Theoretical Computer Science 410 (2009) 376–383

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

What does it mean to say that a physical system implements a computation?[☆]

James Ladyman

Department of Philosophy, University of Bristol, 9 Woodland Road, Bristol, BS81TB, United Kingdom

ARTICLE INFO

Keywords:
Implementation
Realism about computation
Landauer's Principle
L-machine

ABSTRACT

When we are concerned with the logical form of a computation and its formal properties, then it can be theoretically described in terms of mathematical and logical functions and relations between abstract entities. However, actual computation is realised by some physical process, and the latter is of course subject to physical laws and the laws of thermodynamics in particular. An issue that has been the subject of much controversy is that of whether or not there are any systematic connections between the logical properties of computations considered abstractly and the thermodynamical properties of their concrete physical realizations. Landauer [R. Landauer, Irreversibility and heat generation in the computing process, IBM Journal of Research and Development 5 (1961) 183–191. Reprinted in Leff and Rex (1990)] proposed such a general connection, known as Landauer's Principle. To resolve this matter an analysis of the notion of the implementation of a computation by a physical system is clearly required. Another issue that calls for an analysis of implementation is that of realism about computation. The account of implementation presented here is based on the notion of an *L-machine*. This is a hybrid physical-logical entity that combines a physical device, a specification of which physical states of that device correspond to various logical states, and an evolution of that device which corresponds to the logical transformation *L*. The most general form of Landauer's Principle can be precisely stated in terms of *L*-machines, namely that the logical irreversibility of *L* implies the thermodynamic irreversibility of every corresponding *L*-machine.

1. Introduction

The idea of computation bridges the abstract and the concrete worlds. For example, suppose somebody claims: 'The computation took ten minutes'. He or she is clearly referring to a concrete property of a particular process in space and time. On the other hand, suppose somebody claims: 'The computation is logically irreversible'. He or she is clearly referring to a general abstract notion of the computation and an abstract property of it, namely that knowledge of the computation and its output is not sufficient for knowledge of its input. Of course, often claims about computation are made in terms of the capacities of Turing machines. The latter are themselves hybrids of the abstract and the concrete, on the one hand sometimes being construed as purely abstract mathematical entities, and on the other seeming to be concrete machines operating in time, since their operations are 'local' and since they were intended to represent what could be done by a human being following an effective procedure.

[☆] I am grateful to Berry Groisman, Hannes Leitgeb, Oystein Linnebo, James Lambert, Stuart Presnell, Tony Short, Mark Sprevak and Chris Timpson for discussion of the issues addressed in this paper.

E-mail address: james.ladyman@bristol.ac.uk.
URL: http://www.bristol.ac.uk/philosophy/department/staff/jl.html.

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.09.047

Discussions of important issues in the foundations of computer science are liable to degenerate into confusion if the distinction between the formal and the physical is not observed. When we are concerned with the logical form of a computation and its formal properties then it can be theoretically described in terms of mathematical and logical functions and relations between abstract entities. However, actual computation is realised by some physical process, and the latter is of course subject to physical laws and the laws of thermodynamics in particular. A controversial issue is that of whether or not there are any systematic connections between the logical properties of computations considered abstractly and the thermodynamical properties of their concrete physical realizations. Landauer [14] proposed such a general connection, known as Landauer's Principle, usually formulated as the claim that the erasure of information in any computational device is necessarily accompanied by an increase in the thermodynamic entropy of the device and/or its environment. Landauer's Principle is the subject of much debate. In particular, Norton [18] and Maroney [17] both argue that it has not been shown to hold in general. To resolve this matter an analysis of the notion of the implementation of a computation by a physical system is clearly required.

