CSci • B607 **Week #14 (a)** **Apr 17, 2001**

Last edited Monday, April 16, 2001

Higher-order discreteness

I. Review and summary: first-order digitality

A. As I hope was clear in last week's discussion, no one has yet laid out an adequate analysis of first-order digitality—the basic case—that answers the various questions we have raised:
   1. What digitality is;
   2. Why digitality is needed, in this (messy) world, in order to obtain the sorts of perfection that Haugeland talks about (and valorises);
   3. What exactly is required, in the general case, in order to "implement" (to any arbitrarily good approximation we please) a digital idealisation on a physical substrate;
   4. Whether the (manifest) impossibility of achieving perfection on the sorts of continuous substrate that we are given, in the physical world, is due to quantum mechanics, general considerations of abstraction, some other form of intrinsically epistemic limitation, or what.

B. Nevertheless, we are far from not having understood anything about this fundamental topic.

C. On the contrary, we have worked our way to some significant results:
   1. Even if we don't (yet) know why, digitality does seem to be a "fundamental metaphysical category" that allows perfection to be built up out of messy raw materials.
   2. Physical realisation
      a. We also have some sense of how digitality allows us to achieve this kind of perfection: as Haugeland says, it is a "practical means to cope with the vagaries and vicissitudes, the noise and drift, of earthly existence."
      b. That is: it allows us to impose a *buffer* between
         i. The system at the level at which we care about it; and
         ii. The system as it is affected by seemingly inevitable forces that cause it to drift, decay, rot, wander, and otherwise fail to stay perfectly in line.
      c. We saw, moreover, *how* digitality achieves this buffering trick: by imposing an "abstract grid" over the physical state space of a system, and then arranging to *confine* the inevitable (ontological) drifting around and the (epistemic) lack of certainty as to what state the system is within the bounds of this imposed grid.
      d. By doing this correctly, one can both (epistemically) predict and/or (ontologically) determine the future of the system (and, correspondingly, extract information about its past) in an absolute way, without any influence of the details of the details over which the digital idealisation abstracts.
      e. In sum, the digital abstraction keeps the inevitable drifting and buffeting
         i. *Below* the constitutive level of analysis, and
         ii. *Within confines*, so that it can never wreak havoc on the higher-level idealisation.

3. Costs
    a. One thing we didn't emphasize—but that can be predicted from the foregoing analysis—is that a great deal of the authentic detail that obtains, in the states of a physical system that is being understood in terms of a digital abstraction, is lost.
    b. That is: the digital idealisation, exactly by being an abstraction, ignores *a huge amount* (in the limit: an infinite amount) of the physical detail of any system that realises it. That is: digital implementation *poses limits*.
    c. One obvious question to ask, given this observation, is (i) *how much physical detail is thereby lost* (i.e., by analysing a system as digital), and (ii) what the benefits and costs are, of ignoring it.
    d. Some partial answers are evident:
        i. *Benefits:* by throwing all that detail away, you throw away *everything that is unpredictable or messy*. In that way, digitality provides benefits from "decay" and other forms of buffeting that are endemic aspects of life in our material world.
            α. What is gained, in recompense, is (epistemic and ontological) perfection.
        ii. *Costs:* there is a huge "power of computing" cost. Suppose—as seems realistic—that there are at least a trillion (10^{12}) bits of information in the measurable physical state of a system that is realising "one bit's worth" of a digital idealisation.[1]
            α. That means that, by taking a system *as* digital, one is throwing away at least a computational factor of 10^{12}.
            β. That is a pretty high cost to pay. But it goes to show how much the perfection is normally worth, to us.
4. Uses
    a. This explains something that Carver Mead has said, which we quoted at the beginning of this part of the course: that there are circumstances when analog computing is highly recommended, over digital computation, *because the information loads are so high*.
    b. Can we say anything about when those costs wouldn't be worth paying?
    c. Yes, we can. The answer is clear from what has been said, above.
        i. The times we might want to avoid paying the cost (of loss of computational power) are when the "bad properties" that we are avoiding aren't likely to be too severe.
        ii. In particular, we might be able to avoid paying the loss of computational power when
            α. There is reason to suppose that we have the input to the problem we are trying to solve in that highly dense, continuously encoded form; and
            β. When we *don't expect that the deleterious buffeting will be too severe.*

