On the philosophical basis of computational theories

Oded Goldreich
Department of Computer Science
Weizmann Institute of Science
Rehovot, Israel.
oded.goldreich@weizmann.ac.il

February 14, 2014

Abstract

In this short note we exemplify the problems that may arise when presenting computational theories based on theories of natural phenomena that lack computational complexity components. These problems may arise when the latter theories refers to environments of unbounded size. In contrast, the standard theory of computation (e.g., based on the model of Turing machines) refer implicitly to theories about bounded portions of the environment (e.g., the mechanics or electricity of possible implementations of a Turing machine).

We start by recalling the postulates that underly all reasonable computational models, as presented in standard computational classes and texts. We then focus on exemplifying what happens when these postulates are ignored.

The metaphysical foundations of all reasonable computational models

A reasonable model of computation consists of large *environments* that are *transformed according to simple and/or local rules* (having small description). The vague dichotomy between "large" and "small" is typically modeled by the formal dichotomy between infinite and finite. The *environments* are bit strings *of arbitrary, a priori unbounded, length*, where bits represent binary attributes.[1] In contrast, the *rules-of-transformation* are local and describe how an *a priori bounded part* of the environment changes, or more precisely how the value of each bit in the environment is updated as a function of a fixed (bounded) number of bits. The computation consists of the sequential (or parallel) applications of such rules.

Indeed, the models of Turing machine and Boolean circuits fit the foregoing paradigm.[2] In contrast, applying arbitrary linear transformation to the entire environment does not fit

[1] We comment, in passing, that the notion of a binary attribute is one of the most basic philosophical notions (cf., from the most ancient philosophies to Structuralism).

[2] We refer to Boolean circuits of bounded fan-in. Boolean circuits of unbounded fan-in are defined in order to facilitate the introduction of a complexity measure that reflects alternation of gate types in the former circuits.

the foregoing paradigm, and it is philosophically unclear why general linear transformation of an unbounded dimension should be allowed while other transformation are not allowed: The mere fact that some physical theory is postulated in terms of a certain type of transformations is far from providing a good justification (for considering any such transformation as an admissible rule of transformation in a computational model).

The consequences of violating these postulates

The purpose of this note is to demonstrate why justifications of the foregoing type do not suffice; that is, a computational model cannot be justified by merely asserting that it is consistent with some theory of natural phenomena.[3] Such a consistency is at most a necessary condition (i.e., it is necessary assuming the theory of nature is correct), but it is not a sufficient condition. This is demonstrates by intentionally presenting a computational theory that we expect to be deemed unreasonable by any reader, although this computational theory can also be "justified" by an analogous reference to a theory of natural phenomena (which lacks computational considerations). Specifically, this computational theory is based on (a careless abstraction of) the electrical realities of electronic components that are used for implementing Boolean gates. We consider an electrical component that implements a Boolean OR, and we call the underlying model an *electrical OR*.

Definition 1 (electrical OR): *An* electrical OR *has two input wires and one output wire, where each wire carries either a* (noisy) *low value or a* (noisy) *high value. The base* (noise-free) *low and high values are denoted ℓ and h, respectively. The* (noisy) *value of a wire has the form $\ell + (h-\ell)\cdot b + \epsilon \cdot \sum_{i\geq 1} 2^{-i}\cdot v_i$, where $\epsilon \in \{\pm(h-\ell)/3\}$ and $b, v_1, v_2, ... \in \{0,1\}$. Any value of the above form is called* legal*; that is, a* legal value *v satisfies*

$$v = \ell + (h-\ell)\cdot b + \epsilon \cdot \sum_{i\geq 1} 2^{-i}\cdot v_i, \tag{1}$$

where $b, v_i \in \{0,1\}$ for every $i \geq 1$. This legal value is said to represent the Boolean value *b. The functionality of the OR-gate, denoted $\Gamma : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, maps every pair of legal values for the input wires to a legal value for the output wire such that the output value represents the Boolean OR of the representations of the input values.*

Does the above electric OR (rather than the underlying Boolean OR)[4] represent a reasonable model of computation? After all, it seems that such a representation of Boolean values is in the basis of all electronic computers and is consistent with the relevant theories of electricity.

We believe that the answer is NO, and the issue is that we did not fully specify the gate's functionality (but rather allowed arbitrary electrical gates as long as they satisfy

[3] In addition to the fundamental point made next, we wish to express our opposition to the term "Law of Nature": We know of no "laws of nature" – we only have our theories regarding what these laws are or may be.

