3
COMPUTATION AND INFORMATION

Gualtiero Piccinini and Andrea Scarantino

Computation and information are closely related. There are several notions of computation and several notions of information, so we need to be careful in sorting out their relations (Piccinini and Scarantino 2011).

Varieties of computation

The paradigmatic notion of computation is digital computation. Digital computation is the manipulation of discrete states, which we call "digits." For present purposes, digits are values of variables that can take one of finitely many different states, such as 1 or 0. A digit in this sense need not represent a number (or anything else); it is simply a state that a computer can distinguish reliably from other kinds of state. Digital computers manipulate sequences of digits according to a rule defined over the digits; the rule specifies a relationship that must hold between input digits and output digits; typically, the manipulation proceeds by means of a series of discrete steps that change the digits. Words made out of letters from the English alphabet are strings of digits in the present sense. A computation over such digits could be defined by the following rule: take English words in an arbitrary order as input and return them in alphabetical order as output.

Ordinary digital computers are not limited to performing one type of computation on their inputs. In other words, they are not limited to following one rule for processing digits. Rather, digital computers can follow any rule that can be given to them in the form of a computer program specifying the relevant steps to be performed. A computer program is a sequence of instructions, which are strings of digits that determine which step the computer is to perform at any given time based on the input it receives and the internal state it's in. In other words, a program spells out a rule for manipulating input digits in light of the computer's internal state.

By executing instructions, digital computers are sometimes said to process instructional information, which is defined in terms of the operations performed by the computer in response to their instructions (see Chapter 10).This sort of information should not be confused with the types of information we discuss below, which are not carried by computer instructions but rather by inputs or data stored in memory.

Programs are written in appropriate programming languages, and some programming languages are more powerful than others because they allow programmers to spell out a larger class of rules over strings of digits. In 1936, logicians Alonzo Church and Alan Turing argued that certain canonical programming languages (such as Turing machines, recursive functions, and the lambda calculus) are adequate for spelling out *any* rule for manipulating digits that can be spelled out at all. Such languages are all computationally equivalent to one another – any rule that can be spelled out within one of them can be spelled out within all of them. The resulting view is the Church-Turing thesis; it says that any digital computation that ~~can be performed at all~~ can be performed in accordance with instructions spelled out within these canonical programming languages. It is a compelling view that has much evidence in its favor and no evidence against (Kleene 1952).

Turing also showed how to design a computing machine that can execute *any* program written within a canonical programming language. Given any arbitrary sequence of instructions written in a canonical programming language, such machines take one instruction at a time, execute the instruction on the relevant data, and then move on to the next instruction. Machines with this property are called *universal machines*. Since ordinary computers can execute any program on any input until they run out of memory or time, they are universal in Turing's sense.

Being a universal machine is sometimes confused with being able to perform *any* digital computation, or any computation whatsoever, or even with *being able to do anything*. Computers cannot perform arbitrary activities simply by computing; for example, they cannot create a sculpture or cook a meal simply by manipulating digits, although they might contribute to creating a sculpture or cooking a meal if their computations drive suitable machines for sculpting or cooking. More significantly, computers can only perform computations for which there are programs. By definition, the only computations for which there are programs are defined over a domain that has at most countably many entities – for instance, strings of letters from a finite alphabet. In addition, the vast majority of functions from strings of digits to strings of digits are such that there is no program for computing them. The best known example of a function for which there is no program is the halting function, i.e. the function that returns, for any program and any input, whether the program will return an output and stop computing or will simply continue to compute forever without ever returning an output. Therefore, universal machines, including ordinary computers, can only compute a small subset of all the functions from strings of digits to strings of digits.

Some authors have speculated that there might be physical systems that can compute something that universal machines cannot compute (e.g., Hogarth 1994). Such hypothetical physical systems are called hypercomputers. Somehow, hypercomputers would be able to compute at least one function for which there is no computer program, such as the halting function. As of yet, there is no compelling evidence that any hypercomputer can be built.

