

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

University of Wollongong
Research Online

Faculty of Engineering and Information Sciences - Papers: Part A

Faculty of Engineering and Information Sciences

2011

Digital Computation as Information Processing

Nir Fresco
University of New South Wales, nfresco@uow.edu.au

Publication Details

Fresco, N. (2011). Digital Computation as Information Processing. AISB Convention (Society for the Study of Artificial Intelligence and Simulation of Behaviour) (pp. 58-64). York, United Kingdom: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Digital Computation as Information Processing

Abstract
It is common in cognitive science to equate computation (in particular digital computation) with information processing. Yet, it is hard to find a comprehensive explicit account of concrete digital computation in information processing terms. An Information Processing account seems like a natural candidate to explain digital computation. After all, digital computers traffic in data. But when 'information' comes under scrutiny, this account becomes a less obvious candidate. 'Information' may be interpreted semantically or nonsemantically, and its interpretation has direct implications for Information Processing as an objective account of digital computation. This paper deals with the implications of these interpretations for explaining concrete digital computation in terms of information processing. To begin with, I survey Shannon's classic theory of information, and then examine how 'information' is used in computer science. In the subsequent section, I evaluate the implications of how 'information' is interpreted for an Information Processing account. The key requirements for a physical system to compute are then fleshed out, as well as some of the limitations of such an account. Any Information Processing account must embrace an algorithm- theoretic apparatus to be a plausible candidate for explaining concrete digital computation.

Keywords
processing, digital, information, computation

Disciplines
Engineering | Science and Technology Studies

Publication Details
Fresco, N. (2011). Digital Computation as Information Processing. AISB Convention (Society for the Study of Artificial Intelligence and Simulation of Behaviour) (pp. 58-64). York, United Kingdom: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

This conference paper is available at Research Online: http://ro.uow.edu.au/eispapers/1726

Digital Computation as Information Processing

Nir Fresco (student)[1]

Abstract. It is common in cognitive science to equate computation (in particular digital computation) with information processing. Yet, it is hard to find a comprehensive explicit account of concrete digital computation in information processing terms. An Information Processing account seems like a natural candidate to explain digital computation. After all, digital computers traffic in data. But when 'information' comes under scrutiny, this account becomes a less obvious candidate.

'Information' may be interpreted semantically or non-semantically, and its interpretation has direct implications for Information Processing as an objective account of digital computation. This paper deals with the implications of these interpretations for explaining concrete digital computation in terms of information processing. To begin with, I survey Shannon's classic theory of information, and then examine how 'information' is used in computer science. In the subsequent section, I evaluate the implications of how 'information' is interpreted for an Information Processing account. The key requirements for a physical system to compute are then fleshed out, as well as some of the limitations of such an account. Any Information Processing account must embrace an algorithm-theoretic apparatus to be a plausible candidate for explaining concrete digital computation.

1. INTRODUCTION

According to the Information Processing (hereafter, IP) account, for a system to be deemed a computing system it needs to process data, which carries information. It is often assumed, particularly in cognitive science discourse, that symbolic computation models can freely be described as information processing models. This is the motivation for this paper, which deals with the question whether concrete digital computation (i.e., digital computation as it is actualised in *physical systems*) can be adequately explained solely in information processing terms.

Furthermore, any resulting IP account hinges on the interpretation of 'information'. It can be interpreted semantically or non-semantically (and more specifically quantitatively as 'Shannon information' or Algorithmic Information). It is questionable whether an IP account of computation must presuppose semantic information. The important question is then whether computing systems traffic in semantic information inherently, or whether they traffic in data or non-semantic information, which in turn could be assigned some meaning by users[2].