Another issue that calls for an analysis of implementation is that of realism about computation, where this is the view that whether or not a particular physical system is performing or implementing a particular computation is at least sometimes a fact that obtains independently of human beliefs, desires and intentions.[1] Unless we have a precise account of implementation it will not be possible to decide whether or not realism about computation is correct just because it will not be clear what 'computation' means. Indeed, it is only when we are talking about concrete computers and computations that realism in the sense defined above is an issue, and it is the notion of implementation that matters for assessing in virtue of what physical systems and processes can be said to be computers and computations. Various arguments have been put forward purporting to show that any physical system of sufficient complexity trivially implements all computations (see Putnam [22] and Searle [24]). These arguments have prompted defenders of realism about computation to develop accounts of implementation to which they do not apply. Ladyman et al. [11] present an analysis of what it is for a physical process to implement a logical transformation that was motivated solely by an attempt to clarify the content of Landauer's Principle and provide a general proof of it. They did not consider the question of realism about computation. In order to clarify the status of Landauer's Principle it is necessary precisely to define a computation, and what it means to say that a computation is physically implemented. It is also necessary to precisely define logical irreversibility and thermodynamic irreversibility. It is also important to make a clear distinction between the abstract and physical domains, and to avoid talk of logical 'processes' and refer instead to logical transformations and their implementation by *families* of physical processes. The result of this analysis is the notion of an *L-machine*. This is a hybrid physical-logical entity that combines a physical device, a specification of which physical states of that device correspond to various logical states, and an evolution of that device which corresponds to the logical transformation *L*. The most general form of Landauer's Principle can be precisely stated in terms of *L*-machines, namely that the logical irreversibility of *L* implies the thermodynamic irreversibility of every corresponding *L*-machine.

In this paper this account of implementation is explained and motivated, and its implications assessed. First, in the next section, realism about computation and the difficulties it faces are reviewed. In Section 3, the abstract notion of a computation is formally captured by the idea of a logical transformation. In Section 4, the relevance of how we think about computation for the status of Landauer's Principle is explained as well as the elementary thermodynamics necessary to precisely formulate Landauer's Principle. Section 5 features the promised account of implementation, and Section 6 compares it to other recent accounts of computation in the literature. Finally, Section 7 assesses its implications for realism about computation and the use of Landauer's Principle in foundational debates, and suggests where further work is needed. The results of Sections 3–5 were originally published in Ladyman, et al. [12] and discussed in Ladyman [11], and Sections 2, 6 and 7 present new material.

2. Realism about computation

The term 'realism' is used in many ways. As stated above, realism about computation is the view that whether or not a particular physical system is performing a particular computation is at least sometimes an objective matter that depends not at all on our beliefs, desires and intentions. The caveat 'at least sometimes' is necessary here because a realist about computation need not believe that all instances of computation should be realistically construed. The computational theory of mind (first developed by Putnam [20] and Fodor [9], and for a recent introduction see Horst [10]), according to which mental processes such as language comprehension are computations, presupposes realism about computation. If whether or not the human nervous system implements particular computations is not a natural fact about the world that is independent of whether we represent it as doing so, then the computational theory of mind fails to naturalise the mind. Similarly, claims that the universe is a computer, that computation occurs in bacteria and cells, and so on must be understood merely as claims about how we think about these systems if realism about computation is false. Realism about computation is also

[1] An anonymous referee objects that realism usually involves a commitment to the existence of entities rather than merely to the objective truth values of propositions. While this is the case in the context of mathematical or scientific realism there is also a well established usage of the kind I employ in the context of morality and aesthetics in which domains realists are not necessarily inclined to be saddled with an ontology of moral or aesthetic objects or properties.

presupposed by attempts to use computational principles such as Landauer's Principle to dispel Maxwell's Demon. Realism about computation has been challenged by Hilary Putnam and John Searle among others. Searle [24] argues against realism about computation by reductio: if it were sufficient for some physical system to implement a particular computation that within it there be a pattern of activity with a particular structure, then since most macroscopic physical systems are hugely complex, there will be some pattern of activity within almost every macroscopic system of almost any particular structure, and so almost every macroscopic physical system would implement any given computation. Hence, it cannot be purely in virtue of the patterns of physical activity that systems implement computations, and since these are the only features of physical systems about which we should be realists, realism about computation fails. Putnam [22, pp. 120–125] argues that every ordinary open system realises every finite automaton because it is always possible to take a sequence of the successive temporal states of such a system as representing the successive states of the automaton. Clearly, the basic idea of both arguments is that provided a physical system has sufficient complexity it can be said to implement any computation and therefore there is no fact of the matter about systems implementing one computation in particular.

A common response to these arguments (for example, [2] is to claim that for a computation to be implemented depends not only on what a system does but also on what it would do were its initial state different in an appropriate way. For example, a handheld calculator can only be said to calculate the sum of 7 and 5 when the appropriate buttons are pressed and the screen outputs 12, because had different buttons been pressed it would have given a different output. Similarly, a logic gate that gives the correct output '1' when the inputs represent '1' and '1', is only computing AND if, had the inputs represented '1' and '0' it would have given the output '0'. In general, functions are maps from many different values in the domain of the function to values in its range, but in general implementations are only of a particular instance of the function, in others words, they are processes that evolve from a state that represents one value in the domain to the state that represents the corresponding value in the range. The upshot would appear to be that realism about modality is a necessary component of realism about computation, where realism about modality is the claim that there are mind-independent facts about counterfactual as well as factual matters.

