

The Digital Abstraction

I. Preliminaries

- A. First-order digitality
  1. Today we are going to conclude our analysis of first–order digitality
  2. This is not, unfortunately, because we have a final diagnosis or adequate theory. Such a theory has yet to be formulated (by anyone).
  3. But by the time we are done we will have seen a lot: what is at stake in something's being digital, answers to at least some of the questions with which we started, and a sense of the form that an ultimate solution will have to take.
- B. On Tuesday we will get to the notion of *higher-order digitality* (which I keep talking about).

II. First-order Digitality (review of Haugeland and Goodman)

- A. Haugeland
  1. Last week we looked at Haugeland's analysis of
     - a. Digitality as *perfection* ("positive read/write techniques")
  2. And we concluded
     - a. It wasn't a bad description of what digitality *gives you*, in this (messy) world;
     - b. But it was a better analysis of *what digitality is for* than of *what digitality is*.
  3. We noted that he analysed systems at a *single level of abstraction* (nothing special about *symbols* or *semantics* or *intentionality)*
     - a. In spite of our interest in computing as a semantic or intentional phenomenon, this "single-level" analysis fits our requirements better: understanding the way in which (some) computing is digital.
  4. Engineering
     - a. Finally, we concurred with Haugeland's engineering stance: that digitality is "*a practical means to cope with the vagaries and vicissitudes, the noise and drift, of earthly existence"*
     - b. On the other hand, we pointed out that Haugeland, somewhat ironically, doesn't explain the very property of how digitality manages to do exactly this: transcend the inevitable buffeting and decay of the world.
     - c. That is: Haugeland claims *that* digital systems accomplish this miracle—but he doesn't say *how* digitality does so (or, rather, what digitality really is—how it works—such that it manages this achievement).
     - d. So he seems to leave the major mystery unsolved.
- B. Goodman
  1. So we turned (on Tuesday) to Goodman, to see whether his analysis would supply us with a better characterisation

2. Goodman *does* analyse the digitality of *semantic* or *signifying* systems
3. He laid down a total of six characteristics:
   a. Three syntactic
      i. **SYN·1: Character (i.e., type) indifference**
      ii. **SYN·2: Syntactic disjointness**
      iii. **SYN·3: Syntactic finite differentiation**
   b. Three semantic
      i. **SEM·1: Unambiguity**
      ii. **SEM·2: Semantic disjointness**
      iii. **SEM·3: Semantic finite differentiation**

C. Analysis
1. What do we make of Goodman's characterisation?
2. There are several things to say.

III. Analysis of Goodman

A. First, since (as mentioned above) what we are after is a characterisation of digitality *per se*, three aspects of Goodman's analysis can be set aside (or at least set to the side)
1. Semantics
   a. The bottom line, wrt digitality in semantically interpreted systems, is this:
      i. *The digitality (discreteness) of the signs (symbols, representational scheme, intentional vehicles, etc.) of an interpreted system must always be distinguished from …*
      ii. *The digitality (discreteness) of what those signs (symbols, vehicles, etc.) signifies (i.e., the semantic realm, the subject matter, etc.)*
   b. Or to put the point more cryptically: one thing we can take, from Goodman, is the reminder that it is always critical (when analysing interpreted systems) to distinguish:
      i. **Digitality of form**
      ii. **Digitality of the content**
   c. E.g., the normal notation for the calculus or differential equations is: a *digital scheme* for representing a *continuous* (non-digital) *subject matter*
      i. This is extremely important.
      ii. Not only does it give us a leg up on analysing digital and non-digital signifying structures; it also helps us (as we said last time) with respect to a spate of traditional distinctions that are often made in discussions of representation: such as that between *linguistic* or *propositional* representation and *pictorial* or *imagistic* representation.
   d. Nevertheless, given our present goal, of understanding digitality in a way that applies equally to *uninterpreted* systems (like Lego and Lincoln Logs), it seems that we can break the semantic aspect of Goodman's analysis out—and take it as orthogonal.
2. Epistemic character
   a. It is also worth noting that there is no direct connection, in Goodman's analysis, between
      i. The application of his analysis to semantically interpreted systems; and
      ii. The lurking "epistemological" character of the analysis itself.

