CHAPTER 1

ω: with a whimper: messy fields

A common "end" of a field is for it to degenerate into details; I believe that this is the usual way fields and subfields of science end. You might call it the "heat death" of a field.

In the below, I outline a theoretical framework (Algorithmic Information Theory), and illustrate with examples.

I see science as the study of concrete patterns (and mathematics as the study of abstract patterns); one often calls these patterns "theories". Within a particular field, you can catalog observations or facts, and find patterns (in the data) and meta-patterns: meta-patterns are patterns not in the data but in the theory.

My suggestion is that in a given field, the natural development is to discover the most elegant patterns (the short theorems and short proofs that say a lot), then get progressively more specialized, eventually descending into tedium.

Note that this is ahistorical: you may only discover the elegant theorems after understanding many inelegant cases, but ultimately you do exhaust the elegant patterns, and even the not-so-elegant patterns, and eventually all that's left is a pile of data.

This "end" is distinct from "computing more decimal places": I don't mean more carefully measuring known phenomena (refining the statements of existing theories), but the kinds of new phenomena that you see.

(To explain the punny title: ω for a "small end". The "whimper" references T. S. Eliot.)

1. Algorithmic Information Theory

This essay owes a profound debt to Chaitin's work; see my essay "Chaitin's nightmare" for a more formal discussion.

Algorithmic complexity formalizes the notion of "simple pattern"; it quantifies Occam's Razor, and formalizes an insight of Leibniz's.

In a nutshell, algorithmic complexity says that the complexity of a pattern is the complexity of its shortest description. Formally, the complexity of a (finite or infinite) string is the length of the shortest program that generates that string (there are technicalities that I'm ignoring, clearly). Thus the binary string 111 . . . that's always 1 has a very short description (namely "print 1 forever"), while a random string is (by definition) one that is its own shortest description (or more formally, the shortest program that generates it is "print (the string itself)").

As a concrete example, I would define the computational complexity of a game (like Tic-Tac-Toe) as the length of the shortest program that has "perfect play".

How well AIT describes science is unclear (scientific theories are not formal systems), but this is at least suggestive.

A key insight is Leibniz's, in section VI of his "Discours de mtaphysique" (Discourse on Metaphysics), quoted by Chaitin: a theory must be simpler than the data it explains. A meta-mathematical consequence, drawn by Chaitin, is that you eventually run out of elegant results: there are more short truths than short proofs.

This formalizes what one means by "all the interesting problems are solved": eventually proofs become long enough that they don't explain any more than a simple enumeration of cases.

Chaitin is most interested in consequences of unknowablity, which has deep epistemoloical consequences. I'm more interested in what this tells us about what we *can* know: my point is not "there are some things we cannot know", but rather "the statements that we make about a given system are take longer and longer to state and explain less and less".

Where does this end? For infinite fields, there is a sharp divide: there are truths we cannot prove. Further, even the patterns we can write down (in finite and infinite fields) become progressively longer, and eventually become either uninteresting in practice or uncomputable in practice (sure, you could write it down, but it would take two million years, even with a supercomputer). This yields a fuzzy practical limit to knowledge.

In practice, the line between a "very specialized pattern" and "pure data" is blurred (does your Connect Four program list 20 possible endgames, or just do brute force solution?).

2. Examples of messy fields

By "messy fields" I mean ones where we can see this limit of knowledge: where in practice there is some "raw data", patterns that cannot be described more simply.

2.1. My motivation. How did I become interested in this? I had been thinking about a few questions, then read Chaitin's "The Unknowable" (available on his website, though I found it in the library). Connecting these, I figured out the content of this essay.

Two question I had been thinking about:

Skill at games: I was thinking about "how do you become 'good' at a game?" (in the context of skill, and of fun; specifically computer games).

Here I was thinking of "know the strategy" versus "have a fast trigger finger", yielding a continuum of games, from pure thought (math games) to pure skill (reflexes). This isn't the same continuum that I'm discussing here (between elegant formal games that have a good answer and messy formal games where you use brute force), but in both cases computers are far better (at reflexes and brute force) than humans are.

Antenna design: I heard that this was a hard field, and thought I might be able to contribute. Instead, I learned something far more interesting: it was a formal field that you *couldn't* solve.

