The Church–Turing Thesis and Effective Mundane Procedures*

LEON HORSTEN and HERMAN ROELANTS
Center for Logic and Philosophy of Science, Department of Philosophy, University of Leuven, Kardinaal Mercierplein 2, B-3000 Leuven, Belgium (email: leon.horsten@arts.kuleuven.ac.be; cvl%gst%lw@cc3.kuleuven.ac.be)

Abstract. We critically discuss Cleland's analysis of effective procedures as 'mundane effective procedures'. She argues that Turing machines cannot carry out mundane procedures, since Turing machines are abstract entities and therefore cannot generate the causal processes that are generated by mundane procedures. We argue that if Turing machines cannot enter the physical world, then it is hard to see how Cleland's "mundane procedures" can enter the world of numbers. Hence her arguments against versions of the Church–Turing thesis for number theoretic functions miss the mark.

Key words. Church–Turing thesis, Turing machine, effective procedure, effectively computable function, mundane procedure.

There exists a large number of variants of the Church–Turing thesis (CT). Schematically, they all have the following form:

Every X Y is Z .

The position Z is occupied by a precise mathematical notion, such as 'λ-computable', 'Turing-computable', 'general recursive', 'Grzegorczyk-computable' (Grzegorczyk, 1957) . . . Y is a nominal expression. Candidates for it are 'function', 'process', 'procedure', 'rule' . . . If 'function' occupies the Y-position, then this function can then be further specified as being $N \rightarrow N$ (where 'N' denotes the set of natural numbers), or $R \rightarrow R$ (where 'R' denotes the set of real numbers), or $A \rightarrow A$ (where A is a relational structure of some sort) (Friedman, 1971), . . . X is an adjectival expression. Notions such as 'effective', 'effectively computable', 'mechanical', 'algorithmical', 'constructive', 'calculable' . . . occupy its place.

When philosophers are speaking of *the* CT, they refer to the "original" CT, i.e. what the founding fathers (Church and Turing) meant when they first proposed it (see Turing, 1936–1937). The idea is that even if there is vagueness or ambiguity in what they proposed, there has got to be something like a core meaning of CT. On the other hand, some variants are universally acknowledged to be properly speaking extensions of CT.[1] For yet others, the matter is not easy to decide. These are typically statements for which Y = '$N \rightarrow N$ function', and Z = 'Turing-computable' (or one of its famous equivalents).[2] So such statements typically differ from each other only in their X-position. The question arises whether (1) these X-terms all stand for the same concept (which can then perhaps be more or less suggestive of its correct analysis), or whether (2) they stand for different concepts,[3] in which case we have different theses. In this latter case, there is the further question whether these concepts are extensionally equivalent or not.

In her paper 'Is the Church–Turing thesis true?', Cleland looks at three versions of CT (Cleland, 1993, pp. 283–284):

CT_1 Every effectively computable $N \rightarrow N$ function is Turing-computable.
CT_2 Every effectively computable function is Turing-computable.
CT_3 Every effective procedure is Turing-computable.

In her discussion of CT_2, Cleland focusses on $R \rightarrow R$ functions, and in her discussion of CT_3, she concentrates on physical (as opposed to purely mental) procedures.

Clearly there is the following ordering of strength:

$$CT_3 > CT_2 > CT_1$$

She regards CT_1 as the "original" CT, whereas CT_2 and CT_3 are recognized to be extensions of CT.

Cleland doubts all three variants of CT, but acknowledges that the ordering of strength of evidence she presents is proportional to the ordering of strength of the variants she discusses. Her strategy is to formulate a new analysis of the *general* concept of *effective procedure*, rivaling the one of Turing (Cleland, 1993, p. 285). This will allow her to reject CT_3. Subsequently she investigates to what extent the considerations that lead her to reject CT_3, also allow her to cast doubt on its weaker relatives CT_2 and CT_1.

In Cleland's account, '*effective mundane procedures*', i.e. everyday procedures such as recipes, directions, . . . are taken as prototypical examples of effective procedures (Cleland, 1993, p. 286). She extracts the general features of 'effective procedures' from an analysis of these mundane procedures. Like Turing machines, mundane procedures can be formulated as "lists of instructions indicating that certain kinds of action are to be performed in a given order in time" (Cleland, 1993, p. 288). But in contrast to Turing machines, mundane procedures *generate causal processes* when they are followed (Cleland, 1993, p. 286). In other words, whereas Turing machines are working in abstract space, so to speak (they operate on abstract entities), mundane procedures are working in the physical world. And a mundane procedure is called *effective* if following it invariably results in a certain kind of outcome (Cleland, 1993, p. 291), namely in the intended physical outcome of the procedure (in the case of a baking recipe, the intended outcome could be a cake). This makes it fair, on Cleland's analysis, to call mundane effective procedures *causally* effective procedures. And it makes effectiveness of a procedure *relative* to what is taken to be the intended outcome of the procedure (Cleland, 1993, p. 294). In sum, mundane effective procedures are a two-tiered affair:

(1) effective procedures specify an *ordered list of instructions* that should be followed,
(2) effective mundane procedures have a *causal* effectiveness.

