

Part IV — Digitality

I. Preliminaries

A. Comments on last week
   1. Last time, we (hurriedly) finished our analysis of the 2nd construal: effective computability
   2. Of many things that could have been pursued in more depth, I wanted to mention three.
   3. First is an issue of theoretical status
      a. When I say that the "right" understanding of theory of computability is as a mathematical theory of causality, or of the flow of effect, and that because such a theory deals with only one of what we are assuming are computation's two constitutive aspects— mechanism, but not meaning—that doesn't meant that I think that what people learned in developing the so-called theory of computability is either insignificant or old-hat.
      b. On the contrary, I think it is of incalculable importance. As I said in a recent paper: "worth a passel of Nobel prizes." Also, I think it is *new*—not just old results in physics rehashed in new dress.
      c. The point is just that, as a pure general theory of effect or causality—as a "mechanical philosophy" it belongs, in the overall intellectual map, closer to physics, rather than being part of what I think of as the **intentional sciences**: those dealing with meaning, interpretation, truth, significance.
   4. Second, I wanted to say something about what I called the "motor theorem"
      a. It was misleading to use the word "theorem"; certainly I haven't proved any such result
      b. In fact we didn't even *formulate* a thesis to the level of exactness that would sustain a proof
      c. Instead, we should think of it as something like a "motor thesis": a hypothesis that given one motor (one source of energy or activity), and enough other static (but moveable parts), one can construct a physical device that, at an appropriate level of abstraction, "does the same thing" as any other physical mechanism in the world.
      d. Obviously, in order for this to be true, one would have to define the level of abstraction; seemingly everything hinges on that. But that is what links us back to the whole issue of effectiveness: the idea is to *identify such a level of abstraction*, at which these very general things can be said.
      e. Defining, carefully, what that level of abstraction is—and showing how it preserves a substantial sense of "effectiveness"—would be one of the tasks of developing the theory in the directions that we have been suggesting here.
   5. Third, I wanted to make a comment about the relation between numerals and numbers.
      a. One thing that was in the notes last time, but that we didn't have time to go over, was an ability, given our new reconstruction, to reclaim something that earlier had to set

aside: instances of *counting* or u*nary notation* as a "grounding" for mathematical representations of numbers.

b. In the first critique, we saw that counting (moving back and forth between exemplified cardinalities of numbers to numerals designating those numbers) involved a kind of "effective crossing of a semantic boundary" in a mathematical case.
c. That is, counting is a kind of *participatory mathematics* (arithmetic)
d. Now, if we understand the operations of Turing machines effectively, we can see that moving from a unary numeral—which, like Goodman's swatches, exemplifies the cardinality it denotes—to a binary numeral is an effective transformation.
e. So one starts to get a sense of how arithmetic designation is causally grounded (at least in this simple case).

B. Overall status report
 1. Given those comments, where are we, in the whole analysis?
 2. Summary to date
  a. Topic of first critique
   i. Interplay of our primary dialectic: meaning & mechanism
   ii. Right in saying that computing involves the two, inexorably
   iii. Wrong in saying that they are as independent as it claimed.
  b. (Positive) result of first critique
   i. Computers are **involved** in their subject matter
   ii. I.e., they are (actively) *participatorily engaged* with them, rather than just bearing a *detached semantic relation* to them.
   iii. This involvement is in the *realm of the effective*
   iv. The inside-outside and symbol-referent boundaries cross-cut, and neither is a moat
   v. Nevertheless, in spite of all these participatory morals, semantic properties may still be non-effective.
  c. Result of the second critique (concluded last time)
   i. The theory of effective computability is a mathematical theory of the **flow of effect**
   ii. A twentieth century theory of **causality**
   iii. Which is great—and useful for our project. But it isn't a theory of computing.
  d. With respect to the original fundamental dialectic, between the effective (mechanism, body) and the semantic (meaning, mind):
   i. First result: the two aspects are not as separate as all that
   ii. Second result: we know more about one of those two aspects (mechanism)
 3. Semantics
  a. The natural thing to do, at this point would be to present theory of semantics!
  b. I.e., figure out the other half of the dialectic
  c. That basic picture—that what stands in the way between us and a comprehensive theory, of the sort that we set out looking for at the beginning, is a theory or understanding of semantics—is where I was, about 15 years ago (in the early days of CSLI)
  d. So I set about trying to develop such a thing
 4. Metaphysics

a. I came to the conclusion that one could not develop a theory of *semantics* (representation, intentionality, etc.) without doing *metaphysics*
b. That is the project of the book I recently completed:[1] to paint a picture in which two subjects classically (formally) taken to be independent, in fact arise together:
   i. **Ontology:** what there *is* in the world; and
   ii. **Epistemology:** what it is to represent or 'register' the world (including semantics)
c. Note: this is not a claim that they are the same thing
   i. Rather: that the story of one involves the story of the other
   ii. I.e., they are *co-constituting*
d. So why don't we look at that story here?
   i. Because it turns out not to have anything specific to do with *computing*
   ii. I.e., it is a *general* metaphysical / intentional / semantic story

5. Computing
   a. Remember: the idea that there is (or will be) a theory of computation requires there being something *special* about computing—something about it *more particular* than just that it involves the dialectical combination of mechanism and meaning.

