Indiscrete Affairs[†]

Brian Cantwell Smith[*]
University of Toronto

Everyone knows that computers are digital. Or at least that *most* computers are digital. Sure enough, there are exceptions: analogue computers, resembling old telephone exchanges, for solving differential equations; ultra-modern analogue VLSI chips that mimic the human cochlea and retina; continuous Turing machines theorized in mathematical papers in computer science; incipient dreams of organic and quantum computers not based on zeros and ones. Still, the invention of the digital computer is widely taken to have been one of *the* major developments in the history of computing. Think of what came along with it: abstract symbols, universal machines, programming languages, data bases, digital controllers—and the internet. To say nothing of CDs and DVDs, personal computers, e-mail, mobile smartphones, electronic gaming, and virtual reality. Somehow or other, **digitality**—or **discreteness**, to use a term that for present purposes I will take to be equivalent[1]—lies at the core of the computer revolution.

More abstractly, computers' presumed discreteness, or "abso-

[†]An abridged version of this paper, with the title "Deconstructing Digitality," was published in *Idea&s*, a semi-annual magazine of the Faculty of Arts & Science at the University of Toronto, «ref yr & pp».

[*]Coach House Institute, Faculty of Information, University of Toronto
90 Wellesley St W, Toronto, Ontario M5S 1C5 Canada

[1]Distinctions between digitality and discreteness can and perhaps even should be drawn—but to do so would run beyond the scope of this paper.

luteness," plays a major role in our computational *Zeitgeist*. That computer science is a "formal" discipline, that computing is amenable to mathematical analysis, that computer science is a science—all these classifications rest on the premise that the appropriate theoretical concepts for studying computing have a formal, or discrete, character. Similar assumptions underlie the widespread view that computers are nothing more than dry and desiccated machines. Indeed, it is exactly the alleged contrast between the cut-and-dried, neat and sharp categories of the formal computational world, and the messy, contested, inevitably metaphorical and, ultimately, "wet" categories of human life-as-lived that drives the wedge, many people would say, between the monstrously mechanical and the sacredly humane.

But is that correct? Are computers, in fact, digital?

And what does "digital" mean, anyway? What would be it for the myth to be true?

1 Perfection and Protection

What does 'digital' mean? That is difficult to say[2]—but perhaps less difficult to picture.

As suggested in figure 1, two things are required. The diagram is essentially metaphorical—using a square wave as something of an icon of the more abstract conception of digitality. The first requirement, signified by the flat top in the middle, is that, to be digital, or to exemplify a digital property—i.e., to occupy the 'digital' region represented by middle square—requires a kind of

[2]Good philosophy of digitality is thin on the ground. The two main writers are Nelson Goodman (see for example chapter 4 of his *Languages of Art*, «ref») and John Haugeland (*Artificial Intelligence: The Very Idea* «ref», Introduction to *Mind Design II* «ref», and "Analog and Analog" «ref»). Of the two, Haugeland does a better job of articulating the consequences of digitality (reliability, resistance to degradation, support for perfect copies, etc.), whereas Goodman deals more with what it is to be digital—what something must be like, apparently, at least in this world of ours, to achieve the standards of reliability, unambiguity, copyability, etc. that Haugeland articulates. Goodman also distinguishes *syntactic* from *semantic* discreteness. But questions remain. In order for something to be discrete, for example, must there be a continuous background metric (spatial or temporal?) with respect to which the digital phenomenon is discriminated?

homogeneity or *internal uniformity*. If a computer is in a digital state (0 or 1, paradigmatically, at the "lowest" computational level[3]), then there is not supposed to be any state-internal variation: no matters of degree, no possibility for the system to be *partly* 0, or *mostly* 0, or *vaguely* 0, or *more-or-less* 0. The machine is either in state 0 or it is not—black and white, cut and dried. Everything is nice, determinate, and clean.

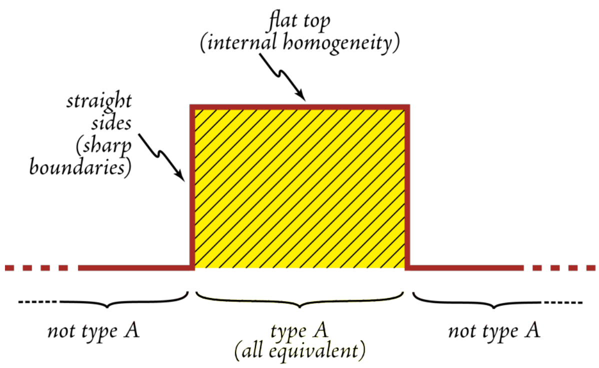

Figure 1 — Classical Digitality

So that is digitality's first aspect: complete (for the relevant purposes) internal homogeneity. The second aspect, signified by the vertical edges in the diagram, has to do with a digital state's boundaries: they must be absolutely sharp. Whether or not a system is in a given state—on or off, 0 or 1, yes or no—must be a totally and completely definite question. Either it is, or it is not, with no room for ambiguity or matter of degree. Systems outside the indicated region in figure 1 are not in state *A*, as surely and perfectly and absolutely as systems inside the region are in that state. Thus the structure illustrates what is never found in nature: an absolute, perfect, 90° cliff.

Needless to say, nothing in the real world is quite so neat. But that is all right. In fact that is why digitality is such a metaphysically powerful invention: it is expressly aimed to accommodate such cases. Departure from the ideal is not so much forbidden (which would be difficult to achieve, let alone sell for cents per gigabyte) as it is somehow, almost magically, rendered irrelevant. I.e., the idea is not that things are discrete in some absolute or ul-

[3]The traditional labeling of the two binary states in a computer is not without problems. They are normally be understood in terms of (or by analogy with) the *numbers* 0 and 1, though it makes more sense of elementary coding and arithmetic practices to associate them with binary *numerals* ('0' and '1'). For an exploration of these and related issues see Smith (forthcoming)—especially Volume ■■ (Digital state machines).

timate metaphysical sense, but that they are *fashioned so as sustain a digital level of description*. Rather than eliminating variation, which would be impossible, we build digital systems by arranging things so that the inevitable individual variation does not matter—such as voltages wandering up and down around some established standard (2.3 volts, 1.6 volts, whatever). To whatever extent is necessary, offending properties are cleaned up, boxed in, confined to certain limits, kept from spilling outside a certain protected region, so that errors do not accumulate or propagate, or the results get out of hand. The trick, that is, is to ensure, with respect to the overall or future state of the system—i.e., with respect to everything that matters about the system at the digital level of abstraction—that all present and future behaviour, such as whether the system will be in state *B*, depends only whether the system is now in state A_1 or A_2 or…or A_i, not on the *way* in which it is in one or other of those states. As long as that condition is met, then any potentially distracting variations will be locally contained—washed away, made invisible. As a result, the relation of the system to the (digital) property of being in state *B* is reduced to a single "bit" of information. Yes or no. On or off. Black or white.