However, such facts of the matter about what a system would do given different initial conditions are not sufficient for realism about computation because we must also suppose that there is a fact of the matter about what the physical states of the system represent. In the above examples, it is only because the interpretation of the states of the physical systems in question is held fixed that they can be said to implement a particular computation. Hence, realism about computation is closely related to realism about representation. The Greek letter π represents the ratio of the circumference to the diameter of a circle, but this is a purely conventional fact that does not obtain independently of human beliefs, desires and intentions. The question then arises as to whether there are any forms of representation that are not similarly dependent on us. The representation of concepts and objects by words of natural language is usually arbitrary but other kinds of representations, for examples maps and pictures, have structural similarities with that which they represent. Putnam [21] considers the case of an ant that crawls across sand leaving a trace that happens to look like a passable caricature of Winston Churchill. He argues that it is nonetheless not a representation of the man because the ant did not intend it as such, and in general rejects the possibility of naturalizing representation. On the other hand, many philosophers and scientists have advocated theories according to which at least some representations can be considered natural. One prominent approach that is often applied to animal signaling systems and human natural language is so-called 'teleosemantics', according to which a property or state of an organism represents some feature of the world just in case it evolved in virtue of the adaptive value of carrying the relevant information about the world. Hence, representative content is tied to biological function.[2] It seems then that realism about the computations that occur in the nervous systems of organisms, and about natural computation in general depends on realism both about modality and realism about what particular states of the systems represent. Teleosemantics offers one way that the latter might be defended but this account does not extend to systems that have not evolved according to natural selection and hence whose states cannot be assigned objective representational roles on the basis of their functions.

3. Logic

In this section a general framework for describing the purely logical properties of computations considered as abstract entities is described. Among computer scientists it is common to think of computation in terms of Turing machines. There are a number of reasons why this approach is not followed here. Firstly, as suggested in the Introduction above, Turing machines seem to sit between the formal and physical realms and hence are not suited for the project of precisely clarifying the relationship between these realms. Secondly, Turing introduced his machines to capture the idea of what could be computed by an effective procedure that a human being could in principle follow. While it is true that the class of Turing-computable functions is equivalent to the class of functions computable by the Lambda calculus, by Post machines, Markov algorithms, and Kleene's Formal systems, there are notions of computation with respect to which Turing non-computable functions are computable. Whether or not so-called hypercomputation is physically realizable (perhaps by quantum computers) it seems appropriate to adopt a definition of computation that is completely general and hence broader than that captured by the operation of Turing machines. Finally, the approach taken here is adequate for the task at hand and seems to be

[2] A recent collection of the latest thinking about teleosemantics is MacDonald and Papineau [16]. Sprevak [25] defends realism about computation using a teleosemantic account of representation.

conceptually minimal and introduces no unnecessary features, and these facts seem sufficient to justify its use. While the formalization presented here was designed to address Landauer's Principle it is not particular to that context and is intended to be completely general.

In general, a computation thought of as an abstract entity is simply a single-valued map L from a finite set X of input states, into a finite set Y of output states (i.e. each input state is mapped by L to a unique output state). Often the maps of interest are logical transformations. For example, consider the case of binary-valued logic, in which the input and output states are bit-strings (with 0 and 1 usually representing 'false' and 'true', respectively); the mapping L can be represented by a truth table, or as a digital circuit constructed from some set of universal gates (e.g. NAND and COPY). In connection with Landauer's Principle, it is important that such logical operations can be reversible, in which case knowledge of the operation and the output is sufficient for knowledge of the input, or irreversible, in which case there is more than one input state that it is mapped to a single output state. Hence, we say that a logical transformation is *logically reversible* if and only if $L : X \to Y$ is a one-to-one (injective) mapping.[3] If L is not a one-to-one mapping, then it is *logically irreversible*.