C. Plan
1. For rest of the main body of this course, we won't have more to say about semantics (or, for that matter, metaphysics and ontology)
2. In the last week, however (April 24 & 26) I may give a brief picture of the resulting semantical story—with maybe a bit of a hint of how it relates to metaphysical issues.
3. For the remainder of the course—the next two or three weeks—instead, we will turn to an aspect of computing that, as we've said from the beginning:
   a. Is critical to our current conception
   b. Seems (at least on the surface) *orthogonal* to the main mind/body dialectic
4. I.e.: the notion of *digitality*
5. So that will be—i.e., we now enter—Part IV of the course: an examination of the fifth construal: **digitality**

II. Introduction

A. Fifth construal

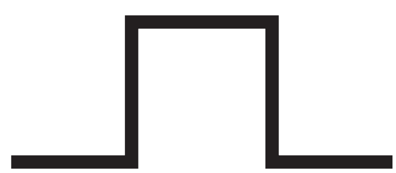

Figure 1 — Digitality

1. Since the beginning, we identified digitality with a construal on its own (DSM): that to be a computer is to be a digital state machine
2. Also since the beginning, though, I have suggested what should now be emphasized: that I will not so much take this to be a candidate for a theory of *what it is to be a computer*
3. Doesn't deal with the mind/body (semantic/effective) dialectic, for example
4. Instead, as already suggested, I take it to be a crucial *aspect* or *characteristic* of computing

---

[1] *On the Origin of Objects*, MIT Press, 1996.

5. So what we want to ask are questions like these:
   a. What is it to be digital?
      i. Why are so many computers digital?
      ii. I.e., why has digital computing been so successful?
      iii. These questions assume the existence of an independent (for now informal) notion of computing that isn't prejudiced as to whether it is digital or not
   b. What is digitality *good for?*
   c. What is it *not* good for?
   d. What are the prospects for non-digital computing?

B. Intuitions
1. There are two ways digitality is usually thought about, intuitively
   a. In a two-way contrast between **digital** and **analog** computing
   b. In terms of the **0s and 1s** in terms of which (so many) current computers are built
   c. We'll say a word about each
2. Wrt the digital/analog distinction
   a. The terms "analog" and "digital" are (in my view) somewhat unfortunate
   b. By rights, "analog" computing should mean a computation (computational process, whatever) that is an *analogue* of its task domain or subject matter
   c. *Analog* computing should ≈ *continuous* computing only if task domain is continuous
   d. Think of what we said about natural numerals and natural numbers
      i. In many cases (at relevant levels of abstraction), the "numeral" realm is almost identical to (or at least isomorphic to) the number realm
      ii. Seems like a good example of "analogue"
      iii. But we would never call it analog computing
   e. But it is hard to break tradition
   f. For the record
      i. I will by and large talk about "continuous" rather than "analog" computing, in setting up a contrast to "digital" or "discrete" (which I will take as approximate synonyms)
      ii. Also, will use two spellings:
         α. 'Analog' for what people normally call analog (i.e., continuous) computing
         β. 'Analogue' to mean computing that bears an isomorphic or resemblance relation to its subject matter.
   g. More generally, the issue here will be our first question

> **Q1** What are digital (discrete) systems good for? What are continuous systems good for? What are the constitutive properties of each?

3. Wrt. the digital substrate of 0s and 1s
   a. One of the questions on the first problem set had to do with whether the digital / continuous distinction (as we will now call it) crosses implementation boundaries
   b. As most people realized:

   i. It doesn't cross downwards: digital systems can be implemented on top of continuous ones
   ii. It *sort of* doesn't cross upwards: continuous systems can be *approximated* (in certain conditions) on top of a digital substrate. Or at least *some* continuous systems can be implemented on top of digital substrate. Or something like that.
  c. This generates the second question we will try to come to grips with:

> **Q2** What are the consequences of (reasons for? reasons against?) implementing on a digital substrate?

  d. The trick will be to keep the digitality at the level of the implementation, rather than assuming that because an implementing layer is digital, that what is implemented is (in any simple sense) digital
  e. Cf. Thelonius Monk recorded on a CD.
   i. There is something interesting about the fact that CDs are digital.
   ii. But (fortunately) that doesn't mean that the music, or his style of improvisation, or the growls or fades, attacks or quotations, *at the level at which the music is the jazz that it is*, are necessarily digital.
  f. As we will see (one of our main results) much of the press (cf. Negroponte's book—but theoretical press as well) makes the mistake of propagating digitality upwards.