---

[1] Suppose that "+1", in some computer, is realised by voltages between 1.4 and 2.4 volts (i.e., 1.9 volts ± 0.5 volts), and that the clock rate of the computer is 1 GHz. And suppose, furthermore, that it is theoretically possible to measure voltages to within a nanovolt (10^{-9}), and times to a picosecond (10^{-12} seconds). Then in a sense the computer in question has approximately a trillion ($10^9 \times 10^3$) times more computational "horsepower" than is used, in the digital idealisation. The problem, of course, is that this excess horsepower can't be "controlled" in the appropriate way. But you can see why Mead is tempted, in cases where the "buffeting and drift" aren't too severe, to use the systems in analog form—and thereby regain something like that trillion -fold increase in computational power.

d. Both of these situations obtain exactly in cases of *direct perception.*

e. That is: it makes *perfect sense* for Mead to use analog chips in the circumstances of analog of visual sensory processing. For think about the problem that his chips are dealing with: of extracting information out of the incident optical and auditory stimulus.

f. Since, in general, this signal is very close to that which it is a signal of, there hasn't been much chance for the buffeting to get in the way—between sign and signified

g. We can expect that in similar cases, analog computing might be the recommended pathway.

h. It also explains why there is all sorts of current work on continuous (analog) computing fits in.

  i. Cf. Moore et al.: proofs that continuous Turing machines can "solve problems" (i.e., produce effective behavior) that is outside the scope of ordinary (digital) computers.

  ii. Cf. Jonathan Mills

5. Implementation

a. Finally, we pointed out the wonderful irony about digital systems: that although the digital abstraction (as we have just seen) is radically "higher-level" than the physical state description of the system that realises it, it is nevertheless radically *lower*-level than the level at which we care about that which is digitised (e.g., music, maps, photographic images, etc.).

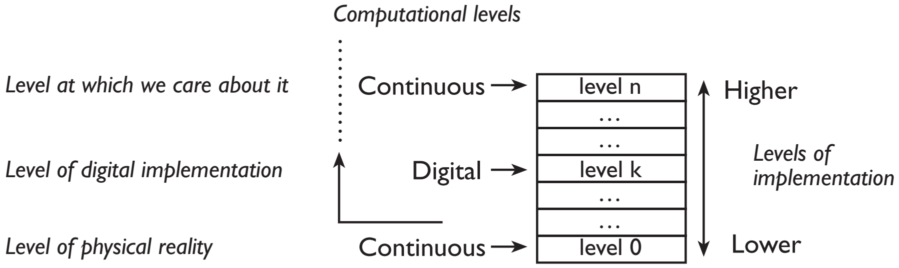

Figure 1 — Digital implementation (typical three-level structure)

b. That is, with respect to the implementation hierarchy (or "hierarchy of nature"), the basic picture one often aims for is three level, as given in figure 1.

c. Indeed, a great deal of the work, in digitisation projects, is to ensure that the digitisation is *invisible*—i.e., that what are known as "digital artifacts" don't intrude into the appreciation of the system *as what it is*, which is often as *continuous* (cf. type setting, fonts, images, sound, etc.)

d. What this means is that the *importance* of the "digital revolution" is, in a way, that we can *slip perfection in underneath the level at which we care about things.* It is this that has unleashed the radical power of the intentional era of computers.

D. Comments

1. Two final comments.

2. Theoretical status

a. First, I keep saying that we "don't yet have a theory" of digitality.

b. Given that pessimistic assessment, it may be worth noting that the summary given here (i.e., the one we have just rehearsed, in I.C.1–5, above[2]) is as much of an analysis of digitality as anyone else has ever produced!

- c. Not only that: many would *call* what we have delineated here a theory. In particular, we have as much of an account—and in fact a better account, I would submit—than either Haugeland or Goodman, or for that matter anyone else.[3]
- d. So when we say that our account is (nevertheless) "not a theory," it is important to understand that the demands we are imposing on what it takes to "count as an (adequate) theory" are unusually high. (But I believe that is the right stance.)

3. Dimensionality
   - a. Second, we also pointed out that, in general, systems that we analyse are *multidimensional*, and that in some of the dimensions, even in so-called "continuous" systems (such as analog radios), there are lots of digital or discrete dimensions (e.g., a discrete number of parts, a discrete wiring diagram, a discrete number of wires and connections, etc.)
   - b. This reliance on **discrete categorisations**, even when analysing continuous systems, is not only very telling, in its own right; it also meshes with the fact (which we noticed in looking at Haugeland's and Goodman's analyses) that the characterisations of digitality also relied—in what may ultimately be a viciously circular way—on discrete characterisations (such as the requirement that certain questions have "yes/no" answers, or that one be "able to determinately tell" whether such and such is the case.
4. What this all leads to is a double recognition:
   - a. Not only does digitality, and the notion of discrete distinctions, reach deep into
   - b. It also applies, as well as to computers, to our analyses of intentional (and perhaps other) phenomena.
5. This suggests that we back up one level, and look at discrete *concepts*.
6. Exactly this is the point of Haugeland's analysis of higher-order discreteness, to which we can now (finally) turn.