A general account of Information *Processing* based on 'Shannon information' is outlined here in the context of concrete digital computation. I begin by surveying Shannon's classic theory of information in section 2, and then examine whether Algorithmic Information theory significantly changes the resulting IP account, in the third section. Subsequently, in section 4, I discuss the implications of how 'information' is interpreted for explaining concrete digital computation. The key requirements implied by the IP account for a physical system to perform digital computation are explicated in section 5. Eventually, in section 6, I examine the limitations of this account and argue that any IP account must embrace an algorithm-theoretic apparatus to be a plausible candidate for explaining concrete digital computation.

2. THE RECEIVED THEORY OF INFORMATION IN COMMUNICATION

The most influential theory of information in communication and engineering was introduced by Claude Shannon in 1948. He showed how information could be transmitted efficiently across communication channels by means of encoded messages. Shannon [1] attempted to solve the "fundamental problem of communication": finding the optimal manner by which messages from a source of information are exactly or approximately reproduced at their destination [2]. According to Norbert Wiener [3], one of the simplest unitary forms of information is the recording of a choice between two equiprobable basic alternatives. A sufficient condition for a physical system to be deemed a sender or receiver of information is the production of a sequence of symbols in a probabilistic manner.

Moreover, Shannon [1] and Wiener [3] analyse an information-generating system in terms of five essential components: an information source, a transmitter, a channel, a receiver and a destination. The information source produces a message to be communicated to the receiver. The transmitter operates on the message to produce a signal suitable for transmission over the channel, which is simply the medium of signal transmission. The receiver reconstructs the message from the signal. And the destination is the system for which the message is intended. So communication amounts to the source of information producing a sequence of symbols, which is then reproduced by the receiver to some degree of accuracy.

Nevertheless, 'Shannon information' does not entail any semantic content or meaning. Shannon's information theory approaches information syntactically as a physical

[1] School of History and Philosophy, University of New South Wales, Sydney, Australia. Email: *Fresco.Nir@Gmail.com*

[2] I do not mean to imply here (as John Searle would) that the idea of concrete computation requires something like a knower or an observer. A computing system will continue computing even in our absence, as long as it does not break and has a constant supply of energy to run.

phenomenon: whether and how much (not what) information is conveyed [4]. On the other hand, a stronger sense of 'information' (i.e., semantic information) entails that messages have specific meanings by representing how things are or could be.

According to Shannon's theory, 'information' is interpreted in the weaker sense. 'Shannon information' is different (but not distinct) from the ordinary usage of 'information'; it tells us nothing about the usefulness of or interest in a message. The basic idea is coding messages into a binary (or any other) system at the bare minimum of bits we need to send to get our message across. Even in this limited sense, the amount of 'information' conveyed is as much a property of our own knowledge as anything in the message. If we send the same message twice every time (a message and its copy), the information in the two messages is not the sum of that in each. Rather the information only comes from the first one (assuming it was successfully transmitted) [5].

The important aspect of 'Shannon information' is that the message is selected from a set of possible messages. A message, composed of symbols, is a physical structure discriminated by the probability of selecting it over other possible messages. So a nonsensical message composed of the sequence of symbols '%3-4Y7@*' could in essence generate more information than a meaningful message (in the ordinary use of 'information') such as 'daughter' in reply to some question. This could be the case, if the message '%3-4Y7@*' would be more "surprising" than 'daughter' [2]. Receiving a message containing the former string could change the recipient's circumstance from not knowing what *something was* to knowing what *it is*. The more possible messages a recipient could have otherwise received, the more "surprised" the recipient is when it gets that particular message [5].

3. INFORMATION IN COMPUTER SCIENCE

Shannon's theory of information is the reigning theory in communication and can adequately explain network communication between computers, encoding and decoding of messages, message transmission through data buses or network cables and so on. Algorithmic Information theory, which was introduced by Andrei Kolmogorov, Ray Solomonoff and Gregory Chaitin, deals with the complexity of data structures and can be described as the borderland where information and digital computation meet. It formally defines the *complexity* or the *informational content* of a data structure (e.g., a string) as the length of its shortest self-delimiting algorithm running on a Universal Turing Machine (henceforth, UTM) [6] [7]. The algorithmic information of any computable string is the length of the shortest algorithm that computes it on a UTM.