We should note that Cleland does not restrict behavior of performing lists of instructions to humans, or even to living organisms. For towards the end of the paper (Cleland, 1993, pp. 306–307) she suggests that a material object traveling through space can be seen as following a rule.

Cleland says that on her construal of effective procedures as causally effective mundane procedures, the question whether the versions of CT that she discusses are true becomes at least in part an *empirical* question (Cleland, 1993, p. 309):[4] we must investigate if there are procedures in the physical world (that meet some conditions, e.g. that compute $N \rightarrow N$ functions) that are not Turing-computable.

Now Cleland exploits her analysis of 'effective procedure' to reject CT_3. Her point is simply that whereas physical procedures operate in the physical world, Turing machines operate in an abstract world. *Causal* processes, yielding physical results, are operating in the physical world. Turing machines can only yield abstract results. So there are things that effective procedures can do which cannot be done by Turing machines. Of course it may be that *implementations of Turing machines* can do these kinds of things, but implementations of Turing machines are not Turing machines (by the definition of Turing machines!) (Cleland, 1993, pp. 301–302). And even if Turing machines can "simulate" all causally effective procedures, then at best there may be a structural correspondence of some sort between certain Turing machines and certain mundane effective procedures (Cleland, 1993, pp. 302–303). But again: that alone does not undermine the point that Cleland wants to make, for structural correspondence is not the same as identity. So her thesis still stands: CT_3 ought to be rejected.

Let us now look at CT_2 and CT_1. Even though she admits that she does not present knock-down evidence that they are in fact false (Cleland, 1993, p. 286), Cleland does think that there are good reasons to seriously doubt these theses too (Cleland, 1993, p. 304). Let us first look at CT_2. She proposes us to consider an example that goes more or less along the following lines. Take a Newtonian universe – let us suppose for simplicity that it is 2-dimensional: one dimension for space (**s**) and one for time (**t**) – and a material object (**o**) moving at unit speed in this universe. This situation is diagrammed in Figure 1 (see next page). This object **o**, traveling through space, can be seen as pairing places with times (Cleland, 1993, p. 306). So if we let these places and times stand for the corresponding real numbers, then the object can be seen as computing the identity function $f(x) = x$ from R to R (Cleland, 1993, p. 306). Now one can grant that this Newtonian universe is not our universe. But it does seem that our

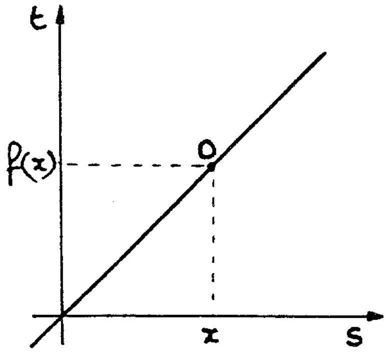

Fig. 1.

universe could have been like this. And it seems likely that whatever the exact physical-mathematical structure of our universe is really like, there is a good chance that the conditions needed for the construction of situations that can be interpreted as computing a real-valued function (such as continuity conditions and the like) do obtain in it. So we have good reasons to doubt CT_2.

Actually, we do not see why Cleland does not extend this argument to CT_1. Consider a device that measures the spin of an incoming electron. Such a device can be seen as computing a $N \rightarrow \{0, 1\}$ function, namely a function from incoming electrons (time-ordered) to spins $(+1/2, -1/2)$. If time is infinite, and the machine is left running forever, and after every point in time there will be electrons entering the machine (and perhaps some other not too far-fetched conditions have to hold), then the function it computes is total. Moreover, it seems at first sight extremely unlikely that the resulting function would be Turing-computable. Nevertheless, we do not want to exclude that physical arguments can be advanced to show that the result is after all Turing-computable. Working out the details of such examples is always tricky, so that it may be fair to say that there are at present no absolutely convincing examples of physical processes computing functions that are not Turing-computable (Cleland, 1993, p. 285).

In any case, for several reasons Cleland's line of reasoning is unacceptable. Let us look again at the example of the object moving through Newtonian space.