E. Plans

1. For the rest of today, we will talk about **higher-order digitality** or **higher-order discreteness** (and the lack thereof!).
   - a. This notion, I believe, has tremendous intellectual repercussions—not only for our analysis of computing, but for many things going on in science these days (such as the incredible popularity of the notion of emergence).
   - b. I have also put a short paper of mine, called "Indiscrete Affairs," onto the web site, which deals a bit with first-order digitality, but mostly with this higher-order notion
2. On Thursday, we will start our review of the whole course—trying to put together the morals we have amassed, with respect to our overarching goal: of understanding the conceptual structure of the terrain of computing.
3. Next Tuesday, there will be *no class* (I will be out of town).
4. The final class of the semester will be next Thursday, April 26, 2001.

---

[2] Plus the things we said, for example, last time—about the two-phase character of digital realisation.

[3] I have gotten into trouble in this regard in the past, over this point.

II. Discreteness at the level of concepts

A. First-order vs. second-order digitality
  1. Think of simple mechanics: forces, masses, accelerations, etc.
  2. The *measure values* of these concepts are continuous
  3. But the **concepts themselves are discrete**
    a. E.g., force of 74.184 newtons, mass of 2.941 grams, velocity of 55.392 meters/second
    b. But one does not have something which is a 1/3 a force, and 2/3 a mass. Or a concept that at the outset is force, but asymptotically approaches velocity
  4. At the first-order or object-level, one can imagine (especially field-theoretic) physics being a sea of spatio-temporally extended waves, with continuously-varying values
  5. But up one level, at the level of the concepts themselves, the image is much more like that in the opening scene of the movie 2001: silent, cold steel monoliths, perfect and homogeneous in themselves, with absolutely empty space between them.
  6. It is this which I having been calling **second-order** or **higher-order digitality**
  7. Field theoretic interpretations of physics, that is, are:
    a. *First-order continuous*
    b. *Second-order discrete*

B. Computation
  1. This distinction raises all sorts of interesting questions for computing

Q1 Are computational categories higher-order discrete?
Q2 Is 2nd-order discreteness implied by
  a. 1st-order continuity?
  b. 2nd-order discreteness (e.g., as in programming languages)?
Q4 Do 1st- or 2nd-order discreteness cross implementation boundaries?

  2. (To cut to the chase) The answer to all four questions, I will argue, is **no**.

III. Higher-order non-discreteness

A. Introduction
  1. Potential examples of higher-order non-discreteness (i.e., categories that are non 2nd-order discrete):
    a. Music: < *jazz*, *rhumba*, *reggae*, *fusion*, … >
    b. Psychology: < *chutzpah*, *bravado*, *ego*, *self-confidence*, *assuredness*, *pride*, … >
    c. Common-sense: < *morning*, *afternoon*, *evening*, *night* >
      i. If I say bring something over this afternoon, and you turn up at 7:30 p.m., did you bring it over in the afternoon?
      ii. The *answer isn't clear* (it certainly isn't 0.213 afternoon, or anything like that).
      iii. I.e., higher-order indiscreteness is not higher-order *continuity.*
  2. In fact we can say something stronger:

◆ Clear cases of *continuity* look as if they are *higher-order discrete*!

3. So maybe "continuity" is not the right opposite to discrete!
4. In fact maybe (reflexively) "continuous" and "discrete" are not *higher-order discrete concepts!*
    a. Cf. 'formal" not being a formal category

B. Computation
1. On the surface, it looks as if all sorts of crucial and constitutive computational *categories* aren't higher-order discrete
    a. *Object-oriented*
    b. *Efficient*
    c. *Distributed*
    d. *User-friendly*
    e. *Secure*
    f. Being *Fortran*, *Java*, *Postscript*, … ?
    g. *Computer?*
    h. …
2. Similarly, there seem to be an equal number of *non-(perfectly)-discrete distinctions*
    a. Data structure / programming language
    b. One implementation level / another
    c. Interpreted / compiled
3. More seriously, there are presumptive claims—e.g., that only "official" properties permeate abstraction-boundaries of virtual machines—that in practice don't seem to be discrete
    a. "Expert" programmers are those who know about underlying implementation decisions, and tailor the higher-level code in order to exploit "non-published" facts about underlying mechanisms.
    b. Cf. Kiczales et al. on meta-object protocols
    c. Cf. IRL work (Wegner et al.) on "grey box abstractions"