Given this account of computations qua abstract entities an important fact can immediately be seen. A logical transformation is a map from a *set* of logical states to a *set* of logical states, whereas a physical process is a change in a physical system whereby it goes from a *particular* physical state to a *particular* physical state. Hence, strictly speaking a physical process cannot be said to implement a logical transformation because all it could ever do is implement the part of the map that takes one of the logical input states to another logical input state. In terms of the truth table that represents the logical transformation AND for example, clearly a particular physical process could only be said to implement a single row of it. For a physical system to implement a logical transformation there must be a family of processes and each of the physical states that represent the logical input states must be taken by one member of the family to the appropriate physical state, that is the one that represents the right logical output state. It should be noted then that here again the modal notion of implementation comes to the fore. A process in some physical system can only be said to implement a computation if the former is one of a family such that had the input been different another member of the family would have evolved it into the appropriate output. The notion of implementation of a logical transformation by a physical device presented in Section 5 below incorporates this essential feature of implementation and further explains it.

4. Thermodynamics

Discussions of Landauer's Principle have been obscured by a lack of clarity about when computations are being considered qua abstract mathematical objects and when they are being considered qua concrete physical processes. For example, Landauer's Principle is often generalised as follows: (a) any logically irreversible process must result in an entropy increase in the non-information bearing degrees of freedom of the information-processing system or its environment; (b) any logically reversible process can be implemented thermodynamically reversibly (see for example Bennett [1]). Claim (a) is obscure because the notions of information-bearing versus non-information bearing degrees of freedom are unclear in so far as the notion of information is not clearly defined in this context. The idea of information-bearing degrees of freedom perhaps has to do with which parts of a physical system are being used as representations and which are not. (a) can only be assessed when such matters are made explicit. On the other hand, (b) engenders confusion because to call a process logically reversible or irreversible is a category mistake. Computations qua abstract mathematical objects can be logically reversible or irreversible but not physical processes. Correspondingly, only the physical processes that implement computations may be thermodynamically reversible or irreversible. Hence, in what follows the term 'process' always refers to a physical process in which a system starts in some particular state and ends in some particular state.

Everyone in the literature about Landauer's Principle (see for example the papers in [15] as well as [18]) agrees that there are both logically reversible and irreversible transformations, and that every logically reversible transformation is in principle implementable in a thermodynamically reversible way, and that any such transformation can also be implemented in a thermodynamically irreversible way since it could be done non-optimally. It is also uncontroversial that a logically irreversible transformation can be implemented in a thermodynamically irreversible way. So the controversy concerning Landauer's Principle is about whether there are any logically irreversible transformations that can be implemented in a thermodynamically reversible way. Thermodynamic irreversibility is a feature of processes as expressed by the second law of thermodynamics. There is much controversy about how the latter can be justified on the basis of statistical mechanics. Nonetheless, in physics it is usually assumed that phenomenological thermodynamics and in particular the second law stated in terms of the *thermodynamic entropy* is valid.

The basic idea of thermodynamics relevant to computation are here briefly explained. Consider a system in a heat reservoir at temperature T undergoing some thermodynamic process p. Let $\Delta S_{\mathrm{sys}}(p)$ be the change in the entropy of the system during the process p, and let $\Delta Q(p)$ be the heat flow from the system into the reservoir during this process. The second law requires that

$$\text{for all } p,\ \Delta S_{\mathrm{sys}}(p) + \frac{\Delta Q(p)}{T} \geq 0. \tag{1}$$

[3] Note that whether or not L is surjective is irrelevant for present purposes because if there are output states that do not get mapped to from any input states these are irrelevant to thermodynamic considerations about the implementation of the computation.

Identifying $\Delta S_{\text{res}}(p) = \Delta Q(p)/T$ as the entropy change of the heat reservoir, define

$$\Delta S_{\text{tot}}(p) = \Delta S_{\text{sys}}(p) + \Delta S_{\text{res}}(p) \tag{2}$$

as the total entropy change of the system and reservoir together. The second law can then be restated in the familiar form

$$\text{for all } p,\ \Delta S_{\text{tot}}(p) \geq 0 \tag{3}$$

i.e. total entropy is non-decreasing over time.

A process p is *thermodynamically reversible* if and only if $\Delta S_{\text{tot}}(p) = 0$.