Moreover, Chaitin proposes to think of a computing system as a decoding device at the receiving end of a noiseless binary communications channel [6]. Its programs are thought of as code words, and the result of the computation (i.e., its output) as the decoded message. The programs then form what is called a "prefix-free" set so that successive messages (e.g., procedures) sent across the channel can be distinguished from one another. Still he acknowledges that Algorithmic Information has precisely the formal properties of Shannon's concept of information entropy.

Although Algorithmic Information is underpinned by classical computability theory, it too is non-semantic and quantitative as Shannon's theory. It interprets information and measures its quantities in terms of the computational resources that are needed to specify it [4]. Algorithmic Information measures the length of the shortest algorithm that computes a string. That algorithm may even be shorter than the output it produces. For example, representing π or e in a binary notation [7].

Algorithmic Information then may be deemed a competing notion of 'Shannon information' by allowing us to assign complexity values to individual strings and other data types [8]. Whilst Shannon's theory analyses the amount of information in a group of messages based on the probability of the messages, Algorithmic Information theory analyses the complexity of a string as a single message. The relative frequency of the message has no meaning, but there is some shortest program on a UTM that can produce this message. The length of this optimal program is an absolute measure for the amount of information in that message. Both Algorithmic Information and 'Shannon information' give rise to optimal compression codes for information. A bit string '01010101010101' can be compressed in Shannon's sense as *'01'=1;1111111*, or can be programmed in the Algorithmic Information sense as *for x = 1 to 7 write '01'* [8].

4. THE IMPLICATIONS OF INTERPRETING 'INFORMATION' FOR THE IP ACCOUNT

Most structural accounts of concrete computation assume that it is a type of information processing. These accounts consider the unique structural properties of computing systems (namely, their digital architectures) to be their distinctive feature. But in addition to the structural constraint, they also assume a semantic constraint on concrete computation: processing information [9]. Some connectionists, on the other hand, reject the structural constraint, and argue that information-processing properties of digital computation differentiate it from other causal and mechanical processes. On their view, concrete computation is explained in terms of the information transformed, represented, and stored in the process of computing [10].

Still, any current attempt to untangle concrete computation and IP must begin with a distinction between a weaker sense of 'information' (e.g., 'Shannon information' or Algorithmic Information) and semantic information. Using informational language in the stronger sense raises some problems in regard to concrete computation (e.g., does a computer process semantic information even in the absence of its user? is the meaning of the computer-processed information intrinsic? if some

information processing carries meaning and some does not, how are they distinguished? etc.).

Even if we interpreted the IP account in Shannon's sense, it would be hard to accept it as a satisfactory account of concrete digital computation. Brian Cantwell Smith [11] argues that since information theory is not a full analysis of information, it cannot be a solid basis for a comprehensive account of concrete computation. His argument relies on a semantic reading of both information [11] and computation [12], and if 'Shannon information' is not semantic, it cannot adequately explain computation. Moreover, Piccinini and Scarantino [2] maintain that it is not clear how 'Shannon information' is *processed*. Whether 'Shannon information' can be associated with a given vehicle does not depend on any specific physical properties. Instead, it is regarded as a selection of symbols from a given language according to the probability distribution of these symbols. Shannon information does not pertain to individual messages, and individual messages are those that may be created, and manipulated by digital computing systems.

Also, digital computation may be either deterministic or non-deterministic (e.g., probabilistic computation, random computation etc.). Still, most digital computing systems developed in the computer industry are deterministic, since their behaviour is repeatable and systematic. A dry run of a deterministic algorithm (using some test data) should systematically yield the same output when its input and initial state remain unchanged. The state-transitions of Shannon's communication model are probabilistic, whereas the transition probabilities of a Turing machine (hereafter, TM) are all set to 1. For every possible input, there is only one possible state into which the TM transitions [13].