First, it is not clear in which sense **o** can be said to be *carrying out a* (*list of*) *instruction*(*s*). Perhaps "travel through space at velocity 1" is a formulation of the sort of instruction that **o** is following. And perhaps any similar instruction, as long as it is obeying the laws of motion, counts as an instruction that an object can perform. If that is so, then even an object at rest in absolute space and time is performing an instruction ("stay put"), and hence is computing a function. But we do not see in which sense this can be seen as an *action*, "as opposed to something which merely happens or is undergone" (Cleland, 1993, pp. 287–288). Actually it is not clear from her account to what extent all movements have to be seen as actions. For she considers the collapse of a bridge as a generic event which is *not* an action-kind (Cleland, 1993, p. 311, note 9). But cannot the collapsing of a bridge be seen as a complex movement? And if that is so, then what is the relevant difference between the collapsing bridge and **o** moving through space? There seems to be something to explain here. Perhaps Cleland's forthcoming theory of action will answer these questions. So let's put this issue aside.

A more serious point is that the function that **o** computes is not uniquely determined.[5] Even supposing absolute space and absolute time in her example, it seems that "absolute origin of the reference frame" is asking too much: there is no basis in objective reality for that.[6] But if we take a different origin of our reference frame, **o** computes a different function. If we allow galilean transformations of our reference frame, **o** computes even more functions. And why can't we accelerate reference frames any way we like, why can't we associate real numbers with places and times any way we like? If that is permitted, then any $R \rightarrow R$ function is computable (indeed, every such function is then *actually* computed!). But that would mean that the concept of computability, as applied to real numbers, becomes trivial. Cleland owes us a story about the restrictions on association of numbers with times and places. And those restrictions ought no to appeal explicitly or implicitly to any of the "mathematical" notions of effectiveness!

This line of reasoning can be extended to CT_1. If we look only at the positions associated with natural numbers in our space and time dimensions, then **o** computes the identity function from N to N. But if we perform a wild enough permutation of the natural numbers associated with the time-dimension, then the function that **o** computes becomes of an arbitrarily high recursion theoretic complexity.

But it gets worse. If the moving **o** does any pairing at all, then it pairs places with times, *not* real numbers with real numbers. For the same reasons as the ones for which Cleland so emphatically refuses the identification of Turing machines (abstract, mathematical entities) with physical machines, one should refuse the identification of physical entities (places, times) with mathematical entities (real numbers). In a slogan: if Turing machines cannot enter the physical world, then mundane procedures cannot enter the world of numbers (i.e. the mathematical world).

All the arguments that Cleland adduces to reject the former identification can be adduced to reject the latter identification. Let us illustrate this. For instance, one could reply that the structure of space and time *is* in some sense the structure of the real numbers. Therefore when **o** pairs times and places, it pairs real numbers. But saying that (the structure of space) = (structure of time) = (structure of the real numbers) can only mean that there is an *isomorphism* between these structures. But again, as Cleland herself emphasizes, isomorphism is not identity, times and places *are* not numbers, and identity is what we need for saying that **o** computes a *number-theoretic* function.

Cleland could bite the bullet and say that places and times *are* real numbers.

Then indeed **o** can be interpreted as computing an $R \rightarrow R$ function. For reasons that have been elaborately discussed in the philosophy of mathematics, we think that is not an attractive option. And even ignoring the well-known problem that there seems no *preferred* way of identifying times and places with real numbers, there is the question why, if real numbers occupy the physical universe, Turing machines (which are also mathematical entities) cannot also exist in the physical world. It seems that you cannot have it both ways. Either you make a strict separation between the physical and the mathematical universe (as we believe Cleland does), and then there seems no way that physical procedures compute on numbers. Or you somehow try not to make the separation so strict, but then you have to be very careful that your Turing machines do not enter the physical world. The main thesis of this paper is that this is the challenge that Cleland is facing.

Let us pause a moment and take stock. Cleland analyses effective procedures as effective mundane procedures. On this analysis, CT_3 is clearly false. This is so, *not* for empirical reasons (or one would have to take the term 'empirical' in a very broad sense), but for conceptual reasons ("abstract machines cannot generate physical, causal processes"). But if we are right, then on this analysis CT_1 and CT_2 come out true (and not "probably false", as Cleland suggests), for mundane effective procedures can *never* compute number theoretic functions (these are also *conceptual*, nonempirical considerations).

But surely this is a problematic outcome. It trivializes all number theoretic variants of CT (i.e. variants of CT which pertain to the effective, mechanical, algorithmical, . . . computability of *number theoretic functions*). Having a false antecedent, all such statements are vacuously true. In particular, this holds for CT_1. But the "original" CT surely has more content than this. And this can only be if unlike Cleland's notion of effectiveness, the notion of effectiveness that Church and Turing were using does not have a built-in component of causality. So rather than seeing Cleland's analysis as an improved analysis of the concept that Turing analyzed in terms of Turing machines, we propose to regard it as an analysis of a different concept. Causal effectiveness is a physical concept, whereas mathematical effectiveness, the concept analyzed by Turing, is an informal mathematical concept. Also, CT_2 is devoid of empirical content. This stands in contrast to the mainstream literature, where the interesting number theoretic extensions of CT are mostly taken to have empirical content, and to be hard to decide (see Earman, 1986 and Penrose, 1989). In sum, the thesis that physical machines can effectively compute number theoretic functions is regarded as obviously true, and rightly so (after all, that is what our personal computers do!).