If $\Delta S_{\text{tot}}(p) > 0$, the physical process p cannot be run in reverse, as the reverse process p' would have $\Delta S_{\text{tot}}(p') < 0$, and hence violate the second law. Therefore, any process p for which $\Delta S_{\text{tot}}(p) > 0$ is *thermodynamically irreversible*. As is well known, there are a number of formulations of the second law that are provably equivalent to this, modulo certain assumptions.[4]

Since the implementation of a logical transformation by a physical process must be defined with respect to a family of physical processes rather than with respect to a particular physical process, it is the thermodynamic status of the whole family that is of interest. A family of physical processes will be said to be thermodynamically irreversible if and only if at least one of its members is. This is important for the definition of irreversibility for L-machines in the next section.[5]

5. Implementation: The notion of an *L*-machine

Having clarified computation qua logical entities and the idea of logical reversibility, and having outlined the thermodynamics of physical processes, it is now possible to consider the relationship between the logical and physical realms. We are concerned with a simple question namely what does it mean for a physical system to implement a logical transformation. For a logical transformation to be physically implemented there must be: A physical device, a specification of which physical states of that device correspond to the possible logical states (call the former *representative states*), and a time evolution operator of that device. This combined system is an *L-machine*. Here L names a particular logical transformation, so there are L_{AND}-machines, and so on.

The time evolution operator must generate the relevant family of processes, and the reliability of the implementation consists in the time evolution operator being such as to ensure that *whichever* of the representative physical states the device is prepared in, it ends up in the appropriate representative state.[6] Maroney [17] considers only individual processes and this leads him to conclude that Landauer's Principle can be violated. Recall that it was pointed out above that an individual process can only implement part of a logical transformation. The example given was that of AND and how an individual process can only implement one row of the corresponding truth table. The significance of this becomes clear when we consider the 'Reset' operation represented by the truth table

Input	Output
0	0
1	0

'Reset' is the simplest logically irreversible transformation. Let the input states be x_1 ='0' and x_2 ='1', and the output states be y_1 ='0' and y_2 ='1', the map $L_{\text{Reset}} : X \to Y$ corresponding to the Reset operation is given by

$$L_{\text{Reset}}(x_1) = L_{\text{Reset}}(x_2) = y_1. \tag{5}$$

Every logically irreversible transformation is equivalent to a logically reversible transformation plus one or more Reset operations. To see this consider an arbitrary logically irreversible transformation. It can be converted into a reversible transformation if a copy of the input state is appended to its output. This clearly allows the input state to be recovered

[4] In thermodynamics various operational assumptions are made that allow the definition of the thermodynamic entropy of individual macroscopic states up to a constant (see, for example, Fermi [8], Chapter IV). This is almost universally accepted, however, there is a good deal of controversy about the assignment of entropy to probabilistic mixtures of macrostates (for example, see Norton [18]). For example, consider the mixture of macrostates M_i, with probabilities q_i. Assuming that the assignment of entropy to such a state is legitimate, it might be supposed that it is simply the average of the individual entropies $S(M_i)$; explicitly, $\sum_i q_i S(M_i)$. However, it is common to also include a term to represent the contribution to the entropy of the probability distribution itself; explicitly:

$$S_{\text{mixture}} = \sum_i q_i S(M_i) - k \sum_i q_i \ln q_i. \tag{4}$$

The latter term is an information theoretic entropy and its inclusion in thermodynamic calculations currently lacks rigorous foundational justification. However, such a justification is given by Ladyman, Presnell and Short [13]. There are two proofs of Landauer's Principle in Ladyman et al. [12] one of which depends on the use of the information theoretic entropy and one of which that is independent of it.

[5] This is directly analogous to how logical reversibility is thought about. For example, AND is said to be logically irreversible because at least one of the outputs is mapped to by more than one input.

[6] Note that the requirement of complete reliability is an idealization as all that is required in practice is that the appropriate physical state be arrived at with very high probability.

from the output state. To obtain a transformation logically equivalent to the original irreversible transformation we simply reset the copy.

Reference is often made to ‘erasure’ in discussions of Landauer’s Principle. However, ‘erasure’ can be taken to refer either to ‘Reset’ or to any other transformation where the output is independent of the input (including probabilistic transformations sometimes referred to as ‘randomising data’). So ‘Erasure’ is an ambiguous term and indeed Leff and Rex [15, pp. 22–25], refer to ‘erasure/resetting’. Maroney [17] also considers the probabilistic transformation ‘RAND’ which outputs 0 or 1 each with 50% probability regardless of the input. This is not a logical operation according to the definition above and ‘RAND’ is not in the scope of the present discussion. As ‘Reset’ is a logical operation, ‘Reset’ and not ‘RAND’ is normally used in practical computation. Clearly if we only cared about the thermodynamics of a single process in considering the implementation of RESET then we could implement it in a thermodynamically reversible way for the top row of the above truth table could be implemented by process that simply left the physical state of the system unchanged. That this is not adequate as an implementation of RESET is apparent when we insist that for the operation to have been implemented it must be the case that had the input state been 1 the corresponding output state would have been arrived at.