Be that as it may, analysing digital computation using information-theoretic language could be very constructive. Smith [11] asserts that the IP account could indeed serve as the grounds for a plausibly comprehensive theory of concrete computation. But on his view several theoretical issues must first be addressed. Firstly, only a semantic theory of information stands a chance of doing justice to computation. Secondly, 'information' must be analysed in a manner that does not entail pan-informationalism. Otherwise, this could lead to a dangerous equivocation: if any object can be described in informational terms, then the nature of all objects is genuinely informational. At least prima facie, concrete digital computation being driven by the executed software seems most likely amenable to instructional information (e.g., do X if Y, otherwise halt) [14].

5. THE KEY REQUIREMENTS IMPLIED BY THE IP ACCOUNT

The key requirements for a physical system to perform digital computation implied by the IP account are fourfold: 1. having the capacity to send information, 2. having the capacity to receive information, 3. having the capacity to store and retrieve information, 4. having the capacity to process (or transform) information. The choice between a semantic and a non-semantic reading of 'information' affects both the characterisation of the resulting IP account and the key requirements it implies for computing systems, as will be shown below. The following discussion remains neutral on the semantic vs. non-semantic reading of 'information', unless specified otherwise.

The first key requirement implied by the IP account is the system having the capacity to send information. Whether 'information' is interpreted semantically or non-semantically, concrete computation requires a source of information to transmit the data. Regardless of the medium by which the data is transmitted (e.g., via data buses, network cables, etc.), the sender is responsible for the data transmission. The sender prepares the messages to be sent to the receiver and encodes them for transmission. To emphasise, in computing systems the sender and the source of information may be distinct entities. For instance, whilst the computer's main memory could be a source of information (e.g., a stored instruction), the memory controller is responsible for fetching the data from memory and transmitting it to the CPU. The memory controller acts as the sender, but not as the source.

Analogously, the second key requirement implied by the IP account is the system having the capacity to receive information. If the former requirement necessitated a sender to transmit the message, this requirement necessitates a receiver on the other end to accept it. The absence of a receiver on the other end means that the computation remains unexecuted (or in a suspend mode). For instance, an instruction, which was fetched by the main control unit from the memory, but not received by the receiver (the CPU in this case), will not be executed. Also, a computer program (the sender), which sends an input/output signal to the operating system (the receiver), will enter the suspend mode until its I/O request is acknowledged.

Furthermore, when construed as information processing, concrete digital computation is at best incomplete in the absence of either a receiver or a sender. Unlike a microphone acting as a sender of information even in the absence of a receiver (the audience), a computing system must have both a sender and a receiver that are well coordinated. The information contained in a message may indeed not depend on the receiver's learning something from it, or even being able to decode the message [15]. But if the receiver is absent or unable to decode the message in a computing system, the computation will be either incomplete or incorrect. Suppose that the CPU (the receiver) is unable to correctly decode the instruction from the main control unit (the sender), it will fail to execute the instruction hindering the overall computation.

The third key requirement implied by the IP account is the system having the capacity to store and retrieve information. Computing systems store and retrieve digital information, which can be thought of as a series of bits. The storage and retrieval of information in a computing system should be well synchronised, as one always presupposes the other. Without the system having the ability to retrieve the data, there is clearly very little sense to storing it in the first place.

Lastly, the fourth key requirement implied by the IP account is the system having the capacity to transform information. This requirement cannot be dismissed, as it the essence of processing of information. It is also the most problematic requirement, which becomes even more stringent when 'information' is interpreted semantically (as will be shown below). It is important to emphasise that transforming information does not amount to merely encoding and decoding information. Those are methods that typically preserve the information while converting it into a coded form and vice versa, and are useful in the communication of signals or messages. Transformation of information is more than that, it is characterised as the creation (e.g., a new database table containing salaries of employees), modification (e.g., giving some employees a pay rise) and destruction of information (e.g., deleting some records of employees, who left the company, from the system).