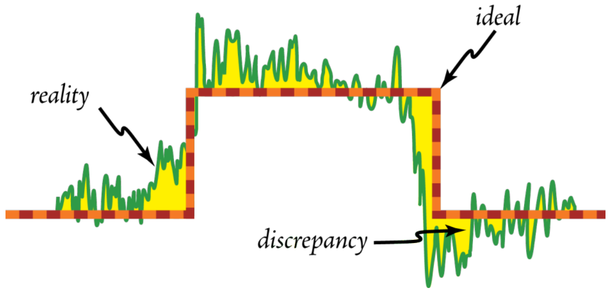

Figure 2 — Digitality as Ideal

You can see what is going on in figure 2. Taking an electrical pulse as paradigmatic, the green line indicates what the electrical circuit is actually like. The dotted red line indicates the "digital idealization." The yellow region indicates the "discrepancy" or "departure from the ideal"—the difference between idealization and actuality.

The amazing accomplishment, for digital systems, is that they are built to work *as if they were red, instead of what they actually are, which is green*. In constructing the rest of the system, that is, or in analyzing its behaviour, you can *assume* that it is red—in spite of the fact that the red line does not exist!

That digital systems can be assumed to be operating in terms of the digital ideal, instead of their concrete continuous messiness, which only approximates the ideal, is a much more impressive achievement than may be obvious—easily, in my view, worth a passel of Nobel prizes. It is certainly far from obvious that such a construction is possible. Normally, though idealizations in engineering are ubiquitous, discrepancies from ideality mount up in their impact. If you were to build a building with this kind of error between how it was supposed to be and how it was actually built, it would likely fall over. If it were a nuclear power plant, it would leak. Digital computer systems, on the other hand, are constructed so that—even with hundreds of millions of parts, changing states billions of times per second, there is not a single case in which, at the relevant level of abstraction, the discrepancies ever "push the system over the edge" into another digital state.

The crucial phrase in that last sentence is 'at the relevant level of abstraction.' Contrary to popular myth, the very lowest levels of computers, far from consisting of adamantine 0s and 1s, are not all that stable. Situations regularly occur where the implementing physical parameters get out of hand, wrecking any simple digital abstraction. Compact disks are a dramatic example: a fingernail scratch can leave a wake of devastation hundreds of bits wide. Cosmic rays and conveyor-belt motors at security checkpoints similarly can produce similar decay, to say nothing of a background slow drift and general disintegration in underlying materials. In a curious sense, in fact, modern digital media are more vulnerable than traditional non-digital ones. As is often pointed out, high-quality paper can last for hundreds or even thousands of years; disk drives are lucky to last ten. Optical media do better, but only somewhat, with current estimates of their longevity running only for a few decades.

How is the digital abstraction maintained, given these inevitable processes of dissolution? An extraordinarily impressive surrounding structure of routines and mechanisms prop up the digital abstraction. Compact disks employ staggeringly complex error recovery schemes to preserve and even recover the idealized digital "signal" in the face of catastrophic tracks of microscopic destruction. Laptop memory is rewritten every fifteen milliseconds, in order that rapidly accumulating "bit-rot" does not take over. Internet packets are checked and resent when they have eroded en route beyond the point of digital recognition. Disk headers are stored redundantly; fragile memories are backed up on disks; mission-critical applications are run in parallel on identical computers, in case one fails. The full gamut of such coding strategies and error recovery schemes is extraordinarily impressive. Certainly the popular idea that a visitor from Mars could examine a single CD and simply "read off" the music is a severe stretch, if not an outright error.[4]

What is digitality for? Why all the fuss? Why construct a system that—at least at this abstract level—is so pure, so crystalline, so fixed? Haugeland gives a particularly apt answer. Digitality, he writes, is:

> "*a method for coping with the vagaries and vicissitudes, the noise and drift, of earthly existence.*"[5]

Discreteness, that is, is more than anything else about **protection**—protection from the ravages and uncertainty and exigencies of the local surround. Things might get cold; winds might blow; the power supply might suffer a brown-out; moth and rust might corrupt; someone at the next table might say something distracting. If you are a digital system you need not care; your constitution guarantees that you will not be unseemingly buffeted by such local aberrations. You will not be unseemly at all, in fact. In a certain sense, digital systems are *intrinsically perfect*.

2 The User Experience

How do we experience the digital? At one level, the answer is obvious, or anyway familiar: we construct programs, automate processes, store data, send e-mail, post messages on social networking sites, interact with other users, manipulate "information." All of these things "exist"—are coherent and intelligible—at the digital level of abstraction. But that is not all. Something else we do, as

[4]For an account of how this is actually achieved, see «ref AOS volume V.»
[5]Haugeland, John, "Analog and Analog," *Philosophical Topics*, Spring 1981.

quickly as we have achieved the digital level, is do our best to hide it.

