nilekani, CEO of Infosys Technologies, an Indian outsourcing company at the World Economic Forum in 2004: "Everything you can send down a wire is up for grabs" (reported by Drezner 2004). Another set of ethical issues concerning the Internet has to do with confidential, privacy, identity theft, etc. It is left as "an exercise to the reader" to say more.

References

Barendregt, H.P. (1984), *The Lambda Calculus: Its Syntax and Semantics,* North Holland Publishing

Barwise, J. and J. Seligman (1997), *Information Flow: the Logic of Distributed Systems*, Cambridge Tracts in Theoretical Computer Science, vol. 44.

Belnap, N.D., Jr. (1992), "A Useful Four-valued Logic: How a Computer Should Think", in *Entailment: The Logic of Relevance and Necessity*, Vol II, A.R. Anderson, N.D. Belnap, Jr, and J.M. Dunn, Princeton University Press; first appeared as "A Useful Four-valued Logic", *Modern Uses of Multiple-valued Logic*, J.M. Dunn and G. Epstein (eds.), D.Reidel Publishing Company, Dordrecht, 1977, and "How a Computer Should Think", *Comtemporary Aspects of Philosophy*, G. Ryle (ed.), Oriel Press, 1977.

G. Birkhoff and J. von Neumann (1936), *"*The Logic of Quantum Mechanics*," Annals of Mathematics*, vol. 37, pp. 823-843.

Bostrom, N. (1998). "How Long Before Superintelligence?" *International Journal of Futures Studies*, 2, http://www.nickbostrom.com/superintelligence.html.

Bostrom, N. (2003), "Are You Living In a Computer Simulation?", *Philosophical Quarterly*, 53, pp. 243-255.

Brown, J. (2001), *Quest for the Quantum Computer*, forward by D. Deutsch**,** Touchstone (Simon and Schuster). Originally published *as Minds, Machines, and the Multiverse: The Quest for the Quantum Computer.*

Collins , G. P. (2006), "Computing with Quantum Knots," *Scientific American,* April, pp. 57-63. Co.

Curry, H. B, and R. Feys (1958),. *Combinatorial Logic*, vol. 1, North Holland Publishing Co.

Dalkilic, M. M., W. T. Clark, J. C. Costello, P. Radiovojac (2006), "Using Compression to Identify Classes of Inauthenic Texts," *Proceedings of the 2006 SIAM Conference on Data Mining*, http://www.siam.org/meetings/sdm06/proceedings.htm.

Devlin, Keith (1997), *Logic and Information*, John Wiley & Sons.

Drezner, D.W (2004), "The Outsourcing Bogeyman," *Foreign Affairs*, May/June.

Dretske, F. (1981), *Knowledge and the Flow of Information*, 2nd Ed., Oxford Univeristy Press.

Dunn, J. M. (1976), "Intuitive Semantics for First-Degree Entailments and Coupled Trees," *Philosophical Studies*, vol. 29, pp. 149-168.

Dunn, J. M. (1985), "Relevance Logic and Entailment," in *Handbook of Philosophical Logic*, vol. 3, eds. D. Gabbay and F. Guenthner, D. Reidel, Dordrecht, Holland, pp. 117-224.

Dunn, J. M. (2001a). "The Concept of Information and the Development of Modern Logic," in *Non-classical Approaches in the Transition from Traditional to Modern Logic, ed. W. Stelzner, de Gruyter.*

Dunn, J. M. (2001b). "Ternary Relational Semantics and Beyond: Programs as Data and Programs as Instructions," *Logical Studies* (on line journal), no. 7, Institute of Logic, Russian Academy of Sciences, Special Issue: Proceedings of the International Conference *Third Smirnov Readings* (Moscow, May 24-27, 2001), Part 2, http://www.logic.ru/LogStud/.

Dunn, J. M. (2001c). "A Representation of Relation Algebras Using Routley-Meyer Frames," *Logic, Meaning and Computation: Essays in Memory of Alonzo Church,* eds. C. A. Anderson and M. Zeleny, pp. 77-108. Preliminary version in Indiana University Logic Group Preprint Series, IULG-93-28, 1993.

Dunn, J. M., Hagge, T. J., Moss, L.S., and Wang, Z. (2004), "Quantum Logic as Motivated by Quantum Computing," *The Journal of Symbolic*. vol. 70, pp. 353-359.

Dunn, J. M. and Meyer, R.K (1997). Combinatory Logic and Structurally Free Logic" (with R. K. Meyer), *Journal of the Interest Group in Pure and Applied Logic*, Oxford University Press, vol. 5, no. 4, pp. 505-537.

Fetzer, J. H. (2004), "Information: Does it Have to be True?", *Minds and Machines,* vol. 14, pp. 223-229.

Floridi, L (1999), Philosophy and Computing - An Introduction. Routledge, London.

Floridi, L. (2003), "Outline of a Theory of Strongly Semantic Information," *Minds and Machines,* vol. 14, pp. 197-221.

Floridi, L. (2004), editor, *The Blackwell Guide to the Philosophy of Computing and Information*, Blackwell Publishing.

Floridi, L. (2005), "Is Information Meaningful Data?", *Philosophy and Phenomenological Research*, vol. 70, pp. 351-370.

M. H. Freedman, A. Kitaev, and Z. Wang (2000), "Simulation of topological field theories by quantum computers'," 17th of March 2000, Physics e-Print archive, http://arXiv.org/abs/quant-ph/0001071.

Grover, L. (1996), "A fast quantum mechanical algorithm for database search," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 212-219

Heim, M. (1994). *The Metaphysics of Virtual Reality*. Oxford University Press.

Kurzweil, R. (1999), *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, Viking: New York.

Plotkin, G. D. (1976), *"*A Power Domain Construction," *SIAM Journal of Computing* , 5, pp. 452-487.

C. E. Shannon (1948), "`A mathematical theory of communication,"' *Bell System Technical Journal,* vol. 27, July and October , pp. 379-423 and 623-656.

Shor, P (1994), "Algorithms for quantum computation: Discrete logarithms and factoring," *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pp. 124-134.

Stoy, J. (1985), *Denotational Semantics: The Scott-Strachey Approach to Programming Languages,* MIT Press, Cambridge, MA, 1985.

Turing, A. M., 1936-7, "On computable numbers, with an application to the Entscheidungsproblem," *Proceedings of the London Mathematica. Society,* ser. 2, vol. 42, pp. 230-265.

W. Weaver and C. E. Shannon (1949), *The Mathematical Theory of Communication,* Urbana, Illinois: University of Illinois Press, republished in paperback 1963.