[4] We are, of course, not concerned with the standard use of electrical gates as defining Boolean gates. In such a standard use, the specifics of the gate's (real) functionality (as captured by Γ) are ignored. We are concerned of a hypothetical theory that may postulate specific functionalities Γ and try to capaitalize on such postulates.

the requirement of Definition 1). This does not matter as long as one only relates to the Boolean functionality that underlies Definition 1, but things change if one postulates that any gate that satisies Definition 1 can be realized (and constitutes a reasonable model of computation). In order to clarify this deficiency, we need another definition.

Definition 2 (residual representation): *The* residual sequence represented by a value v *as in Eq. (1) is the infinite sequence* $(v_1, v_2, ...)$.

While we specified how the gate transforms the *Boolean values* represented on its wires, we did not specify how it transforms the *residual sequences* represented on these wires. A few hazardous options arise. Basically, the gate functionality may transform the values v^1 and v^2, which represent residual sequences $(v_1^1, v_2^1, ...)$ and $(v_1^2, v_2^2, ...)$ into a value v^o representing a residual sequence whose i^{th} bit (i.e., v_i^0) is the result of applying a hard Boolean function, denoted F_i, to the i-bit long prefixes of the input residuals; that is, $v_i^0 = F_i(v_1^1, v_2^1, ..., v_i^1, v_1^2, v_2^2, ..., v_i^2)$. This hard Boolean function may be the halting problem's predicate, its time-bounded version, a hard monotone function such as the max-clique predicate, etc.

We stress that the foregoing model does not seem to contradict any relevant law of electricity. Furthermore, this model can be revised in various ways to make the transformation of the residual sequences more "smooth" (in various sense).[5] It also seems consistent with the relevant laws of electricity that we can (1) assign wires electrical values such that the prefixes of the residual sequences represented by them equal any desired bit string, and (2) measure the value of each wire to any desired accuracy. The reader can easily see the hazardous consequences of these assumptions.

What went wrong was the neglect of computational complexity considerations in the definition of the computational model. The source of trouble is the implicit postulate that asserts that *whatever is not forbidden explicitly by the relevant electrical theories, can actually be implemented at no cost*. More refined theories may assert that actions such as (1) and (2) above do have a cost, and that this cost is related in some way to the length of the bit string being "written" into the value or "read" from it. They may also assign a cost to different gate functionalities, and we do expect that the cost will correspond to the computational complexity of the transformations that are performed by these gates. Needless to say, such refined theories will incorporate computational complexity considerations, which will prevent hazardous consequences.

[5] In particular, note that so far we have modeled a single gate, and all hazards will follow from abusing a single gate. In this single-gate model, each bit of the residual sequence represented in the output was a complex function of prefixes of the residual sequences represented in the inputs. In the rest of this footnote, we briefly describe an alternative model that refers to collections of such gates (i.e., circuits) rather than to a single gate. In this model we only use complex functions when determining the most significant bit of the residual sequence represented in the output, but we allow gates to "sense" the size of the circuit in which they reside and select a functionality accordingly. Specifically, when being part of a circuit of size n, each gate uses a function F_n to determine the most significant bit of the residual sequence represented in its output (where this function is applied to the n-bit long prefixes of the residual sequences represented in its input wires).

The **lesson** is that when natural phenomena are used as a basis for the definition of computing devices, the theories about these phenomena must incorporate elements of computational complexity, which theories of natural phenomena typically lack.

A special case of incorporating elements of computational complexity is when refering only to natural phenomena of a priori bounded size; in such a case, the "complexity" of the natural phenomena is a constant and it can be taken as a unit cost (as is done in the reasonable models of computation that were briefly reviewd at the beginning of this note). The problem is with theories of natural phenomena that are of unbounded size and with ignoring the question of the "complexity" of these phenomena.

A remotely related work. This note is remotely related to Adi Shamir's paper "Factoring Numbers in $O(\log n)$ Arithmetic Steps" (*IPL*, Vol. 8 (1), pages 28–31, 1979), which shows that a RAM with unbounded size registers that can perform the standard arithmetic operations can factor integers in linear number of steps. In contrast to the current note, Shamir's paper has real technical contents (i.e., showing that $n!$ can be computed in $O(\log n)$ steps on such a RAM). Furthermore, unlike our definition of an electrical OR, it is less obvious that the unbounded RAM is unreasonable as a model of computation, since this model refers to computational complexity considerations when postulating that only (unbounded) versions of simple arithmetic operations are allowed. Still, the unbounded RAM model lacks in allowing transformations that depends on unbounded number of bits to take place at unit cost. On the other hand, the hazardous consequences in the unbounded RAM model are confined to unrealistic integer factoring algorithms, whereas the electrical OR model allows to compute any function.