The transformation or processing of 'Shannon information' is problematic, because its focus is not on the content of individual messages. To restate Wiener's claim [3], a sufficient condition for a physical system to be deemed a sender or receiver of 'Shannon information' is the production of messages in a probabilistic manner. Processing 'Shannon information' can be the modification of the state or strings states that may result in changes of the conditional entropies among the states. It can also be the elimination of possibilities (reduction in uncertainty) represented by a signal or the introduction of redundancy to offset the impact of noise and equivocation. But again, sending the same message twice (to offset the impact of noise) does not yield information that is the sum of that in each. Similarly, elimination of redundancy does not reduce the underlying informational content that is conveyed.

Moreover, construing concrete digital computation as information processing requires more than merely communicating information in a non-deterministic manner. Telephones (not the *voice over internet protocol* systems) are information processing systems, but they are not digital computers [13]. They can be used to transmit information, but they certainly do not compute in any non-trivial sense. Computers do indeed encode, decode and transmit information, but they also perform tasks with inferential import (when I try to divide a number by 0, a good algorithm should yield an error message from the computing system). Yet, this requires a way of distinguishing the differences between the informational *contents* of the messages. Shannon's information provides the procedures for selecting messages, but it lacks this capacity [15].

Still, this ability to distinguish between different contents is necessary for modifying or adding new justified information. Shannon's information theory tells us about the probabilities associated with symbols from a given language, but it is indifferent to the content of the messages. For instance, the strings S_1 and S_2 have the same length (including that of their symbol constituents), but a different composition of symbol constituents. S_1= "All cars have four wheels"; S_2= "All cats have four ankles". Let us suppose that S_1 and S_2 are equiprobable (so in Shannon's sense, they are both potentially equally informative). Let S_3 be "Bumblebee is a car". By using Universal Instantiation, for example, one can infer some new justified information[3]: S_4= "Bumblebee has four wheels" from S_1 and S_3. This new information must also be true, if S_1 and S_3 are true (here enters semantic information again). It tells us something else about Bumblebee (that Bumblebee has four wheels).

On the other hand, S_2 and S_3 do not yield new justified information using Universal Instantiation (similar to S_4). One cannot validly infer any new singular statement about Bumblebee from the universal statement S_2, as Universal Instantiation does not apply to S_3 (for Bumblebee is not a cat). In order to apply rules of logic as a means of generating new true information, the symbolic constituents of strings must be distinguishable. But according to Shannon's information theory we may encode and transmit S_2 (rather than S_1) and S_3 to the recipient (since S_1 and S_2 are equiprobable). Yet, the recipient will have learned nothing new from S_2 and S_3 in this case.

Furthermore, when 'information' is construed semantically (as proposed by some philosophers) its transformation requirement becomes even more stringent. The syntactical manipulation of messages must be done in a manner that always preserves their semantics. Typically, rules that are applied in the transformation process must be truth preserving[4]. At the very least, new justified information has to be consistent with prior "known" information. If conjunction, for instance, is applied to add new justified information, then the conjuncts C_1 and C_2 must be neither contradictories nor contraries. Otherwise, their conjunction 'C_1 and C_2' would be false.

Likewise, when syllogistic rules are applied in the process of transforming semantic information, syllogistic fallacies must be prevented[5]. For instance, when deductive inference is used to validly infer P_3 from the premises P_1 and P_2 (where $P_1 \neq P_3$ and $P_2 \neq P_3$), then P_3 must be true to be deemed new (or modified) semantic information. Not only that, but the error detection mechanism employed by the computing system must be such that it verifies that every single premise (P_1 and P_2, in the example above) is true[6], even if the deductive argument is valid. Thus, to extract new semantic information, sufficient scrutiny is required to ensure its truth and coherence. The same principle also applies to other types of

[3] There is an ongoing debate regarding information in deductive inferences. Some, including John S. Mill and the logical positivists, have argued that logical truths are tautologies, and so deductive reasoning does not add any new information.