Think again about CDs—but this time, about the music. For example, think of a recording of Charlie Parker. Or of a compact disc of Thelonius Monk—of *Ruby My Dear*, say, or *In Walked Bud*, or *Straight No Chaser*. Or of a scanned original of a handwritten Walt Whitman poem. Or a recording of a late-night phone conversation with a lover. In each case, the medium or substrate will be digital in several respects: frequency, volume, hue. Yet, it does not follow—and this is the point—that the music itself, or the nuances of the image, or the inflection in the caller's voice, are thereby themselves rendered phenomenologically discrete. Rather, what these examples show is that you can implement or encode or represent something non-digital on a digital substrate, but *continue to experience it as continuous*.

top	· not digital	←	"user"
intermediate	· digital (discrete)	←	"computational"
bottom	· not digital	←	"physical"

Figure 3 — Three levels of abstraction

This fact—about the relation among one and the same system at distinct levels of description, only one of which is digital—may in the end be as important to the computer revolution as (or even more important than) the simpler fact that there is one level of abstraction at which most computers can be taken to be digital. [[…not just below…]] And then, "underneath" the digital abstraction, there is another physical level, at which the system or machine is again not digital.[6] The situation is depicted in figure 3. Even if it has grown familiar to the point of the banal, it is still amazing that we can construct a single system—one and the same "thing," a single patch of metaphysical reality—that can be analyzed, simultaneously and correctly, at three different levels of abstraction: (i) a top level, such as music, poetry, and the like, implemented (encoded, represented, constructed, etc.) on top of (ii) a "digital" level (the non-physical digital abstraction or idealization depicted as a dotted line in figure 2, which obeys the criteria

[6]More likely, it will be coherent or intelligible at the level of Maxwell's (continuous) equations.

of perfect discreteness), implemented, in turn, on top of (iii) a bottom physical level, at which it is again not discrete.

Arranging things in this triple-decker fashion simultaneously gives you the best of all possible worlds. It is fortunate that the lowest level, the level of the physical substrate, is not digital, since that means we can actually build things out of circuit components, metal parts, light guides, slightly varying components, and so forth—i.e., stuff made out of the messy, decaying, material clay supplied to us as the basis of all that exists. If we arrange that layer properly, however, mechanically and dynamically, we end up with a device that, at a higher level, *supports the digital abstraction*, with all of the resulting perfection discussed earlier: freedom from buffeting, protection from the ravages of time, insulation from unwanted or unwarranted influence. The astonishing part is that this protection from the world's dishevelment apparently extends upwards to all levels implemented on top of it. And yet—and this is the crucial part—this immunity of upper levels from buffeting and decay is accomplished without requiring that the higher level phenomena itself (the music, the meaning, the caller's sotto voce intimations) themselves be rendered (at least experientially) digital or discrete. In virtue of being "digitized," that is, the music, meaning and intimacies need in no salient way themselves be neatened, straightened up, clarified or disambiguated. No boxing on the ears is required in order to force them into the strictures of the discrete.

When we talk about "digitizing" music and art, in other words, strictly speaking we are using shorthand for "digitally encoding." To render the music itself digital would mean taking away from the Bird the ability to transform one melody continuously into another, or to build gradually from a whisper to a growl, or to have every performance of the "same" tune be unique. Fortunately, digital music does not require that.

Overall, I believe that the simplest way to understand the achievement of the digital age is in terms of figure 3's three-level structure. This is what our future rests on: an intermediate level of digitality, sandwiched between a lower, non-digital level of the brutely physical, subject to inexorable material buffeting and decay, and an upper, non-digital level of music, meaning, social praxis. Between the two lies the abstract, but terrifically consequential, intermediate, digital level, which, by virtue of its achievement of almost magical perfection, affords the upper level complete protection from the ravages of the underlying lower-level physics, thereby enabling arbitrary mobility, perfection and replication, without requiring that that upper level itself be digital.

The *protection* of the digital without the *price* of the digital—that is what the intermediate level provides to everything above it. Moreover, and non-trivially, given that we have the intermediate level of digitality, we can use it to harness the almost arbitrary powers of algorithms, programming, data, and information processing, in order to engender limitless patterns of transformation and interaction, configured so as to instill arbitrary creativity in the uppermost level.[7]

It is a three-level confection of historic power—with society, needless to say, dining out on the results. And remember: the different "levels" are not separate, modular pieces of an integrated whole. They are all the very same system or phenomenon, analyzed at different levels of abstraction.

… Figure out how to incorporate the following section into the foregoing (it is from a different version) …

3 Sustaining the digital abstraction

Are actual computers digital? Do they meet this ideal standard?

In one sense the answer is yes—but to a much lesser degree, and in a much more complex way, than is normally imagined. Contrary to popular myth, the lowest physical levels are not all that stable. Situations regularly occur where the implementing physical parameters get out of hand, wrecking any simple digital abstraction. Compact disks are a dramatic example, where a fingernail scratch can leave a wake of devastation hundreds of bits

[7]For example: digital "filters" and algorithms are now regularly employed, at the digital level at which music is encoded, to perform adjustments that are intelligible at the higher, implemented level—such as subtracting a soloist (for Karaoke), compensating for room acoustics, adding echoes or other fabricated artifacts, etc.

wide. Cosmic rays and the conveyor-belt motors at security checkpoints can similarly produce decay, to say nothing of a background slow drift and general disintegration in underlying materials. In a curious sense, in fact, modern digital media are more vulnerable than traditional non-digital ones. As is often pointed out, high-quality paper can last for hundreds or even thousands of years, hard disks are lucky to last ten. Optical media do better, but only somewhat, a best lasting a few decades.

Given these inevitable processes of dissolution, a surrounding structure of routines and mechanisms put in place to preserve—and prop up—the digital abstraction. Optical disks (such as CDs, DVDs, and Blu-Ray) employ phenomenally complex error recovery schemes so as to preserve and even recover the idealised digital "signal" in the face of microscopically devastating tracks of destruction. Internet packets are similarly checked and resent when they have eroded en route beyond the point of digital recognition. Disk headers are stored redundantly; fragile memories are backed up on disks; mission-critical applications are run on multiple 'identical" computers in parallel, in case one fails. The full gamut of such coding strategies and error recovery schemes is extraordinarily impressive. Certainly the popular idea that a visitor from Mars could examine a DVD, for example, and simply "read off" the music is a severe stretch, if not an outright error.[8]

In general, that is, the "digital" level of abstraction-the level at which two copies of the "same" CD are identical, for example—is higher (more abstract) than the level at which they are physical tokens. It also takes clever design and on-going work to maintain. This is one reason why different pressings of the "same" CD can sound different, different digital pressings of the same print look different, etc. We see and hear at the lower, continuous, physical level-and so we are vulnerable to what is digitally ignored: the ineliminable roughness, the necessity of approximation, contingent

[8]The situation is more than a little bit reminiscent of what has happened with regard to our understanding of DNA. Whereas it was first (mistakenly) thought that dna "contained" all the information about the structure of the phenotype, it has more recently been recognised that this idealisation is quite severely awry. Only within the context of a surrounding pool of RNA, proteins, etc.—all structures "encoded for" by the DNA itself, of course—can the "code" within the DNA be interpreted or effective.

particulars of the given concrete token.