Formally, an L-machine is an ordered set

$$\{D, \{D_{\mathrm{in}}(x)|x \in X\}, \{D_{\mathrm{out}}(y)|y \in Y\}, \Lambda_L\} \tag{6}$$

consisting of

- A physical *device* D, situated in a heat bath at temperature T.
- A set $\{D_{\mathrm{in}}(x)|x \in X\}$ of macroscopic input states of the device, which are distinct thermodynamic states of the system (i.e. no microstate is a component of more than one thermodynamic state). $D_{\mathrm{in}}(x)$ is the representative physical state of the logical input state x.
- A set $\{D_{\mathrm{out}}(y)|y \in Y\}$ of distinct thermodynamic output states of the device. $D_{\mathrm{out}}(y)$ is the representative physical state of the logical output state y. Note that the set of input states and output states may overlap.
- A time-evolution operator Λ_L for the device, such that

$$\forall x \in X, \Lambda_L(D_{\mathrm{in}}(x)) = D_{\mathrm{out}}(L(x)). \tag{7}$$

An L-machine $\{D, \{D_{\mathrm{in}}(x)|x \in X\}, \{D_{\mathrm{out}}(y)|y \in Y\}, \Lambda_L\}$ physically implements L in the following sense. If D is prepared in the input state $D_{\mathrm{in}}(x)$ corresponding to the logical input state $x \in X$, and is then evolved using Λ_L, it will be left in the output state $D_{\mathrm{out}}(y)$ corresponding to the logical output state $y = L(x) \in Y$. This physical process is denoted by p_x.

$$\begin{tikzcd} x \arrow[r, "L"] \arrow[d, equal] & y \arrow[d, equal] \\ D_{\mathrm{in}}(x) \arrow[r, "\Lambda_L"'] & D_{\mathrm{out}}(y) \end{tikzcd} \tag{8}$$

Fig. 1. The relationship between the logical states x and y and their representative physical states $D_{\mathrm{in}}(x)$ and $D_{\mathrm{out}}(y)$, and the logical transformation L and the physical time evolution operator Λ_L.

Now consider the thermodynamics of the process p_x. If the entropy of the system in the state $D_{\mathrm{in}}(x)$ is $S_{\mathrm{in}}(x)$, the entropy of the system in state $D_{\mathrm{out}}(L(x))$ is $S_{\mathrm{out}}(L(x))$, and the heat flow from the system into the reservoir during the process is $\Delta Q(p_x) = T\Delta S_{\mathrm{res}}(p_x)$, the total entropy change $\Delta S_{\mathrm{tot}}(p_x)$ for the process will be given by

$$\Delta S_{\mathrm{tot}}(p_x) = S_{\mathrm{out}}(L(x)) - S_{\mathrm{in}}(x) + \frac{\Delta Q(p_x)}{T} \geq 0. \tag{9}$$

This particular process will be thermodynamically reversible if $\Delta S_{\mathrm{tot}}(p_x) = 0$.

An L-machine is *thermodynamically reversible* if and only if $\forall x \in X, \Delta S_{\mathrm{tot}}(p_x) = 0$ (i.e. if all of the processes p_x are thermodynamically reversible). An L-machine is therefore *thermodynamically irreversible* if there exists an $x \in X$ for which $\Delta S_{\mathrm{tot}}(p_x) > 0$.

Using the above definitions Ladyman et al. [11] prove Landauer’s Principle from the Kelvin statement of the Second Law of Thermodynamics using a thermodynamic cycle.[7]

[7] Implementing L by implementing some other ‘stronger’ L' from which the outputs of L can be deduced, for example, the logical transformation L' corresponding to the combination of L and keeping a copy of the inputs, is ruled out by this definition. Ladyman et al. [12] justify this restriction. Note also that in the above definition a unique representative state is assigned to each logical state as this makes for a clear and simple analysis. However, in general it could be allowed that more than one physical state represents the same logical state, and Ladyman et al. [12] consider such machines too.