[4] Induction, abduction and non-monotonic logic do not abide by the same principle, and their application does not guarantee the truth of any new information that they potentially create. Both abductive reasoning and non-monotonic logic play an important role in artificial intelligence and should not be discounted, but they exceed the scope of this paper.

[5] In particular, when interpreting semantic information as being necessarily true [4] [15] [16].

[6] Immediate inferences from categorical propositions, for instance, do not require the same error verification mechanism. From the categorical proposition 'no dogs are cats', we can immediately infer that 'no cats are dogs' by swapping the predicate term and subject term of the original proposition. The truth of one of them guarantees the truth of the other.

transformation rules like existential generalisations, universal instantiations, inductive inferences and so on.

Still, digital computation will proceed (or fail) regardless of the truth-value of the information processed by the computing system. Gricean non-natural meaning of signs (e.g., three dings of the bus bell indicating that the bus is full) does not require a correspondence to the state of affairs in question (e.g., whether the bus is actually full). In a similar manner, non-natural information may be processed by the computing system without any correspondence to an external state of affairs. There is always a possibility that a computing system will produce an incorrect output as a result of a miscomputation[7] (i.e., a mistake in the computation process due to an error in the executed algorithm or a hardware malfunction). In that case, the only viable option is that the miscomputation misrepresents the state of affairs in question[8]. But from the system's "point of view", this wrong output has no less (or more) meaning than the correct output (which might correctly correspond to some state of affairs). Whether a computation represents some state of affairs or not is a contingent fact.

6. SOME LIMITATIONS OF THE IP ACCOUNT

When 'information' is interpreted narrowly the resulting IP account cannot adequately explain *how* computation is executed and how it differs from *miscomputation*. Any plausible account of concrete digital computation must be able to explain Turing computability, for it lays the ground rules for all existing digital computers as well as for programming languages. Any account of Turing computability has to at least be able to explain the three key algorithmic notions of input, output, and procedures. But an IP account of concrete computation, which is based on 'Shannon Information' or Algorithmic Information, does not adequately explain those three key notions.

'Shannon information', for one, only makes sense in the context of a set of potential messages that are communicated between a sender and a receiver and a probability distribution over this set [8]. There is no room for a probabilistic selection of messages in describing deterministic procedures. There must be a *specific set* of messages that are selected, encoded and transmitted in the same order in accordance with the specific steps of the procedure, regardless of the probabilities associated with each message (or its symbol constituents).

Whilst Algorithmic Information is indeed based on TMs, it is still insufficient as an IP account of concrete computation. Algorithmic Information theory's interest in TMs is limited to finding the shortest program that runs on a UTM and generates a particular string as its output. But the purpose of Algorithmic Information theory is simply to measure the amount of information conveyed by that string or its complexity, rather than being *about that program.* So the best one could hope for in relation to Algorithmic Information explaining a particular procedure is either measuring the information conveyed by that procedure (as a string) or determining whether it is the shortest one for generating the output it produces. Yet, many computer programs running on multitudes of different systems are neither the shortest nor the most efficient for achieving their tasks.

Another challenge for an IP account of concrete computation, which is based on 'Shannon information' or Algorithmic Information, is identifying miscomputations. Miscomputations that are the result of a hardware malfunction could be explained by some breakdown of the communication channel, for example. But other miscomputations resulting from errors by design or a malformed algorithm cannot be easily explained, since neither Shannon's information theory nor Algorithmic Information distinguishes messages by their *contents*.