   b. In particular, in his characterisation of finite differentiation (more on this in a moment), he said that a system was finitely-differentiable just in case, for any two types σ_1 and σ_2 and token **s**, one must be able *tell* that **s** is not an instance of at least one of σ_1 or σ_2.
   c. This notion of "telling" was important: just as in Haugeland's case, it seemed that issues of epistemic determination or knowledge were creeping in, or lying under the surface.
   d. Whatever we might end up wanting to say about this epistemic character, though, it is important to realise that this epistemic character (of digitality) it is orthogonal to the resulting application of the notion of both sides of a relation of semantic directedness.
3. Ambiguity and Disjointness
   a. *Disjointness* (not allowing one token to be an instance of more than one type) is also something that, while intelligible for Goodman's purposes, seems stronger than necessary for characterising digitality per se.
   b. It has more to do with Goodman's particular project of defining *notation systems*.
   c. So that, too, we can set aside.
   d. Ditto for *ambiguity*. It is clearly interesting (and often important) whether a signifying system is ambiguous or not. But doesn't have much to do with *digitality* per se.

B. Given that, only two properties remain, in Goodman's analysis, that are relevant to our project:

1. **Type indifference:**
   a. All instances of a digital type are equivalent—for whatever purpose it is that you are analysing the system as digital
   b. I.e., the *flat top* (in the square wave icon or metaphor)
   c. Somehow a *digital typing* is one such that all tokens of the type are absolutely and unequivocally equivalent (for what purposes? more on that in a moment).
   d. This remains extremely important.
2. **Finite differentiation**
   a. The other property that Goodman highlights is that of finite differentiation
   b. Goodman's characterisation: for any two types σ_1 and σ_2, and a given token **s**, one can tell that **s** is not an instance of at least one of σ_1 or σ_2.
   c. Does this mean that digital types have sharp sides? I.e., that questions of instance or membership are determinate?
   d. I.e., does this requirement correspond to our "sharp sides" aspect of the digitality icon?
   e. It seems not, on the surface. For any given type (e.g., marks 12" & 24" long), according to Goodman, there may always be ambiguous cases.
   f. That is: it looks as if the finite differentiation requirement is *radically different* from the "sharp sides" requirement. One (sharp sides) says that there *won't* be matters of ambiguity about whether a token is a token of a given type; the other (finite differentiation) recognises that there *will* be matters of ambiguity.
   g. So there is a tension—between what Goodman is saying, and what we have said.
   h. What is going on?

C. Level-internal vs. level-external analyses

1. Haugeland claims that Goodman's analysis of finite differentiation is deceiving: according to him, Goodman is *confusing levels of implementation.*

2. Remember, Haugeland is more disposed to focus on *what the digital abstraction is like* than on *how it is realised or implemented*, or *whether it applies*.
3. In particular, given token **s** and types σ_1 and σ_2, part of what I will call a **digital registration** of a system (i.e., a system analysed at a digital level of abstraction), there are two questions:
   a. *Level-internal:* which type σ_1 or σ_2 **s** is an instance of
   b. *Level-external:* whether **s** is of type σ_1 or σ_2 at all
4. With respect to the first, level-internal question, Goodman and Haugeland both require that the answer be determinate (yes/no)
   a. But this starts to look familiar: as if we are back to where we were before.
   b. That is, when all is said and done, as regards what it is like *within* the digital registration, we have something like the following:

Haugeland:	digitality = perfection
Goodman:	digitality = determinateness

   c. These look very similar. The only problem, of course, is that they also look **circular**.
   d. The claim about definiteness, at a certain level of abstraction, is essentially one that issues be *clear and distinct*, with black and white yes/no answers.
   e. This seems to simply be *more discreteness*. But that is frustrating—and raises profound conceptual issues.[1]
   f. Is it really impossible to get at discreteness, without using discreteness?
   g. Does this relate to Haugeland's claim that digitality is a "fundamental metaphysical category"?
   h. While we probably are not going to be able to answer this question in anything like a final form, we will get deeper into it when we turn, next time, to the question of higher-order digitality.
5. With respect to the second, (level-external) question, Haugeland doesn't address himself, and Goodman allows the answer to be indeterminate.
   a. This ties into the issue of *implementing digitality on a continuous substrate*.
   b. And what we can see is that there is a very fundamental challenge, that neither writer has adequately addressed
      i. The bottom line (about digital systems) is not only
         α. That the digital registration *works*, once one has it—and works perfectly, from a level-internal point of view; but also
         β. That it is *practically possible*—it applies, even if not perfectly (in this messy world!) to perfectly buildable systems, *as close to perfectly as we require*.
      ii. That is: it is not as if, in designing digital systems, we have a form of perfection that, while complete and immune to failure if achieved, is nevertheless almost insurmountably *difficult to achieve*.
      iii. Rather, it is a form of perfection that is (as close to perfectly as we wish) *achievable*.