6. Related work

In this section some other recent accounts of the nature of computation are reviewed and compared with the present account. The trivialization arguments against realism about computation referred to in Section 2 work only on the assumption that it is admissible to freely associate particular states of physical systems with particular computational states. In practice of course it is only possible to use a system as a computer if: (a) the relevant physical states are distinguishable by us (with our measurement devices); and (b) it is possible for us to put the system into a chosen initial state so as to compute the function in question for it. However, in principle some reason must be given as to why a system with respect to which these constraints are not met cannot be considered to be computing nonetheless. One obvious way of blocking the trivialization arguments then is to stipulate that for physical states to count as computational states they must be genuinely representational. Hence, most accounts of computation (see, for example, Churchland [3], Cummins [4], Dennett [6]) agree with Fodor [9, p. 180] and the present account that computation requires representation. The further question that then arises is whether or not the representation relation itself admits of a realist interpretation. Pragmatically, we are able to pick physical states and assign representational content to them but are there any systems whose states have their representational content independently of such stipulations? Clearly the states of the devices we use as computers do not, but what of the physical states of our brains or of molecules involved in genetic 'computations'. Many would argue that in such cases the representation relation can be naturalized since the states in question are part of systems which evolved to have the function of representing the state of the external world, and the structure of proteins, respectively. Hence, realism about computation for naturally evolved systems is viable only if realism about their representational content can be sustained perhaps by something like the teleosemantic account. The present account has nothing to contribute to this project and simply takes the representation relation for granted in the analysis of implementation. Where it differs from the influential theories of computation like that of Fodor is in dispensing with notions of syntax and symbol processing, relying instead on the simple idea that an implementation of a function is achieved when a representation relation can be established between logical and physical states, and when with respect to that relation the time evolution of the system maps physical states to physical states in accordance with the function. The notion of syntax in any case seems to rely upon the notion of representation in real cases. Physical states of certain kinds are lumped together, small differences are ignored, and they are regarded as tokens of a particular type, but this is just a low level case of representation, as when various marks on paper are all regarded as instances of a certain letter of the alphabet.

Not all accounts of computation rely on the received idea that representation is necessary for computation. A recent dissenting view is that of Piccinini [19]. His account individuates computational states according to their functional role. Another dissenting view is that of Egan [7] who argues that real computation can be understood purely in terms of the mathematical theory of computation. I will not offer a full critique of their views here but I do think that neither Piccinni's nor Egan's theory has the resources to solve the problem of distinguishing between the computation of AND and the computation of OR presented in the next section.

Another view of computation is that of Chalmers [2], Copeland [5] and Scheutz [23] who all argue that computations are to be individuated in virtue of their causal powers. One problem with this view is that it entails that every causal process is a computational process. A further problem is that the notion of causation is itself highly problematic and controversial. In any case, I shall not discuss the causal approach further here but for criticisms see Sprevak [25].

7. Reflections

It is important that the labelling of physical states used to represent logical states is essential to the identity of a L-machine. For example, exactly the same device and time-evolution operator could be used as part of both an L_{AND}-machine, and an L_{OR}-machine by the appropriate relabelling of the physical input and output states. Hence, when it comes to physical implementation it is clear that certain logical transformations are equivalent just because an appropriate translation scheme allows any physical process that implements one of them to be used to implement the other. OR and AND are like this. If we permute which physical input states are taken to represent '1' and '0', and do the same with the physical output states then the implementation of OR becomes the implementation of AND and vice versa. AND and OR are structurally the same. This provides an argument against those who argue that representation is not necessary for a particular process to implement a particular computation for they lack the resources to distinguish between the implementation of AND and OR. While the implementation of one of these logical operations may be part of a bigger computation and have a functional role with respect to it, this need not be the case. AND and OR may be implemented as single computations and an account of implementation should have the resources to describe this.