Arguably, when information is interpreted semantically it must yield knowledge [4] [7] [11] [15] and that implies a further requirement for a semantic IP account of concrete computation. This additional requirement is that by processing information the computing system has to yield knowledge, which is either derived from its user (or programmer or interpreter) or intrinsic to the system. Plato defined knowledge as a true justified belief (which was widely accepted in modern philosophy[9]). Semantic information must tell us something true about some state of affairs, that is, yield knowledge. One option then is that this knowledge is derivative and used by the knower, who uses the information produced by the computing system[10]. Another option is that this knowledge is intrinsic to the computing system that traffics in information.

The latter option has been challenged by many philosophers [18] [19] [20] [21] [22] [23] and it is not at all clear that there is compelling evidence to support it. There is only a limited sense in which a digital computing system "understands" or "knows" something. A digital computer understands the semantics of its machine language. This understanding can be attributed to structural properties of the machine's architecture and language as well as causal links between bit patterns, memory addresses, primitive operations etc. Computers manipulate information that they need not understand, although they copy it, compare it with other information and change it

[7] Besides these two types of syntactic miscomputation, there is also the possibility of a semantic miscomputation relative to some task domain (e.g., the Roomba indoor cleaning robot that may malfunction eventually when it operates under abnormal operating conditions such as an airfield). However, this semantic miscomputation can also be reduced to either one of the syntactic miscomputations above.

[8] There is always also the remote possibility of a double negation. Suppose that the computing system did not correctly represent some state of affairs when the computation was initiated. But then the system's miscomputation incidentally results in a correct representation of that state of affairs.

[9] Edmund Gettier [17] has challenged Plato's widely accepted view of knowledge as Justified True Belief. He argued that truth, belief, and justification are not sufficient conditions for knowledge. He showed that a true belief might be justified but fail to be knowledge, because the belief might be true by sheer accident.

[10] Indeed, this option is no more problematic than an encyclopaedia yielding knowledge for its readers.

[24]. This is the basis for an internal meaning of its information processing.

But that does not imply that the computer manifests any beliefs that are associated with these operations. Suppose we replace the doorbell with a digital computer that emits the sounds: “someone is at the door”, only when someone pushes the door button. When someone pushes the door button, the computer picks up the information about it, processes it and delivers an output. However, this output is not a belief in someone being at the door, anymore than the doorbell would have believed that [15]. Roy Sorensen [25] makes a further distinction between information conveyed by assertions and displays. When an answering machine utters the sounds: "Mr. Smith is not at home", it simply displays this message, rather than assert it. The machine does not believe that Mr. Smith is not at home (he may even be home). Similarly, when a computer weather program displays a rainy weather forecast for tomorrow, it does not believe that it will rain tomorrow, although this output may be based on a reliable source of information. There is no intrinsic belief or knowledge in these information-processing systems.

7. CONCLUSION

Although the IP account, on the face of it, seems like a natural and promising candidate for explaining concrete digital computation, it is less than obvious. This is to a large extent dependent on how information is interpreted and what the resulting IP account is. Some argue that an adequate theory of information must give an account of information as semantic content [15] [4]. Smith [11] asserts that even on a semantic interpretation of information, the IP account is still inadequate, because information, which depends on a counterfactual correlation with the world, is objective. But without further restriction this account leads to pan-informationalism. And indeed an IP account, which leads to pan-informationalism, is not falsifiable and incapable of non-trivially explaining concrete digital computation.

Nevertheless, the IP account must embrace an algorithm-theoretic apparatus to be deemed adequate for explaining computation. As I have argued above, it is the processing part of IP that is very problematic. Arguably, even Algorithmic Information in the form of Kolmogorov-Chaitin-Solomonoff complexity will not do the trick for the IP account. Though it interprets information in terms of the *computational resources* needed to specify that information, it is a measurement method (i.e., analysing complexities, probabilities and randomness), rather than a descriptive one (e.g., explaining whether a miscomputation has just occurred).