So far, we have focused on ordinary digital computers. There are other kinds of computing systems, such as quantum computers and analog computers. They perform, respectively, *quantum computation* and *analog computation*. Quantum computers manipulate qudits according to an appropriate rule defined over the qudits (plus, possibly, internal states of the quantum computer). Qudits are like digits in that they are defined in terms of finitely many possible states, but are unlike digits in that they can be in a superposition of the different states. (A superposition is a quantum state that is a mixture of two "basis" states; for example, a quantum state composed in part by a 1 and in part by a 0.) Analog computers manipulate continuous variables in accordance with an appropriate rule defined over the continuous variables (plus, possibly, internal states of the analog computer). Continuous variables can take any real value within a certain interval, and in this respect they differ from digits.

Quantum computers can compute the same functions as ordinary digital computers – they are computationally equivalent – but in some cases they can do it more efficiently. Analog computers compute a different sort of function from digital computers, but their computations can be approximated by digital computers by encoding analog inputs as strings of digits and performing digital computations on those.

Artificial neural networks are another important class of computing devices, and they perform many different kinds of computation. They are sets of connected units in which each unit processes a variable in accordance with a rule defined over the variable. Typically, artificial neural networks have units that receive inputs from the environment (input units), units that yield outputs to the environment (output units), and units that communicate only with other units in the system (hidden units). Each unit receives input signals and delivers output signals as a function of its input and current state. As a result of their units' activities and organization, artificial neural networks turn the input received by their input units into the output produced by their output units.

Some artificial neural networks have a fixed structure and process strings of digits in discrete steps. Such networks perform digital computations just like ordinary digital computers. In fact, ordinary digital computers are a special kind of artificial neural networks of this kind – what is special about them is how large and well organized they are. Other artificial neural networks have a structure that changes over time and so is subject to learning; they may also process digits but not by means of discrete steps; rather, they manipulate digits by means of continuous internal processes. Yet other artificial networks manipulate continuous variables, like those manipulated by analog computers.

What of the neural networks to be found in real brains? They process trains of action potentials or spike trains, which are a special kind of signal transmitted from neuron to neuron. Some authors have argued that spike trains are neither strings of digits like those manipulated by digital computers nor continuous variables like those manipulated by analog computers; if this is correct, computation over spike trains is a different kind of computation than either digital or analog computation (Piccinini and Bahar 2013).

To capture all of these kinds of computation, it is useful to introduce a generic notion of computation. Computation in the generic sense is the manipulation of a vehicle according to a rule defined over the vehicle (plus, possibly, the internal states of the system). Depending on which kind of vehicle is manipulated, we obtain digital computation, analog computation, quantum computation, etc.

An important feature of computation is that it is defined solely in terms of specific degrees of freedom of the vehicles that are manipulated, without considering any further physical properties that are specific to the implementing medium. Because of this, the same computation can be implemented in any physical medium (mechanical, hydraulic, electronic, etc.) that possesses the relevant degrees of freedom. In this sense, computation is *medium-independent*. Medium independence entails multiple realizability – there are many ways of realizing the same computation. But multiple realizability does not entail medium-independence – many properties are multiply realizable but medium-dependent, because they are tied to specific physical effects. For example, there are many different ways of producing light, so light production is multiply realizable; but light production is medium-*dependent*, for it always depends on light being produced, which is a specific physical effect.

We now turn to information.

Varieties of information and their relations to computation

How are generic computation and information related? It depends on what sort of information we are talking about (see Chapter 10). In this chapter, we distinguish three notions of information: Claude Shannon's non-semantic notion of information, natural semantic information and non-natural semantic information.

Shannon (1948) was interested in solving the "fundamental problem of communication": reproducing at a receiver's end (e.g., an airplane) a sequence of symbols generated at a source (e.g., a control tower). Shannon showed that there is a way to talk about information transmission between source and receiver without making any assumptions about what the symbols mean, if anything.

