CSci • B607 **Problem Set #5** **Apr 17, 2001**

 Last edited Tuesday, April 17, 2001

On Effective Computability

Due — Tuesday, May 1, 2001 (midnight)

I. Effective Computability

A. Our analysis of the second, effective computability construal (EC) ended up in an argument that what is currently received as the official "theory of computation" is, in point of fact a mathematical theory of the flow of effect—i.e., something like a mathematical theory of causality.[1]

Laying out this claim (with its associated argument) is complicated by the fact that people come to the topic with very different assumptions. Some of the complexity one encounters in discussing it, that is, reflects a non-uniformity in general understanding of the issues involved. Some people assume computing is a purely *concrete, physical* phenomenon; others assume that it is a purely *mental* phenomenon; still others, that it is entirely *abstract.* And so on.

One way to understand the non-uniformity in the community is this; that the reconstruction we have been engaged in is much *finer-grained* than usual—with regard to all sorts of issues, but especially with regard to the three issues that constitute our investigative dialectics: (i) between issues of meaning and issues of mechanism; (ii) between things analysed concretely, and things analysed abstractly; and (iii) between phenomena at the level of the subject matter and phenomena at the level of the theory.[2] It is not just that traditional approaches aren't as careful to sort these issues out as (I argue) they should be; it is also that their avoidance of sorting them out, I believe, that has allowed people to *seem* to agree—in spite of approaching the constitutive issues from vary different points of view.

B. Identify just one such point of view, from which someone could understand the traditional account (i.e., could understand the traditionally-conceived theory of effective computability—Turing machines, computability limits, complexity classes, etc). This might be (a view held by) a former version of yourself, your current self (if you weren't at all convinced by the arguments given in class), someone you know, a position that you find expressed in the literature, whatever.

In a single paragraph (just a few sentences), describe this chosen person's view of the traditional theory of computing. I.e., describe whether they view computing, on the traditional analysis, as abstract or mathematical, concrete or physical phenomenon, mental or epistemic, etc. In

---

[1] Note that we argued for this conclusion in spite of the fact that no one *takes* it to be this way. That is: we didn't argue that people *think (or have historically thought)* that the theory of computability is a theory of causality. Such considerations about the *intent* of prior theorists was considered to be besides the point. Rather, what we claimed was that, *independent of what anyone has ever thought they were doing, in framing this theory*, what they were really doing was formulating a mathematical theory of effectiveness or causality.

[2] Another issue that lies under the surface—which will be the focus of the third (rule-following) construal—is that between "the one and the many" (e.g., between types and their tokens or instances).

doing this, say a word about what they think the fundamental computability limits are (i.e., whether they are theorems in pure mathematics, abstractions of physical limitations, conceptual limits to our insight or understanding, etc.).

C. Then, in your own words, write a 2–4 page letter (i.e., about 1000 words) to this person, explaining the new view of programs, effectiveness, computability, etc., that were advocated in Part III of the course. The virtue of having chosen a *particular* point of view, for your target audience, is that you can (if useful) directly contrast the new view with the person's prior understanding.

As far as possible, frame your analysis in terms of the three dialectics mentioned above. Try to be as clear and distillative as possible—i.e., so that your letter could be understood by someone (say, a professor in a computer science department or logic program) that has not taken this course.

—— *end of file* ——