4 Conceptual discreteness

From what has been said so far, you might take the conclusion to be this: that (i) while nothing is *physically* digital—i.e., discrete at the underlying physical level, (ii) we can, nevertheless, build physical things to sustain a digital (i.e., "computational") level of abstraction, (iii) on top of which we implement all kinds of non-digital things. Doing so gives these implemented things an unprecedented degree of stability and mobility—even virtual perfection. Society's slogan, on this view, should be "The Digitally Implemented Age," not "The Digital Age." And that is where things would stop.

It is not bad, as a first cut—but even it is wrong. And this time, it is a major falsehood—or perhaps we should say, an *expensive* falsehood. Getting over it will cost a great deal of the modern intellectual tradition.

The problem is that there is a more abstract form of digitality—what Haugeland calls "higher-order digitality"—that applies, not to the specific waveforms and measurable quantities of a concrete phenomenon, but to the very concepts themselves, in terms of which things are explained. Thus, consider force, mass, velocity, charge—staple concepts in physics. Specific forces and velocities can be as continuous as you please (23.759 kilograms, $0.3335640951981521 \times 10^{-8}$ seconds, etc.). However, the concepts in terms of which such things are analyzed are as pure, discrete and distinct as any digital states: nothing is ½ of a force and ½ of a mass, or partway between a momentum and duration. The concepts of physics are like the monoliths at the opening of the movie *2001*: unadulterated and distinct.

To make this concrete, I will call a concept **higher-order digital**, or **higher-order discrete**, just in case, to continue using the vocabulary form figure 1: (i) it is internally homogeneous, in the sense that there is no matter of degree, no "internal" structure, to its exemplification; and (ii) its boundaries are absolutely sharp, in the sense that whether or not something exemplifies the property is a clean, pure, absolute, binary, determinate, yes-no issue. These properties are to be contrasted with being **first-order digital** or

first-order discrete, which would hold in case the concept or notions takes the entities that fall within its extension to be discretely divided. Thus in classical physics, the notions of mass and velocity are first order continuous but higher-order discrete, since both masses and velocities can come in any real measure,[9] but as already noted there is no such thing as being somewhere between a mass and a velocity. The informal division of the day into morning, daytime, evening, and night, however, is a system of concepts that in contrast are first-order discrete but not higher-order discrete,[10] since they *do* divide the day into four discrete chunks, but not in an absolutely principled and dichotomous way; whether a given time is night or morning (such as in the early dawn light) is not an absolute question; it is not meaningless to say call such a time *partly night*, and *partly morning*.

For a more complex example, consider gender. "Being male" would be higher-order discrete just in case: (i) there were no facts of the matter, indeed no coherence to the idea, about *how* male something or somebody was; (ii) there was no internal structure to a given particular person's *being* male; (iii) if the "way that Andy is male" and the "way that Bill is male" were wholly interchangeable; (iv) just in case some things were male, and some things were not male, but no things—because of the verticality of the edge or boundary—were *ambiguously*, or *vaguely*, or *partially*, or *unstably*, or *contestedly*, male. And as the articulation makes clear, these absolutist criteria are not conditions that the (at least present-day) concept of gender meets.

By the same token, consider the notion of *arrogance*—and the boundaries between it and various nearby notions, such as *pride*, *egocentrism*, *self-confidence*, *braggadocio*, and the like. Once again, sharp edges do not apply. Nor is the issue just epistemic—an issue of uncertainty, of unclarity in the judging whether someone is one or other. More strongly, the point is that the concept itself is not—and could not be—sufficiently precisely determined for

---

[9]Remember that this is classical dynamics, not quantum mechanics.

[10]I do not say they are higher-order continuous. Articulating the conditions on conceptual continuity is a more difficult project than can be taken up here. One consequence of this way of analysing things, however, is already evident: that digitality (discreteness) and continuity are not precise opposites, nor do they form a mutually exclusive exhaustive pair.

there to be an exact metaphysical answer as to whether someone is arrogant or not. Moreover, the internal structure of arrogance is not uniform, either—implying that the concept is not internally homogeneous. People are more or less arrogant, arrogant in this or that particular way—in ways that make a difference, not only in general, but in particular *with respect to their arrogance*.

It might be thought that these examples are useful because of the ways in which they contrast with the computational situation. Computers, many people think, are distinctive exactly because, unlike people and perhaps other naturally occurring organisms, they *do* exemplify such perfected qualities: neatened-up categories, binary distinctions, clean edges. Many people think, in fact (including John Haugeland, in the paper cited above) that computers are *deeply* digital—not just made up ultimately of zeros and ones, in the sense discussed above, but much more generally that *whatever* properties computers have, in virtue of being computational, they have in a perfectly determinate manner. They either are push-down automata or not, universal or not, terminating or not. They either will or will not run Microsoft Word, are or are not connected to the internet, will or will not reboot after a crash. In no case—or so at least the official story claims—will the answer to a constitutive computational question be "sort of" or "somewhat" or "more or less."

It is exactly because of this presumptive (higher-order) absoluteness, moreover, that computer science is widely thought to be a formal discipline, that the study of computers is considered to be scientific, etc. At the same time, the same presumptive (higher-order) absoluteness is what makes computers, in many people's eyes, dry, desiccated, and inhuman. I.e., it is exactly the contrast between the cut-and-dry, neat, sharpened categories of the formal computational world and the messy, contested, inevitably metaphorical, and ultimately "wet" categories of human life-as-lived hat drives the wedge between the (monstrously) mechanical and the (sacredly) humane.