All the above suggests that to be a plausible candidate for explaining concrete digital computation the IP account needs improving. If we want to explain certain cognitive functions computationally in terms of information processing, we should first be clear on how concrete digital computation proper is explained non-trivially in information processing terms. An IP account of computation should explain how a computing system is different from other non computing IP systems such as telephones or radios.

ACKNOWLEDGEMENTS

Thanks to Gualtiero Piccinini and Oron Shagrir for useful comments on earlier drafts of this paper. I would also like to thank some anonymous referees for their comments that helped improve this paper. I am especially grateful to Phillip Staines for his detailed comments and ongoing support.

REFERENCES

[1] Shannon, C. E. (1948). A mathematical theory of communication. Mobile Computing and Communications Review, 5, pp. 1-55.

[2] Piccinini, G. and Scarantino, A. (2010). Information processing, computation, and cognition. Journal of Biological Physics, 37, pp. 1-38.

[3] Wiener, N. (1948). Cybernetics: or control and communication in the animal and the machine. Cambridge: MIT Press.

[4] Floridi, L. (2008). Trends in the philosophy of information. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 113-131. Elsevier.

[5] Feynman, R. P. (1996). The Feynman lectures on computation. Reading, MA: Addison-Wesley.

[6] Chaitin, G. J. (2003), Algorithmic Information Theory. Third Printing. Cambridge University Press.

[7] Dunn, M. (2008). Information in computer science. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 581-608. Elsevier.

[8] Adriaans, P. (2008). Learning and the cooperative computational universe. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 133-167. Elsevier.

[9] Shagrir, O. (2010). Computation, San Diego style. Philosophy of Science, 77, pp. 862-874.

[10] Churchland, P.S., Koch, C., and Sejnowski T.J. (1990). What is computational neuroscience? In E.L. Schwartz (Ed.), Computational neuroscience. pp. 46-55. MIT Press.

[11] Smith, B. C. (unpublished). Construals. volume 1, chapter 2. To be published at http://www.ageofsignificance.org

[12] Smith, B. C. (2002). The foundations of computing. In M. Scheutz (ed.) Computationalism: new directions. Cambridge, MA: MIT Press, pp. 23–58.

[13] Broderick, P. B. (2004). On communication and computation. Minds and Machines, 14, pp. 1–19.

[14] Floridi, L. (2009). Philosophical conceptions of information. In G. Sommaruga (Ed.). Formal theories of information, pp. 13-53. Berlin: Springer-Verlag.

[15] Dretske, F. I. (1981). Knowledge and the flow of information. Cambridge, MA: MIT Press.

[16] Millikan, R. G. (2004). Varieties of meaning: the 2002 Jean Nicod lectures. Cambridge, MA: MIT Press.

[17] Gettier, E. L. (1963). Is justified true belief knowledge? Analysis, 23, pp. 121-123.

[18] Agassi, J. (1988). Winter 1988 Daedalus. SIGArt newsletter, 105, pp. 15–22.

[19] Agassi, J. (2003). Newell's list. Commentary on Anderson, J. & Lebiere, C.: The Newell Test for a theory of cognition. Behavioural and brain sciences, 26, pp. 601-602.
[20] Dretske, F. I. (1993). Can intelligence be artificial? Philosophical studies, 71, pp. 201-216.
[21] Dreyfus, H. L. (1979). What computers can't do: the limits of artificial intelligence. 2nd edition. NY: Harper & Row.
[22] Harnad, S. (1990). The symbol grounding problem. Physica, 42, pp. 335-346.
[23] Penrose, R. (1989). The emperor's new mind. London: Oxford university press.
[24] Sloman, A. (1999). Beyond Turing equivalence. In P. Millican and A. Clark (Eds.) Machines and Thought: the legacy of Alan Turing. Oxford University Press.
[25] Sorensen, R. (2007). Can the dead speak? In S. Nuccentelli and G. Seay (Eds.) Themes from G. E. Moore: new essays in epistemology and ethics. New York: Oxford University Press.