C. Implementation
1. This all ties into what we said before about digital implementation!
2. It is clear that higher-order discreteness does not cross implementation boundaries
    a. Cf. music (our standard example): just because a CD is digitally encoded, that does not mean that the music that is thereby recorded is discrete in any important sense whatsoever (first or higher order)
    b. Similarly, one can build a system (e.g., a binary implementation) of a file system whose higher-level properties—e.g. security, even whether it is a "standard" implementation of Postscript—are not discrete
3. This interacts with a result that we mentioned earlier, about the first-order case:

◆ What is important about digitality—digital computers, the "digital age," etc.—almost certainly has more to do with *digital implementation* than with *digitality per se*.

4. Earlier, when we discussed this moral, we talked about it in a first-order way. But now we can see that it may be just as—or even more—important, because of higher-order issues.

D. Science and vagueness
1. (Potential) non-discreteness of high-level computational categories is *very consequential*

2. For example, it challenges deeply-held assumptions about methods—even whether the field is a science—or, perhaps equivalently and certainly relatedly, whether they are amenable to being *analysed mathematically*.
3. Cf. discussions in epistemology and feminist philosophy of science, about the need to get past "binarisms" (sharp—i.e., discrete—dualities), such as
    a. Representation / ontology
    b. Mind / body
    c. Nature / nurture
    d. Concrete / abstract
    e. Male / female
    f. White / Black / Asian / …
    g. Etc.
    h. …
4. When we come back to summarise the whole investigation—computation next Tuesday; metaphysics on Thursday—this higher-order non-discreteness will play a very significant role

E. This tension between non-higher-order-discreteness and amenity to (traditional) scientific analysis generates a very, very interesting (and consequential) possibility:

1. Emergence
    a. Emergence is one of the hottest concepts in all of intellectual inquiry, these days.
    b. As we have said, here, before, it isn't clear what exactly an "emergent" property is—or why they are so intellectually popular.
2. However in this discussion there is a hint of an answer.
3. If (as we have said) higher-order digitality doesn't cross implementation boundaries, then it is clear that

◆ One can implement non-higher-order-discrete notions on top of higher-order discrete ones!

4. If that is true, then note that this provides a way in which one might be able to "have one's cake and eat it, too", as regards non-higher-order-discreteness.
5. In particular, one could

Emergence

a. Present a system that *at the implementation level* is perfectly higher-order discrete (and therefore amenable to scientific analysis; but
b. At the implemented level, manifests some non-higher-order-discrete property P;
c. Thereby ducking the issue of whether the *implemented* property P is "scientific"!

6. I am not claiming that this is all there is to emergence (I doubt if there is anything that is "all there is to emergence"). But I do think that it is at least one reason for the popularity of emergence.
7. Note, however, that if it is true, the consequences for the character of science are severe!
8. I won't talk more about this here, though we will come back to it in the concluding part of the course.

9. Note, though, that one might think the same for the notion of **formality**: that one could present or analyse a system that, at the implementing level, was *formal*, but in terms of which was implemented a non-formal property P (again, thereby, in a single blow, potentially satisfying two outright contradictory requirements).

F. Conclusion: for now, it is enough to summarize all this in a series of four morals:

M1 Many essential computational categories are not higher-order discrete.

M2 Higher-order discreteness (digitality) does not cross implementation boundaries.

M3 What matters about digital computing is not that the systems we build are discrete, but that they are digitally *implemented*.

M4 This digital implementation may not only be enough (i) to satisfy Haugeland's dictum (of providing a means to cope with the "vagaries and vicissitudes, the noise and drift, of earthly existence"—i.e., of providing protection from the gratuitous buffeting of the underlying continuity of the material world); but may also (ii) provide a way of providing traditional (mathematical) scientific analyses of concepts and categories and phenomena that, in virtue of not being higher-order-discrete, wouldn't on their own be amenable to (traditional) scientific analysis.

IV. Plan

A. With this, we conclude Part IV of the course, on digitality

B. On Thursday (with only two lectures left), we will turn to Part V—Conclusion—and try to bring together everything we have learned into some sort of cohesive summary.

—— *end of file* ——