5 Computational categories

From what has been said so far, you might think that computers, *qua* computers—i.e., computers *at the computational level of de-*

scription—would all be digital, even if we use them as a substrate or vehicle or representation for other non-digital phenomena, from music to thunderstorms to politics to the digestive processes of T-cells. Or, to put the same point another way, you might think that all computational properties would be (higher-order) digital—clear, distinct, sharp-edged, as metaphorically intimated in figure 1. Not only *could* you think that; many people *have* thought it; I myself thought it, for many years. It is a very common view. But it is wrong. At the higher-order level we are now talking about, it is simply false that computational properties are discrete. It is a major falsehood, too—or perhaps we should say it is an expensive falsehood. Getting over it will cost us all of modern metaphysics.

To see why, it is useful to consider a variety of notions in terms of which computation is classically analysed. In each case, I will argue the same thing:

1. The intellectual mythology we have inherited, what it is fair to call the **formal tradition**, in terms of which we presently understand computers, has viewed this distinction (i.e., whatever notion we are discussing) as higher-order discrete.

2. In point of fact, however—in the actual, lived cases of what I will call **"computation in the wild**"—the distinction is not discrete.

3. Not only *is* the notion not discrete; it is *crucially* not discrete. The fact that the systems we build are possible, useful, realisable, interesting, and economically viable depends on the fact that the distinction in question is not, when you actually look at it, sharp-edged, cut-and-dried, determinate-i.e., is not a black-and-white yes/no affair.

These are strong claims, which ultimately require strong arguments. But it is not hard to develop an intuitive feeling for what is going on.

The problem is that actual computer systems deployed in real-world situations betray the fact that a large number of computational categories, in spite of being built on top of our now-familiar abstract form of discreteness, are more like arrogance than they are like mass. Consider four notions fundamental to the analysis of any real-world computer system:

1. **Subject/object**—and allied notions of representation/represented, symbol/referent, sign/signified, and so on
2. **Form/content**—syntax/semantics
3. **Inside/outside**—internal/external, intrinsic/extrinsic
4. **Abstract/concrete**

In each case, concrete, lived experience (rather than theoretical constructs built on assumptions to the contrary) shows that they are far from being neat and clean, "clear and distinct"—i.e., digital—concepts. That is not to say that these (or a host of other such) distinctions are useless, inapplicable or untenable. The point is just that, at best, they demarcate a complex, intermediate region or territory—not a "gradual" or "continuous" or "smooth" compromise, but rather a turbulent locus of ferment and activity, a place where things are stretched and pulled and splintered into a thousand other considerations, considerations that no longer line up and pull in one direction, nor line up and pull in the other, but sunder, cross-fertilize and lead to more distinctions—all the way (as it is said) up to "the edge of chaos."

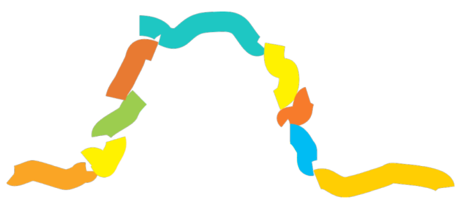

Figure 4 —Boundaries breaking

Ultimately, instead of being discrete, the situation begins to resemble that depicted in figure 4.

Start with the three distinctions listed in figure 5: (i) between a symbol and its referent, (ii) between syntax and semantics; (iii) between the inside of a system and the "external world" in which it is embedded; and (iv) between things that are abstract and things that are concrete. All four are implicated in the analysis of any interpreted or representational system, including not only computers, but also people, and at least arguably such other things as language, books, and e-mail.

The first distinction, between symbol and referent, gets at the ineliminable fact that any interpreted or intentional system describes, represents, encodes information about, or is in some other way "oriented" towards a task domain or subject matter. The second, the more abstract or conceptual split between syntax and semantics, separates concerns about how a system works (i.e., issues about "form," material embodiment, and causal effectiveness) from more distal or interpretive questions having to do what the symbols mean or represent. The third, between inside and outside, is in some sense even more basic: it is implicit in the very idea that the system is a system or entity at all. What is "inside" the body or skin or rack panel constitutes the system itself; what is outside is labelled the context, or environment, or external world. And the fourth …

	Primary	Allied
1.	*symbol* , *referent*	sign/signified name/named representation/represented
2.	*syntax* / *semantics*	form/content
3.	*inside* / *outside*	internal/external intrinsic/extrinsic
4.	*abstract* / *concrete*	

Figure 5 — Traditional distinctions

Again, all four distinctions are very general, as applicable to people and human activity as to any conceivable artificial mechanism. What makes computers special, however, according to the logical and metamathematical traditions from which computer science has inherited its explanatory frameworks, is that these distinctions are thought to apply to computers *in a distinctively discrete way*. First, the inner symbols themselves are thought to be discrete. Second, the categories in terms of which they are analysed, such as meaning and semantics, are thought to have exactly the sorts of sharp boundary and internal homogeneity discussed above. Third, the three distinctions are taken to be aligned, with the symbols, especially with regards to their syntactic aspects, imagined as being on the inside of (and thus part of) the computer; and the referents, implicated in the semantics, on the outside (and therefore not part of it)—again, in a neat and uncontentious way. Fourth, the inner realm of symbols is taken to be abstract, in contrast to the presumptive concreteness of the external realm of referents. Fifth, and finally, the divide between the two realms—between the pure, inner world of discrete, abstract symbols, and the messy external world of concrete referents—is viewed as something of an explanatory moat: a gulf across which theoretical dependence does not cross. Moreover—to put the icing on the cake—it is exactly in virtue of allegedly having this neat overall structure that computers are taken to be scientific: amenable to rigorous, mathematical analysis.[11]

6 Computation in the wild

But is it true? In practice, is the computational realm so neat? No, it is not. And the reasons cut deep.