[1]We will start to explore these conceptual issues when we talk about the higher-order case, next time.

c. I.e., it *can* be achieved. Which means that the question of whether an underlying (continuous) state is an instance of a digital type should, in an unlimitedly high fraction of the cases we are interested in, have a definite answer! (else how could we be sure that it was in the proper digital state?)

d. And yet what is implicit in Goodman's recognition of the difficulties inherent in finite differentiation is that there will always be cases in which the question of whether an (underlying continuous) state is an instance of a digital type will be *ambiguous!* (He prohibits this being the case for *two* digital types at once, but seems to recognise that it cannot be avoided for *one* digital type).

e. So on the one hand we cannot have any level-external indeterminacy (in order to achieve perfection). And on the other hand we are inevitably going to have it?

f. What is going on?

IV. Digital abstraction

A. Introduction

1. While we aren't going to be able to answer this challenge completely, we can start to get at it, I believe (and in the process explain why in any digital circuit there are always two clocks, beating against each other).
2. *But note! The remarks in this section are particularly speculative and not-yet-worked-out!*
3. Basically, one can think of a digital registration as an *abstraction* or *partition* of the underlying physical state space.
4. Take figure 1 as a simple iconic diagram of this abstraction. The basic idea is that the underlying (continuous) physical plane is the space of possible (continuous) physical states, and each square represents a grouping together of an *area* or *region* of physical state types into *one* (single) digital state type.

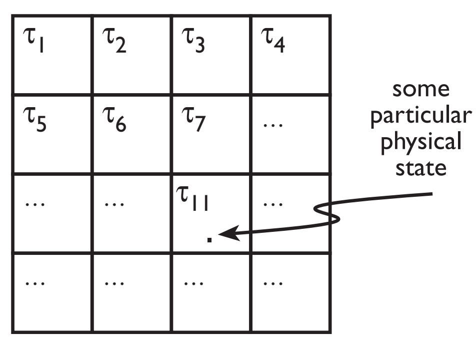

Figure 1 — A Digital Abstraction

5. At any given time, that is, the system will be in a particular physical state (indicated by the dot in the figure). Given that it is in that *physical* state, it can then be identified as being in the digital state type (τ_{11}, in the example).
6. So digital abstractions can be thought of as **abstraction grids** or **equivalence classes**, defined over the underlying (realising) physical states.

B. Perfection

1. The basic idea of perfection is as follows: it is assumed that there is some systematic set of regularities (i.e., some true description) that can be framed in terms of the digital types τ.
2. Although "at the bottom" the system is *really* physical and continuous, if the system is in some physical state σ which means that it is in digital state τ, then the future behaviour and digital state of the system (i.e., *in terms of the digital states*), can be totally and completed determined by the digital state it is in now.

C. Avoiding indeterminacy

1. But there is a problem
   a. If the system is in a physical state σ that is *on the boundary* between two digital states, or even in a physical state that is *on the edge of a single digital state*, then there is going to be ambiguity about whether or not it is in a given digital state (which will in turn undermine the perfection of the analysis in terms of the digital states).
   b. I.e., one can understand it this way: that there is going to be a problem if the state is allowed to be close to the edge.
   c. That suggests, in turn, that the (physical) state should be "kept in the middle" of each of the digital states.