2.2. Game strategy: Chess and Connect Four. Chess strategy (as far as I understand it) illustrates this pattern of exhaustion: there are some general patterns, including a few key openings and closings, but specialized rules grow increasingly baroque and are no simpler than exhaustion.

This is clearer with "Connect Four", which is a far simpler game than chess, but complex enough to illustrate these principles. It has been solved by James D. Allen (1988) and indepently, Victor Allis (1988), and thus we can speak of it definitively. There are 2 basic strategies: a quick win, as in Tic-Tac-Toe (if you have two possible wins on your next move, the opponent cannot block them both) and a slow win (if you will win when your opponent moves in a given column, and they will eventually move there by running out of other moves, then you win: don't move in that column, and avoid obvious vertical losses). Beyond that, there are a few subtler strategies, and a few special cases (which can also be resolved by exhaustion): the point is that the patterns become more specific and less applicable, and eventually you get a complete answer, with the last details being essentially an enumeration.

You can even see this in Tic-Tac-Toe! Basic strategy is to move in the center, or if playing second, move in the corner – and then there are a few details on how to play further cases (they are simple enough that a child can enumerate them, as I in fact did at that age). Allis illustrates this in his thesis.

Returning to chess, the field *has* progressed historically: players identified further patterns (not necessarily "this is the best move in such cases", but at least heuristically "this sort of move tends to be good"), which are referred to as "strategies". The ultimate domination of computer chess proves to most players that chess is fundamentally not a strategic game, but a tactical one – beyond a certain point, brute force beats subtler pattern-finding. Humans are far better strategists than computers (being pattern-recognizers par excellence), while computers are far better, well, computers. Chess theory can still develop (new strategies can be developed), but the endgame is already known: you ultimately just search through solutions.

Now, mathematically, chess is a relatively complicated game (many kinds of pieces, some special rules), but the same sort of analysis applies to mathematically simpler games, like Connect Four (as I've shown) and checkers.

2.3. Antenna design. There is apparently no general theory of antenna design: there are tractible special cases, but in general each antenna must be designed individually: there is no algorithm or even many interesting families of "good antennas".

Even the explanation of why a particular antenna works for a given application sheds essentially no light on how to design the next one, and these days people often use evolutionary algorithms to just "find good answers".

This is most notable in NASA's automated antenna design: genetic algorithms can produce better antennas faster (and more cheaply) than can trained humans, notably for the ST5 project.

This is a priori *shocking*: design and engineering feel like profound, high-level human activies.

The reason computers outperform humans at this is obviously because it plays to computers' strengths: the problem is ultimately *computation*, not *pattern-finding*. The domain is messy: antennas behave in very complicated ways, and an understanding of the principals doesn't take you far: the best antennae are "lucky": they work because enough stars align, and can only be found by trial and error. Indeed, given a good result from a genetic algorithm, you often can't tell why it works, nor do you care.

3. Mathematical Examples

3.1. Computer-assisted proofs: the 4-color theorem, Kepler's Conjecture, etc. The 4-color theorem (every planar graph is 4-colorable) is a notorious example of a computer-assisted proof, as is Kepler's Conjecture (now Hales' theorem). There is a short proof that every planar graph is 5-colorable, but the only known proof of the 4-color theorem reduces to a long enumeration of cases. It is possible that there is no short proof: it just "happens" that none of the counterexamples work, but for no good reason.

It would be more pleasing were there "a good reason": some general principle that implied the 4-color theorem without a computation, but in general there is no reason to expect a short, elegant problem to have a short or elegant solution. (This is a key point of Chaitin's "empirical facts of mathematics".)

A very interesting critique of computer-assisted proofs (whose attribution escapes me) is that the kind of proof that a computer would like (an enumeration, which it could easily check) is a proof that a human wouldn't: this underlines that (human) math is about *understanding* and *patterns*, not simply solving problems.

3.2. Number theory. There are similar possible examples in number theory: perhaps Goldbach's conjecture (every even number can be expressed as the sum of two primes) is true for a good reason for large numbers, and just happens to be true for small numbers: there's no short proof, but you can check every case. I don't know of a particular theorem that has such a proof, but I imagine some exist. One can also imagine theorems that are true for large numbers, but fail in a few small cases, so they are not generally true; there are many theorems which are true for any prime other than 2 and 3, for instance.