The key insight here is that there is something we are informed about when a symbol is selected, namely that all symbols different from it have *not* been selected. Information in Shannon's sense is in effect the reduction of uncertainty generated by the occurrence of one of a set of alternative possible outcomes.

Let us suppose we have an experiment which involves the observation of a random variable X taking values a_1 and a_2 with probabilities p_1=0.9999 and p_2=0.0001 respectively. Before the random experiment takes place, symbol a_1 is almost certain to be selected, and symbol a_2 highly unlikely. The selection of both symbols generates information, since they both resolve the uncertainty characterizing the situation before the experiment takes place.

The selection of a_2 generates *more information* than the selection of a_1, because it is *less expectable*, or *more surprising*, in light of the prior probability distribution. Shannon information is measured in bits, where 1 bit is the information generated by the occurrence of an outcome that has a 50 percent probability of occurring. Any outcome with probability of less than 50 percent will generate *more* than 1 bit, any outcome with probability of more than 50 percent will generate *less* than 1 bit.

Shannon introduced two measures of information and applied them to study the efficient transmission of information across communication channels. The first is *entropy*, an objective measure of the uncertainty characterizing a source as a whole that tells us how surprising the selection of symbols at the source is on average. Zero entropy corresponds to the case in which the selection of a certain symbol has probability 1 and consequently is not surprising at all. Maximum entropy corresponds to the case in which the selection of each symbol is equally probable, in which cases there is maximal uncertainty as to which symbol will be selected.

The second measure Shannon introduced is called *mutual information*. Informally, mutual information is an objective measure of how much we can determine from a variable Y about another variable X. More precisely, mutual information tells us how much the uncertainty characterizing a source X is reduced on average by observing what symbol is selected at the receiver Y. If two variables are statistically independent, then nothing can be known about one by observing the other, and their mutual information is zero. Mutual information is maximal when knowing what symbol is selected at the receiver's end eliminates all uncertainty about what symbol was selected at the source.

Armed with the *non-semantic, objective,* and *quantifiable* notions of entropy and mutual information, Shannon proved a number of seminal theorems, which had a profound impact on the field of communication engineering (see Chapter 10). In the "fundamental theorem for a noisy channel," for example, Shannon proved that it was theoretically possible to make the error rate in the transmission of information across a randomly disturbed channel as low as desired up until the point in which the source information rate in bits per unit of time becomes larger than the channel capacity per unit of time.

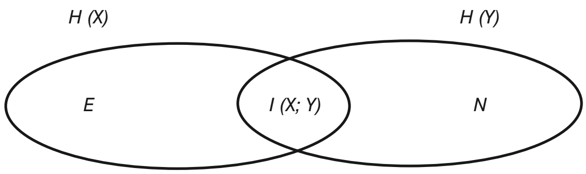

Figure 3.1 The mutual information $I(X;Y)$ is the intersection between the entropy of the source $H(X)$ and the entropy of the receiver $H(Y)$, in the sense that it represents their statistical dependency (it is equal to zero when X and Y are statistically independent). The portion of $H(X)$ which does not lie within $I(X;Y)$ is the equivocation E, the average amount of information generated at X but not received at Y. The portion of $H(Y)$ which does not lie within $I(X;Y)$ is the noise N, the average amount of information received at Y but not generated at X.

The first thing to note is that, strictly speaking, neither the selection of computational vehicles nor their transformation by means of rules needs to generate any Shannon information. This is because Shannon information requires uncertainty, whereas generic computation, in all of its varieties, is in principle compatible with deterministic variables and deterministic rules about their manipulation. In practice, real-world computers show noise at various junctures (e.g. in the communication between components and among computers). Noise is any random disturbance in signal transmission, which may lead to a difference between the received signal and the original signal transmitted by the source. Coding theory, a branch of communication theory focused on the efficient transformation of symbols into signals, can help us design ways of encoding information such that the intended message is retrievable in the presence of noise.