Similarly, IFF and EOR are structurally the same. Assessing whether a physical process can be used to implement such a transformation requires paying attention to the structure of the transformation in the above sense. In the discussion of realism about computation in Section 2 it was pointed out that the most plausible way of defending realism about representation is not applicable to systems that have not evolved by natural selection. However, while in such cases it may not be an objective fact that a particular physical process implements a particular logical transformation, perhaps it is an objective fact that it implements a particular transformation structure since its ability to do so depends on the structure of the system and the laws governing its time evolution and is independent of exactly what the physical states of the system are taken to represent. Certainly, what is relevant to the physics of computation is the structure of the transformation and not its particular nature. (The structure of logical transformations is elucidated by the combinatorial theory of Stirling Numbers of the Second Kind that quantify how many structurally distinct ways there are for n possible inputs to be mapped to m possible outputs. For example, in the case of the familiar truth-functional connectives of propositional bivalent logic $n = 4$ and $m = 2$, and the number of structurally distinct such connectives is 7.)

When it comes to representation it is not the case that we can freely choose any physical states to represent computational states, for as pointed out in the previous section, we must be able to both distinguish the states and manipulate them appropriately. So while our intentions to use particular states as representations may be necessary for computation they are not sufficient. Finally, note that one important lesson of both the debate about realism about computation and the debate about the thermodynamics of computation is that, while modal facts about the physical system used to implement a logical transformation are not sufficient to determine that it implements any particular transformation, they are necessary for the facts about which transformation it can and does implement.

References

[1] C.H. Bennett, Notes on Landauer's principle, reversible computation, and Maxwell's demon, Studies in the History and Philosophy of Modern Physics 34 (2003) 501–510.

[2] D. Chalmers, Does a rock implement every finite state automaton? Synthese 108 (1996) 309–333.

[3] P. Churchland, Neurophilosophy, MIT, Cambridge, MA, 1986.

[4] R. Cummins, Meaning and Mental Representation, MIT, Cambridge MA, 1989.

[5] J. Copeland, What is computation? Synthese 108 (1996) 335–359.

[6] D. Dennett, Intentional systems, Journal of Philosophy 68 (1971) 87–106.

[7] F. Egan, Computation and content, Philosophical Review 104 (1995) 181–204.

[8] E. Fermi, Thermodynamics, Dover, New York, 1936.

[9] J. Fodor, The Language of Thought, Thomas Crowell, New York, 1975.

[10] S. Horst, The computational theory of mind, in: E. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, Fall 2005 edition, 2005.

[11] J. Ladyman, Physics and computation: The status of Landauer's principle, in: S.B. Cooper, B. Lwe, A. Sorbi (Eds.), Computation and Logic in the Real World, Third Conference on Computability in Europe, CiE 2007, Siena, Italy, June 2007, Proceedings, in: Lecture Notes in Computer Science, vol. 4497, Springer-Verlag, 2007, pp. 446–454.

[12] J. Ladyman, S. Presnell, A. Short, B. Groisman, The connection between logical and thermodynamic irreversibility, Studies in History and Philosophy of Modern Physics 38 (2007) 58–79.

[13] J. Ladyman, S. Presnell, A. Short, The use of the information theoretic entropy in thermodynamics, Studies in History and Philosophy of Modern Physics 39 (2008) 315–324.

[14] R. Landauer, Irreversibility and heat generation in the computing process, IBM Journal of Research and Development 5 (1961) 183–191. Reprinted in Leff and Rex (1990).

[15] H.S. Leff, A.F. Rex (Eds.), Maxwell's Demon: Entropy, Information, Computing, Adam Hilger, Bristol, 1990.

[16] G. MacDonald, D. Papineau (Eds.), Teleosemantics, Clarendon Press, Oxford, 2006.

[17] O.J.E. Maroney, The (absence of a) relationship between thermodynamic and logical reversibility, Studies in History and Philosophy of Modern Physics 36 (2005) 355–374.

[18] J.D. Norton, Eaters of the lotus: Landauer's principle and the return of Maxwell's demon, Studies in the History and Philosophy of Modern Physics 36 (2005) 375–411.

[19] G. Piccinini, Computation without representation, Philosophical Studies (2006).

[20] H. Putnam, Brains and Behavior Read as part of the programme of the American Association for the Advancement of Science, Section L (History and Philosophy of Science), December 27, 1961.

[21] H. Putnam, Reason, Truth and History, Cambridge University Press, Cambridge, 1981.

[22] H. Putnam, Representation and Reality, MIT, Cambridge, MA, 1988.

[23] M. Scheutz, When Physical Systems Realize Functions, Minds and Machines 9 (1999) 161–196.

[24] J. Searle, The Rediscovery of the Mind, MIT, Cambridge, MA, 1992.

[25] M. Sprevak, Computation in Mind and World, Ph.D. Thesis, University of Cambridge, 2005.