3.3. Enumerations. There are a number of enumerations in math, where we can write down a list of all mathematical structures, but "this doesn't tell you much": there is not much structure in the list: it is *simply* an enumeration. Often it is the case that the number of items grows very quickly (in terms of some size): we cannot even list all examples, and we can't say much about them other than to list them.

For instance, we can enumerate p-groups of a given order, or even positive definite unimodular lattices of a given dimension.

The number of even lattices in dimension $8k$ goes: 1, 2, 24, more than $1,160,000,000, \dots$. In other words, we understand even lattices of dimension $8, 16, 24$, but in higher dimensions there is no general structure (this doesn't actually follow: it's possible that there is some structure (say, they all fall into a single parameter group), but as far as I know there isn't).

3.4. Combinatorics. A similar situation occurs in many combinatorics questions; an example I know is Golomb rulers (which are eerily similar to Yagi-Uda antennae: it's a question of how to space tick marks on a ruler). The perfect Golomb ruler of order 4 is beautiful, in that slick combinatorics way, and many optimal (or close to optimal) Golomb rulers can be produced by projective planes and affine planes, but for some orders the best rulers have only been found by computer search and don't appear to have any general structure.

3.5. Analysis. I am told that in some areas of analysis, people no longer write papers – they write monographs. Leaving aside possible hyperbole, I find this a believable statement: analysis is a mature field, well-studied for centuries, and one would expect most of the "short papers" to have already been discovered. This is not to say that analysis is a dying field, simply a mature one. Now, if each *statement* of a theorem took a book, then you'd have trouble.

3.6. Moduli spaces. For many classification problems in algebraic geometry, you may have a moduli space with good properties, but the moduli space itself cannot be written down very nicely. For instance for curves, if genus $g \geq 22$, the moduli space of curves is of general type.

Similarly, in the classification of algebraic surfaces, little is known about the moduli spaces, and I'd suspect that they are even harder to describe in general than for curves.

A deep example is the moduli spaces of Riemannian metrics on a manifold, which has a large-scale fractal nature, very related to computational complexity (as Nabutovsky and Weinberger (my advisor) proved).

3.7. Genericity. A related point is familiar from analysis and combinatorics, where a randomly chosen representative may have some property, but it is hard to write down an explicit example: a generic continuous function is nowhere differentiable, but it wasn't until Weierstrass that anyone wrote one down, or even suspected that they existing. Similarly, for expander graphs, a random regular graph has good expansion, but writing down explicit ones is rather difficult.

4. Consequences

The above discussion, being about limitations, may strike you as dreary. It's not, of course: it is an illustration of a common structure of fields of science, namely that elegant patterns take you part of the way, and beyond that you just have data, which you can explore and enumerate, but not reduce to general patterns.

The key point is: a simple formal system may have some interesting patterns, but not everything about it need have a simple description: eventually you just list facts.

As the ad for Othello goes, "a minute to learn, a lifetime to master" (simple rules, complex solution; this is a computer game par excellence: in 1980 a computer beat the human world champion).

This is familiar in much of the natural world, notably biology and linguistics (and, further afield, history): there are patterns in biology, but there's also *simply* data (all the genes in an organism, the whole tree of life, enumeration of species, etc.); likewise there are patterns in language but also simply words. In these fields general patterns get you some distance, but not very. You can say "there are some laws of history, but not enough".

This also formalizes Rutherford's saying that "In science there is only physics; all the rest is stamp collecting." (This saying also may advocate reductionism – only studying the smallest parts is useful – which is completely wrong.): fields are scientifically interesting, indeed are *science* to the extent that they have patterns, elegant results. A field that is *only* tedious, inelegant results is uninteresting *qua* field: it's stamp collecting, not science. To the extent that a field has no patterns, it is simply a mass of data: you can know more or less of it, but there is no sense of "understanding" beyond "knowledge". Such fields are better suited to computers than humans.

To close, a consequence is that there isn't usually a neat closure to fields of science: you gradually exhaust them of interest, but further enumeration of more and more specialized patterns and examples can continue forever – but is best left to a computer.