Cognitive scientists and computer scientists are most interested in the relation between computation and semantic information, i.e., information connected to the meaning of information bearers (rather than merely to how surprising they are). Following Grice's (1957) distinction between natural and non-natural meaning, we distinguish between natural (semantic) information and non-natural (semantic) information.

Natural information is the sort of information smoke carries about fire, whereas non-natural information is the sort of information words like "there is smoke" carry about smoke. Natural information has been accounted for in two main ways. Some have proposed that a signal carries natural information about anything it perfectly correlates with in accordance with a law of nature (Dretske 1981). This law-based theory of natural information guarantees that information is veridical: if the information carried by a signal is underwritten by a natural law, a signal carrying natural information that P entails that P. This is consistent with the view, defended by several authors, that information is always veridical (Grice 1957; Dretske 1981; Floridi 2005). One problem with the law-based proposal is that few of the relations between events that are of interest to information processing organisms and artifacts are law-like. For instance, there is no law of nature that guarantees that if a fire had not occurred, smoke would not have occurred, so on the law-based theory we would have to conclude that smoke does not carry information about fire.

Probabilistic theories of natural information replace the nomic requirement with the requirement that signals must reliably correlate with what signals are about (Shea 2007; Skyrms 2010; Scarantino, Forthcoming). On this view, smoke carries natural information about fire because it reliably correlates with fire, in the sense that the presence of smoke significantly raises the probability of fire by virtue of the causal relation between them. An implication of probabilistic analyses of information is that smoke can carry information about the presence of fire, i.e. increase its probability, even when no fire is present. In this sense, natural information understood under a probabilistic theory is non-veridical (Scarantino and Piccinini 2010; but see Floridi 2005, 2010a).

Computational vehicles may or may not carry natural information about features of the external environment. Many ordinary computations operate on vehicles that carry no natural information about the external environment. For instance, ordinary mathematical calculations carried out on a computer are defined over numbers, regardless of what the digits being manipulated by the computer reliably correlate with. There are also many examples of natural information processing within computing systems. Any inputs to a computer that reliably correlate with some environmental variable carry natural semantic information. A computer may process such inputs to make the information more usable or accessible, to combine it with other information (either from memory or other sensors), or to perform control functions. For example, airplane computers use natural semantic information about the state of the airplane to regulate fuel injection, altitude, speed, and so on.

The processing of natural information must be carried out by means of computations, because natural information is a medium-independent notion. This is because whether natural information can be associated with a given vehicle does not depend on its specific physical properties, but on how it changes the probabilities of the state of affairs it is about (Scarantino, Forthcoming). Since generic computation is just the processing of medium-independent vehicles, any processing of natural information amounts to a computation in the generic sense.

Bearers of non-natural information, unlike bearers of natural information, need not correlate reliably with what they are about. For instance, the words "there is smoke" non-naturally mean that there is smoke whether or not they correlate with the presence of smoke. Thus, there must be an alternative grounding process by which bearers of non-natural information come to bear non-natural information. A convention, as in the case of the meaning of English words, is a clear example of what may establish a non-natural informational link.

Non-natural information need not be based on convention. There may be other processes, such as biological evolution, through which non-natural informational links may be established. A paradigmatic example is offered by vervet monkey alarm calls, which "functionally refer" to different classes of predators despite the fact that no convention has associated the calls with the relevant predators (Seyfarth *et al.* 1980; Scarantino and Clay, Forthcoming). What matters for something to bear non-natural information is simply that it comes to stand for something else relative to a signal recipient. Once this link is established, the bearer of information *represents* a certain state of affairs, and can do so successfully or unsuccessfully. The possibility of error is the key difference between natural information and non-natural information (or representation).