… figure out what is first-order, what higher-order? does it matter? i think so…

To see why, consider what is perhaps the simplest imaginable counterexample (too simple, perhaps, to convince anyone—but maybe still illustrative): an elementary case of counting. What counting illustrates, in a way that doing sums does not, is a computational process that *actually interacts with its subject matter*—namely, with an exemplified situation of some number *n* of objects. When you count five elements in a list, you end up with a representation or numeral '5', designating five. But what you start with is an actual *number* of elements—not a "*numeral*" of elements, a phrase whose very awkwardness betrays the fact that it makes no sense.

And counting is just the tip of the iceberg. As all practitioners

[11]The impact of this alleged divide gets carried over into other fields. Thus cognitive science, based on the hypothesis that minds and intelligence are computational, makes an analogous distinction between: (i) "narrow," brain-oriented, psychological phenomena, assumed to be wholly mechanistic, intrinsic, mathematically analysable-the subject of scientific psychology; and (ii) "broad," social, relational, allegedly non-psychological phenomena, usually left to sociologists or anthropologists or historians. Many critics have argued that these assumptions do not hold in the human case—and thus that people must not be computers. The claim in the text, however, is that they do not hold of computers, either.

know, it is impossible to separate computers from the worlds they represent. Computers are so involved in their task domains, in fact, that it is impossible to sort their interaction into the traditional categories of reason, action, and perception. It is not even enough to generalise to a broader notion of experience. Just think of e-mail, of file systems, of network traffic nodes, of display cards and window systems and run-time compilers. Computers **participate** in their subject matters: they muck around in, create and destroy, change, and constitute, to say nothing of represent and reason and store information about, a hundred realms—new realms, some of them, that owe their existence to the very computers that interact with them. In fact computers are so thickly engaged in their subject matters that it can even be impossible to draw a stable inside/outside boundary. Are the windows on the desktop inside or outside of the computer? What about the disk drive? the file system? the backup tape? the network?[12] Similarly, the boundary between sign and signified, and the corresponding theoretical boundary between syntax (in the generalised sense of the realm of the effective) and semantics (in the similarly generalised sense of a distal realm of that with which computer systems are normatively enjoined to coordinate) is about as far from sharp as it is possible to be. The two sides interpenetrate, not so much in gradual shades of gray as in a profusion of middling, "hybrid" intercalations.

And so the situation, instead of being discrete, begins to resemble that depicted in figure 3 ⟦4?⟧. At least with respect to these first three classical distinctions, that is, real-world in vivo boundaries are far from being clean and sharp. At best, the three notions demarcate a complex region or territory—far from being even "gradual" or "continuous", but rather a locus of ferment and activity, a place where things are stretched and pulled and splinter into a thousand minor considerations, considerations that no

[12]Computers "shake hands" in the same medium as that in which they think. It is as if we humans, upon encountering a friend, could plug our nervous systems together directly—i.e., as if we had "ports" on our nervous systems—without having to transmit the signals through a different underlying medium. Perhaps, if we had developed to perform such feats, the individuation criteria for people would be as messy as they are for modern machines.

longer line up and pull in one direction, nor line up and pull in the other, but sunder, lead to more distinctions, and may even be best described as on the edge of chaos.[13]

With respect to this moral, moreover, there is nothing special about these first three distinctions. Much the same story holds for any number of other constitutive computational properties. Thus consider *abstraction*. For many many years, it was assumed that the way to build complex systems was in terms of so-called "black boxes"—abstractions that presented a fixed and given interface to the outside world, but that completely hid within themselves all internal "details of implementation." As usual, the idea of discrete black-box abstraction had a certain theoretical appeal. But in practice it, too, has turned out to be an unworkable idealisation. As every professional programmer knows, no matter how elegant the formal or explicit interface to a virtual machine, inner implementation details invariably "shine through" and affect the systems built on top of them, in ways that often have dramatic effects on performance. To make a program run fast, that is, you don't just need to know the formal definition of C++; you also need to know (or have experience with) how it is implemented. This non-opacity of abstraction boundaries is even gaining theoretical recognition, leading to the design of fancy mechanisms that allow programmers access to the "innards" of the underlying level. Some have even suggested replacing the notion of a black-box with something like "gray box" or "glass box," in order to legitimate making the workings of the lower level visible.

In the wild, that is, what had been theoretically allegedly to be a fixed, discrete boundary turned out in practice to be something quite different: a locus of negotiation, of communication and sharing of advice, a region rather than a line, where responsibilities and information are exchanged—far more like a market or town square, where consensual agreements are hammered out and maintained in real time as things progress, rather than the pure line of fixed abstraction that the intellectual heritage imagined.

[13]«. Reference, if it is possible to do so coherently, some of the Santa Fe work.»

7 Logic

And so it goes—to deeper and deeper levels. Not only do specifically computational properties fail to be discrete, as we have seen, but the same moral applies to more general distinctions of which computer systems are sometimes used as models: between nature and society, between the sciences and the humanities, between subject and object, between mind and body. Computers are wonderfully disruptive precisely because they make a sham of the ultimate sharpness of every one of these classical dualisms.

Computers are symbol manipulators par excellence, for example, but does that mean they validate those who claim that language is merely an endless play of signifiers? No, they do not; they spend too much time mucking around in their (semantic) task domains. Computers are supposedly objective and natural, or at least naturalistically palatable—i.e., scientifically OK, intellectually respectable, not too spooky. But the stories we tell about them are so thoroughly peppered with intentional vocabulary (*programming languages*, *data bases*, *information highways*, *knowledge representation*, *symbol systems*, and on and on) that this alleged "respectability," intuitively reflected in the claim that computers are "mere machines," may ironically turn out to be sheer prejudice. It is particularly curious that at the very same time that their alleged objectivity recommends them, philosophically, as naturalistic (i.e., as one with the sciences), at the very same time they are candidates for a theory of what it is to be an intentional subject, because of their manifest representational character.