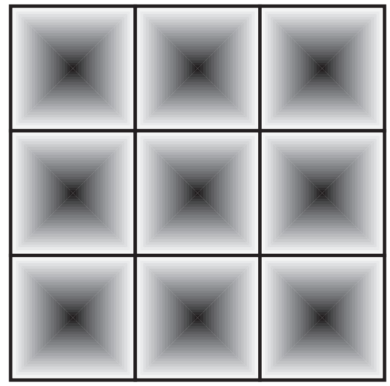

Figure 2a—Avoiding indeterminacy

2. So one might think that we should construct the system to be as indicated in figure 2a: where the *density of gray* corresponds to the probability of the system's being in the corresponding physical state.
   a. Except figure 2a has lines, on the boundaries, which are distracting. So perhaps a better diagram is figure 2b.
   b. Or perhaps what is depicted in figure 2c is better yet, where there is genuine separation (i.e., whole *regions* of separation that are "off-limits", digitally, rather than simply illegal "lines" between otherwise legitimate regions.

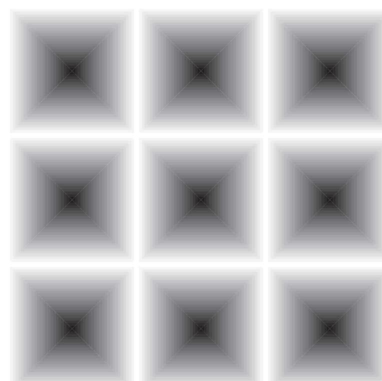

Figure 2b —Avoiding indeterminacy

3. Time
   a. If we are talking about a *static* system, with different *parts*—e.g., a system of letters or words, such that each piece can be determinately separated, and then one can ask, of each separated and determinate piece, which digital state it is in or realises (e.g., which words the pattern of ink represents), then the strategy indicated in figures 2b and 2c will probably work okay.
   b. But this strategy is not going to work for *temporal* systems—like computing.
   c. The problem is that, in figures 2b and 2c, the different density regions represent the physical state of different (physical) *parts* of the system. And so the fact that the regions don't touch (there are zero-probability boundaries separating them) is OK; that can be achieved for different static parts
   d. For a temporal system, however, each "re-

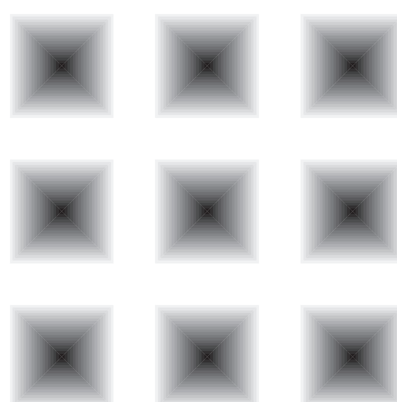

Figure 2c — Interstitial regions

gion"—each digital type—represents the *whole* system, at different times. And by hypothesis, the (underlying) system is *continuous*. So it follows that figures 2b and 2c are impossible—having the intermediate regions be zero-probability would mean that the system would *not be able to make transitions from one state to another*.

   e. For the system to be *real*, it will *have to move continuously from one region to another*.

4. Solution
   a. The solution, I believe—but this is just a suggestion; I have nothing like proved this—is that a successful temporal digital abstraction has to be *at least two-dimensional!*
   b. While in one dimension (of the underlying physical state) the system is moving across the boundary (from one legal region to the other), that dimension can't be used to determine the digital type!
   c. This is what we talked of before, when we were discussing the double clock in all digital circuits, which we analogized to walking: while one foot is moving forward, and can't be rested on, you have your weight on the other one. You transition between the two feet when both are on the floor—and stable.
   d. It seems that this kind of oscillation is necessary for any temporal digital abstraction.
5. This is a very important result, which any adequate account of digital realisation would have to deal with.

V. Multi-dimensionality

A. So far, we have been talking only about a given set of types
B. But in any given system there are *lots* of types, which can be discrete (digital) or analog (continuous) independently.
   1. E.g., an ordinary (analog) radio
      a. *Voltages* and *currents* (and time behaviour) are continuous
      b. But many other things are digital: such as the wires, or the devices themselves (finite, discrete number of transistors, resistors, capacitors, etc.)
      c. Don't have wires spread into *regions* of metal, or non-integral (1.3) numbers of resistors
   2. I.e., behind the continuous low- or object-level values, it seems as if there is a discrete conceptual structure (of components, in this case)
C. Similarly, we talked above about discrete questions (of identity), etc.
D. Next time, we will turn to these conceptual questions

—— *end of file* ——