An important corollary of the distinction between natural and non-natural semantic information is that semantic information of the non-natural variety can be true or false. The statement "water is not transparent," for instance, contains false non-natural information to the effect that water is not transparent. Some authors take false non-natural information to be a genuine kind of information, even though it is epistemically inferior to true information. According to such authors, consumers of information value non-natural information insofar as it is true, but denying that false non-natural information is information too is not advisable, because it prevents us from capturing a large swath of information usages within the sciences (see Scarantino and Piccinini 2010, with response by Floridi 2010b).

Most generic computations process vehicles that carry non-natural information, which may be encoded in the inputs or memory states of a computer. A computer may process data carrying non-natural semantic information to make the information more usable or accessible, to combine it with other information (either from memory or other sensors), or to perform control functions. But a computer can also process vehicles that do not carry any non-natural information, as when computers put lists of meaningless words in alphabetical order. Although the words do not "represent" anything, the computation proceeds just the same. From this it follows that a generic computation may or may not involve the processing of non-natural information. The converse does not hold, in the sense that whenever non-natural information is processed, a generic computation takes place, because bearers of non-natural information carry the information they do by virtue of what they stand for and independently of the physical characteristics of their vehicles, and are in this sense medium-independent.

Computation and information processing are commonly used interchangeably, which presupposes that they are roughly synonymous terms with univocal meanings. As we have seen, this is not the case. There are multiple notions of computation and multiple notions of information, and the relations between them are complex. The claim that a system is processing information may mean nothing more than that the system can be analyzed in terms of its Shannon-information properties. Or it may mean that it process semantic information, of either the natural or the non-natural variety. Furthermore, we have argued that computation does not presuppose information processing. Even though the computations performed by natural and artificial computing systems generally involve the processing of information, computations can at least in principle operate on vehicles that do not carry any type of information.

References

Dretske, F. (1981). *Knowledge and the Flow of Information*. Oxford, Blackwell.

Floridi, L. (2005). "Is Semantic Information Meaningful Data?," *Philosophy and Phenomenological Research,* 70.2, 351–370.

Floridi, L. (2010a). "Semantic Information and the Correctness Theory of Truth," *Erkenntnis*, 74.2, 147–175.

Floridi L. (2010b). "The Philosophy of Information: Ten Years Later," *Metaphilosophy*, 41.3, 402–419.

Grice, H.P. (1957). "Meaning," *Philosophical Review*, *66*, 377–388.

Hogarth, M. L., (1994). "Non-Turing Computers and Non-Turing Computability," *PSA* 1994(1): 126–138.

Kleene, S. C. (1952). *Introduction to Metamathematics*. Princeton, NJ, Van Nostrand.

Piccinini, G. and S. Bahar (2013). "Neural Computation and the Computational Theory of Cognition," Cognitive Science, 34, 453–488.

Piccinini, G. and A. Scarantino (2011). "Information Processing, Computation, and Cognition," *Journal of Biological Physics, 37*(1), 1–38.

Scarantino, A. (Forthcoming). "Information as a Probabilistic Difference Maker," *Australasian Journal of Philosophy*.

Scarantino, A. and Clay, Z. (Forthcoming). "Contextually Variable Signals can be Functionally Referential," *Animal Behaviour*.

Scarantino, A. and Piccinini, G. (2010). "Information without truth," *Metaphilosophy*, *41*, 313–330.

Shannon, C.E. (1948). "A Mathematical Theory of Communication," *Bell System Technical Journal*, 379–423, 623–656.

Seyfarth, R.M., Cheney, D.L., and Marler, P. (1980). "Vervet Monkey Alarm Calls. Semantic Communication in a Free-ranging Primate," *Animal Behaviour*, *28*, 1070–1094.

Shea, N. (2007). "Consumers Need Information: Supplementing Teleosemantics with an Input Condition," *Philosophy and Phenomenological Research, 75,* 404–435.

Skyrms, B. (2010). *Signals: Evolution, Learning, and Information.* New York: Oxford University Press