So what went wrong? When CT and its extensions are discussed, then there is almost always an explicit or implicit assumption that what one is saying holds "up to isomorphism".[7] When we interpret a Turing machine as computing an $N \rightarrow N$ function, we are using the obvious correspondence between the structure of finite strings of symbols on the tape of a Turing machine and the natural number structure. Only if an assumption of isomorphism between the natural number structure and inputs and outputs of physical machines is made can these machines be said to compute number theoretic functions. Under such an assumption a personal computer can be said to *be* a Turing machine. Correspondence assumptions between physical quantities and finite strings of symbols are made when Turing machines are said to compute physical correlations. In sum, isomorphism assumptions are made in the mathematical interpretation of Turing machines, as well in relating Turing machines to the physical world. If one works up to isomorphism, then one can live in two worlds (the world of mathematics and the physical world). Suppose we grant such an assumption, and then read CT_1, CT_2 and CT_3 on Cleland's analysis of effective procedures as mundane effective procedures. Then CT_1 is an open question, as far as we know. CT_2 is still false, for the old reason: Turing machines cannot compute total $R \rightarrow R$ functions. But this has since long been recognized in the literature (see Earman, 1986, chapter VI). In response, proposals have been made to generalize Turing-computability. Even though there is no agreement over what the correct generalization of the notion of effective computability to the real numbers is, it seems to many that Grzegorczyk-computability may be a good candidate. Once a choice is made, there is a real empirical question. Something similar can be said for CT_3, although it will be very difficult to formulate a reasonable analogon of CT on this level of generality.

But Cleland explicitly refuses to make isomorphism assumptions. Her motivation for this is that she wants to ban Turing machines from the physical world. Our worry is that as an unwanted side-effect, her mundane procedures may be banned from the mathematical world, which is where CT was born, and in which it is still strongly rooted (even though the recent literature makes it clear that it has extended its branches well into the physical world). So our conclusion is that it is incumbent on Cleland to show how her mundane procedures can enter the mathematical world, given the fact that she does not allow the isomorphism-assumption. We do not exclude that it can be done, but it is far from obvious how.

Notes

* The first author is a postdoctoral researcher of the Belgian National Fund for Scientific Research. The financial support of this organization is gratefully acknowledged.

[1] For example: statements of CT in which Y = '$R \rightarrow R$ function'.

[2] For example: the statement "Every *mechanically computable* $N \rightarrow N$ function is Turing computable" (Gandy, 1980).

[3] This can even arise when two utterances express the exact same statement of CT, for sometimes the same expression can be used to express different concepts. We think that this is actually happening in the case of Cleland's criticism of CT. Her interpretation of the notion of effectiveness causes her thesis CT_1 to be a thesis that is *not* the original CT (cfr. infra).

[4] We argue below that this evaluation is incorrect if we take Cleland's arguments seriously.

[5] This in itself might not worry her, given that she allows the same causal process to simultaneously implement different procedures (Cleland 1993, p. 304). But there have to be bounds on the possibility of a process to implement procedures, and these are lacking in her account.

[6] The same holds for what we take as the direction of our space-dimension.
[7] These isomorphism assumptions often assume quite a bit of idealization, such as ignoring the internal structure of physical switches in a physical computer, ignoring storage restrictions of actual machines, . . .

References

Cleland, C. (1993), 'Is the Church–Turing Thesis True?' *Minds and Machines* 3, pp. 283–312.

Earman, J. (1986), *A Primer on Determinism*, Dordrecht, Reidel.

Friedman, H. (1971), 'Algorithmic Procedures, Generalized Turing Algorithms, and Elementary Recursion Theory', in R. Gandy and C. Yates, eds., *Logic Colloquium '69*, Amsterdam: North-Holland.

Gandy, R. (1980), 'Church's Thesis and Principles for Mechanics', in J. Barwise, H.J. Keisler, and K. Kunen, eds., *The Kleene Symposium*, Amsterdam: North-Holland, pp. 123–148.

Grzegorczyk, A. (1957), 'On the Definitions of Computable Real Continuous Functions', *Fundamenta Mathematica* 44, pp. 61–71.

Penrose, R. (1989), *The Emperor's New Mind. Concerning Computers, Minds, and the Laws of Physics*, London: Vintage.

Turing, A. (1936–1937), 'On Computable Numbers with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society. Ser. 2*, 42, pp. 230–265.