The failure of discreteness even applies to some of the most foundational distinctions on which all of logic, mathematics, and science are thought to rest: existential distinctions, between and among objects themselves, and logical distinctions, such as that between objects and the properties or types they are taken to exemplify. Formal logic, mathematics, science, and a good measure of modern philosophy, in particular, not only presume a background of objects with precise black-and-white individuation criteria, but even more seriously assume that the goal of scientific discourse is to delineate the objects, categories, and properties in the world so that their boundaries are higher-order discrete, in just the absolutist sense we have been wrestling with. It is this, I believe, that computational experience had shown us, and will increasingly show us, to be an impossible, out-of-date, and ultimately futile game.

At a workshop on representation a few years ago in England, a philosopher argued that philosophy, taking a lead from science, should insist on a very strict notion of object (on clear definitions, precise identity criteria, and the like). As a working scientist, I could only muse that in two decades of wrestling with the essential structure of computing, which is at least a candidate for the most important scientific, let alone intellectual, development of the twentieth century, I had never found any such distilled, lapidary objects. The identity criteria on computational objects simply do not honour this formalist ideal. Think about the property of being an x86 microprocessor,[14] necessary in order to run Microsoft's Windows operating system. Enormous effort goes into defining the exact operating specifications of such commodity chips. And yet numerous issues about what it is to be in this class remain unanswered (as clone manufacturers are continually discovering, to their dismay). Nor is there any reason to believe that the answer is temporal stable. This is the realm of copyrights, patents, and million dollar lawsuits. Among other things, the answer depends on who is asking. For it is widely recognised that to be a legitimate instance of a particular architecture is in part a commercial and political question, involving issues of market share, advertising power, and the like. No two runs of a single chip design are absolutely identical, let alone explicitly different versions, or allegedly the same version from different manufacturers. Even within single companies, in situations when all the market forces press for a common chip type, it turns out in practice to be impossible to guarantee that unresolved boundary cases will not emerge. And what is true for hardware is doubly true of software. That is why software is maintained; it takes money, power, and influence to preserve the identity of a program over time.[15]

---

[14]The processors that power most personal computers—from the Intel 8086 through the Pentium up to present day Core i6s and i7s, and similar offerings from AMD.

[15]Curiously enough, moreover, very much the same conclusions—about the lack of strict individuation criteria, and the concomitant breaking up of the object's boundary—arise in even simple cases of arithmetic.

I once designed a programming language that, unusually, attempted to

Needless to say, what is true of computing is even more true of human experience. Suppose, for example, on a camping trip, after gazing at the sky, that you turn to your companion and say "we probably shouldn't attempt the ascent today; there are clouds covering the north side." And suppose, further, than your friend, having nothing better to do, asks the following pedantic question: "OK; you've been to college; how many clouds are there, exactly?" Your inability to answer cannot be ascribed to merely epistemic doubt or lack of knowledge. Nor does it mean there was anything wrong with your original statement. There is no reason to suppose that there need be any metaphysical fact of the matter—any metaphysical fact as to whether, in some region of the sky, the arrangement of foggy air should count as one cloud, or two. If, as I believe, this is right, then it must be that the competent use of the English plural does not metaphysically require a set of discretely countable individuals in order to be true. The same would apply to a claim that "a program still has bugs." The truth of that statement does not depend on there being strict individuation criteria on bugs—and bugs are surely as computational a concept as

---

maintain strict use/mention distinctions among (i) numbers, (ii) internal structures that designate numbers (internal 'numerals', essentially), (iii) external expressions (like '234') corresponding to those internal structures, (iv) distinct copies of those internal structures, (v) pointers to those individual copies, (v) and so on and so forth. By the same token, a similar set of distinctions was made among sequences, internal structures that designated sequences, external character strings that notated the internal structures that designated sequences, etc. Not only did these distinctions crosscut; they were in turn crossed with several other familiar sorts, such as between types and tokens and instances and uses. To what end? Total confusion! The result was impossible to use.

Semantical clarity, or at least something resembling it, was obtained at the expense of sanity. It turns out that what one wants—and as common sense anyway suggests, at least on reflection—is a system that makes *whatever distinctions are appropriate, in the moment, for the purposes that at that time are being served*. Distinctions need to be made on-the-fly, in response to particular circumstances, not inflicted, as if that were even possible, all at once, at the outset. In real systems, that is to say, with anything approaching the complexity of modern software (note that even Xerox copiers now have multiple millions of lines of code), individuation criteria, and thus object identity, are themselves context-dependent, negotiated, and maintained.

Not even ontology is sacrosanct.

one could please.

The distinction between type and token is similarly crumbly, in lay experience. It is not just that the traditional two-way distinction is not adequate—between abstract type or category, on the one hand, and concrete token or instance, on the other. Nor is it enough to spawn a three-way distinction among type, token, and use—or even one able to deal with more complex cross-cutting spatial and temporal fan-outs of interpenetrating abstraction.[16] Instead, imagine getting up one morning and saying, drearily, "oh no, I still have a headache." Or: "the fog is coming back." Or: "the wind from that direction is typically warm." How are we to understand the referent of the singular noun phrases: 'a headache,' 'the fog,' 'the wind'? There is no reason to suppose that they refer to *types*, in the sense of something that can be "tokened." Nor is there any reason to suppose that they refer to *tokens*, in the sense of something that is of a type. Rather, there is no reason to suppose that the distinction between type and token, or between object and property, in the lived world, is any more of a "discrete" way, with any more sharp and absolute and black-and-white boundaries, than any of the others we have already seen.

Jericho once again. As in figure 4, the boundaries start tumbling down.

Ultimately, in fact, it is wonderful historical irony. Computers are supposedly objective, scientifically "OK"—intellectually respectable, naturalistic, not spooky. It is in virtue of this pedigree that they are *echt* denizens of the modern academy. But this al-

[16]Consider a simple program for computing factorial:

```
procedure factorial(n)
   if n=0 then 1
      else n*factorial(n–1)
```

Suppose that this program is called with the argument '5'. With respect to different readings of the term 'the variable n', there is something of which there is one, something (*spatial* fan-out) of which there are three, something (*temporal* fan-out) of which there are five, and something (both) of which there are fifteen.

leged respectability, so innocuously garbed in the idea that computers are "mere machines," may turn out, historically, to reflect no more than sheer prejudice.