C. **Fixity and fluidity**
 1. Another common (pair of) things assumed about digitality—ironically almost opposite:
  a. Digital systems are **fixed**—unchanging (whereas continuous systems are always likely to slide into a different or new state). A picture of endemic *stability*.
  b. Digital systems are **fluid**—they can be endlessly updated, changed, so that one can publish a Web page that is different for every different person who ever accesses it, etc. Nothing stable anymore. A picture of endemic *instability*
 2. As will be evident to everyone, there is nothing intrinsically contradictory about this. On the contrary, it is clear that the two somehow relate: the very fixity that digitality gives you seems to underwrite the fluidity that digitality *also* gives you
 3. Just want to keep this dialectic in mind, as our third major question:

> **Q3** What kinds of fixity—and what kinds of fluidity—does digitality confer?

D. One final introductory remark
 1. Literature
  a. Ultimately, I plan to write a volume of AOS on digitality (probably volume VI)
  b. But I haven't done so yet; in fact I don't even have a rough draft
  c. So this part of the course will depend on outside authors
 2. That literature divides into two camps
  a. *Philosophical* literature on digitality: Haugeland, Goodman, Lewis, Dretske
  b. *Computational* literature on analog (continuous) computing

3. We will focus more on the former (at least at first)
    a. Start with John Haugeland:
        i. Chapter 2 ("Automatic Formal Systems") of *Artificial Intelligence: The Very Idea*
        ii. The paper: "Analog and Analog" (one of my all-time favourite 10-page papers)
    b. Move on to Nelson Goodman
        i. Chapter 4 ("The Theory of Notation") of *Languages of Art*

II. Haugeland's notion of digitality

A. As will have seen from readings in Parts II and III, Haugeland defines
    1. Computers to be ⇒ formal (specifically: interpreted automatic formal system)
    2. 'Formal' to mean ⇒ "finitely-playable digital token-manipulation"
        a. Note: this is his *conceptual* version of the negative ("independent of semantics") reading of the formal symbol manipulation construal
    3. Digital to mean ⇒ *positive read/write techniques*
B. First, note three characteristic properties
    1. **Copyability:** flawless copying is possible
        a. E.g., Shakespeare sonnets—as opposed to Rembrandt paintings
    2. **Complexity:** interesting cases are complex or composite
        a. Cf. numerals, words, poker chips, computers
    3. **Medium independence:** (exactly) equivalent structures in different media
        a. Cf. chess: can play with wood pieces, metal pieces, on the computer, with people (à la Lewis Carroll), etc.
C. Look at definition
    1. Four parts
        a. Set of types
        b. Set of feasible procedures for writing and reading tokens of those types
        c. Specification of suitable operating conditions
        d. Given those, then procedures for read-and-write cycle are *positive* and *reliable*
    2. Positivity
        a. Positive procedure is one that can succeed absolutely and without qualification
        b. I.e., no ambiguity, vagueness, indeterminacy
        c. Not just *almost* perfect, or *pretty much* but *absolutely perfectly*, *exactly*
        d. Examples
            i. Getting a basket in basketball
            ii. Moving the pawn to K4
            iii. Cutting a board between six feet and six feet and one inch.
            iv. Writing a letter 'A'
            v. Making a copy of a text, or poem, or musical score
        e. Non-examples
            i. Shots in billiards
            ii. Cutting a board exactly six feet long

        iii. A sigh or a smile
        iv. Making a copy of a painting, or a performance of a musical score

D. Metaphysics
    1. The physical world doesn't (H. claims—we will return to this) support digitality directly
    2. So how do we have it?
        a. There is always some slop, variation, inexactness
        b. So how is digitality achieved?
        c. By allowing a "margin for error," within which all performances are equivalent and success is total
        d. Cf. poker chips (as opposed to piles of blue, red, and white sand)
    3. Comment
        a. Cf. voltages in a computer
        b. Not just that there is a margin for error
        c. Engineering requirement: *error* (discrepancy from the ideal) *doesn't propagate*
        d. Ragged edges of a pulse on a line: discrepancy from the ideal square wave don't pile up, so as to push this pulse, or the next one, or whatever, into the other camp

III. Analysis

A. We'll consider positive characteristics and criticisms of Haugeland's view on Thursday.

—— end of file ——