Loosed into the wild, computers play the trumpet outside the digital walls of Jericho. The boundaries of conceptual discreteness are tumbling down.

Why does it matter whether the digital level of abstraction is "real"? That much of what we call digital is neither physically nor experientially digital, but only digitally implemented? That the concepts and categories of computing are not conceptually discrete?

In part, the answer stems from a point with which we started—that notions from the computer revolution, such as digitality and information, have assumed such importance in our collective imaginary. As said there, many people assume there is a fundamental (discrete!) divide between people and computational "machines"—that the latter, by virtue of a presumptive neatness, formality, and cut-and-dried conceptual structure, have no purchase on the contested and metaphorical "wetness" of human existence.

I would be the last to claim that anything anyone has built so far can manifest care, chuckle ironically or make a surreptitious gesture. But it is not a fact from which I would extract metaphysical comfort. We have a long history, after all, of striving to maintain the human as fundamentally distinct from the other systems with which we share our habitat: the heavens before Galileo, the animals before Darwin. Reaching for non-discreteness as a way to secure us from the encroachment of the Information Age is just as likely, in my view, to be grasping at metaphysical straw.

Any importance (and humility) that we humans are worth must stem from concrete facts about our actual existence, not from any presumptive immunity from being reproduced—or perhaps more elementally, from belonging to the world.

8 Successor metaphysics

Enough negative claims. It is boring, ultimately, to say how the world is not. Much more important—to say nothing of more fun—to see how it actually is.

I began by saying that everyone "knows" that computers are discrete. I argued that they are wrong. But it is not digitality *per se*, that has been my primary target. Rather, this investigation grew out of what initially seemed like a much more general project: to understand *formality*, and the even more widespread consensus that computers are formal (that they themselves are formal, that they must be studied formally, etc.). It was evident to me, from the outset, that 'formal' is an amazing—and assuredly *non*-formal word—not a notion that will ever succumb to clear definition. Depending on how you count, there are anywhere from two or three to a dozen distinct meanings of the term—meanings such as "independent of semantics," "abstract," "able to be mathematically modeled," "purely ideal" (as in Platonic forms), and the like. Over many years of trying to make sense of them, it gradually emerged that what lay underneath these various readings, and tied them together into a coherent group, was their common presumption of exactly the sort of higher-order discreteness under discussion here.

Cognitive science's interpretation of 'formal' as meaning "independent of semantics," for example, turns out on sustained analysis to come to neither more nor less than the abstract claim suggested earlier: that computational systems are (allegedly) distinctive, among semantically interpreted system more generally, in that the divide between the syntactic and the semantic is sharp—engendering the claim that in the case of such systems the two realms are "*independent*." Similarly for the "abstract" reading: many things in the world, such as hospitals and birthday presents, are defined at a higher level of abstraction than the purely physical. What makes something like a number or type (but not a hospital) *formal*, in the time-honoured sense of being *abstract*, is the claim that the divide between the abstract object and any physical realisation or instantiation of it is (once again) sharp or absolute.

In fact, if I had to reduce the last century of logic, set theory, mathematics, (academic) computer science, and so forth to a single phrase, I would say the following: that

Formality is discreteness run amok.

Every one of those different readings of formality rests on an assumption about the existence of a strict, black-and-white, cut-and-dry, discrete distinction. Thus a strict *subject/object* split is presumed by the scientific method; a strict *syntax/semantics* split, alleged in cognitive science and the philosophy of mind; a strict *abstract/concrete* split, assumed in recursion theory and the theory of computability. By formal ontology, similarly, is meant ontology where the individuation criteria are discrete—the same presupposition that underlies the rather general reading of 'formal' as 'capable of being mathematically modeled.' Note that physics, the calculus, and continuous mathematics are all formal by this count, as well. Admittedly, these fields license continuous values, but the prior and constitutive higher-order questions, such as whether x is or is not equal to 0.32157, are assumed to have *precise, determinate, yes-no* answers, of exactly the sort that we have been considering here. Nor is repairing to probabilities of any help; the probability of whether a given event P will happen may be 0.62, but the boundary of that 0.62 is as sharp and discrete as any we have yet seen. So too, by the same criterion, are the assumptions of fuzzy logic—still discrete and formal. None of these "weakenings" are anything like strong enough to escape the grip of the formalist tradition.

But that observation in turns points towards the sort of picture I want to construct in its place. To see how it might work, note that if I ask you to write your name on the wall, here, next to where I am working, it does not follow that there is any ambiguity about where I am pointing, just from the fact that there is no discrete fact about my description's reference. And if there is a problem, no doubt we can talk about it, work it out. Only prejudice says that intellectual inquiry must start with the discrete, i.e., with the digital, and build everything up on top of that. Yes, that is the current practice in logic and mathematics: one sets out with discrete sets, constructs the integers out of them, defines continuity as limits of infinite series of discreteness, and models vagueness on top of that-an overall strategy captured in Kronecker's famous dictum that "God made the integers; all else is the work of man." As far as I am concerned, however, *Kronecker got it almost exactly backwards*. Discrete integers are the work of (yes) man; God made everything else.

So here is what I want. I want to start over, at a violently rupturous beginning—a feisty, obstreperous, riotous fount of an overwhelming mass of stuff. That is where we live; that is what we are made of; that is what we inhabit. And that is where I want to ground metaphysics—and to do so with no prior commitment to reductionist formality. In fact I want no prior commitment to any distinctions at all. Not to rational foundations, not to mathematics (which will anyway have to be overhauled), not to the transcendental a priori, not to the very very small. No discrete distinctions whatsoever should be presumed in advance—between might and reason; among truth, beauty, and goodness; between intentional directedness and the directedness of obligation or duty or awe. For to do any of those things would be to build in discrete formal boundaries at the outset. And that, in turn, I am convinced, experience with an in-the-wild practice shows to be a mistake. It is only through lived, complex processes of stabilisation, of domestication—perhaps of taming and of tilling—that we partially and constantly work our patterns in the flux: register objects, temporarily set up negotiable borders, live, practice, and carry on our decidedly informal and indiscrete affairs.