

Conditions of Continued Existence

Closed Loops, Cascading Failure, and the Limits of Control

Forrest Landry

UVSM PUBLICATIONS
Wytheville, Virginia

© 2026 Forrest Landry. Some rights reserved.

Published by UVSM Publications LLC
965 Reed Creek Mill Rd
Wytheville, VA 24382-8287
USA

uvsmpublications.com
contact@uvsmpublications.com

ISBN 979-8-90710-241-5

DOI: https://doi.org/10.67291/BAEE5779

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

You may share this work for non-commercial purposes with proper attribution, but you may not adapt, transform, or build upon it.

Full license text: https://creativecommons.org/licenses/by-nc-nd/4.0/

Author ORCID: 0009-0005-6275-0362

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), except as permitted under the license above, without prior written permission of the publisher.

Contents

Introduction **vi**

Chapter One: Closed-Loop Design and the Non-Entropic Production Cycle **1**

I. The Asymmetry in How We Talk About Closed Loops 2

The Alphabet of a Designed System 3

II. Cradle-to-Cradle as an Upstream Obligation 4

When Something Becomes Pollution 5

III. Non-Entropic Breakdown as the Governing Standard 6

Cycles Within Cycles: Time, Possibility, and the Human Step 8

IV. Toward a Criterion for Non-Toxic Technology 9

Why Technology Is Toxic 10

V. Summary 12

Chapter Two: Abstract Replicators and the Metastability Constraint **16**

I. A Third Constraint 17

II. Two Prior Constraints, Briefly Restated 18

The Two Prior Constraints Stated as Absolutes 20

III. Abstraction Capacity and the Birth of Abstract Replicators 21

IV. The Design-Constraint Reframe: Type 3 Coherence 23

V. The Closed Loop: Nature, Humans, Text, and Back Again 25

VI. Thresholds, Not a Boolean Switch 26

VII. Parts and Wholeness: Why an Epistemology of Symmetry Fails Alone 27

VIII. Metastability as the Governing Criterion 29

IX. Summary 31

Chapter Three: The Propagation of Failure **35**

I. The Nature of the Initial Perturbation 36

II. First-Order Cascade: The Medical System 37

III. The Secondary Cascade: Institutional Fragility 38

IV. The Tertiary Cascade: Economic and Financial Systems 39
V. The Vulnerability of Attention 41
VI. The Coordination Problem 42
VII. Why Modern Coupling Makes It Worse 43
VIII. The General Principle 45
Hardening the Grid: Thresholds, Phases and Enduring Law 47
IX. Summary 52

Chapter Four: Relative and Absolute Limits 56
I. Why the Distinction Matters 57
II. The Trinity of Compute 58
III. The Threshold: Three Features, Not One Machine 59
Narrow and General by Definition, Not by Limit 62
IV. Rice's Theorem and the Collapse of Safety Predicates 63
V. The Ring: Control, Agent, and World 66
VI. A Second Law for the Loop 68
VII. Why Error Correction Does Not Answer This 72
VIII. Summary 73

Chapter Five: Hash Functions, Control Theory, and Computational Phase Transitions 78
I. The Hash Function as Limiting Case 79
II. The Opposite Extreme: Perfect Linearity 80
III. The Spectrum Between: How Hashy Is It? 81
IV. The Three Regions 82
V. Recursive Systems and the Blockchain Analogy 83
VI. Symbolic Dynamics and the Topology of Attractors 84
VII. Physical Phase Transitions 85
VIII. The SAT Problem and Computational Phase Transitions 87
IX. Crossing the Noise Floor: Why Controllability Has a Phase Transition 89
X. Implications for Complex Systems Design 91
XI. Summary 92

Chapter Six: What Makes a Technology Existential 96

I. Tool or Parasite? 97
II. The Boundary That Can Be Defined 99
III. Safe, Dangerous, and Existential 101
IV. Why Exponential Processes Are Hard to See 103
V. AGI Has All Four Aspects 104
VI. Keeping AI from Becoming AGI 106
VII. Summary 108

Terms **113**

Bibliography **134**

Index **142**

Introduction

A hospital becomes the place where the infection spreads; a controller built to keep a system safe meets a question that has no answer; the shield becomes the wound.

The relationship between a civilization and its own technological capacity carries a failure mode that is a constraint to be recognized, and not a defect to be corrected. Certain features of technology, and of the systems into which it is introduced, are conditions a civilization has to meet to go on existing, and they cannot be solved from inside those systems. This constraint is structural, not incidental: no improvement within the technological domain itself can remove it. Each of the six chapters of this volume addresses its own question in its own terms, and each arrives at this structural diagnosis, which recurs across all six, by its own route and in its own register.

The reading order of this volume follows the argument from the concrete to the formal. Each chapter opens with an abstract and closes with a summary and a list of its key concepts, so that a reader can come to any chapter alone; a reader going straight through can pass over them.

The first chapter begins with the simplest case. A closed-loop production cycle requires that the production process itself, at the moment of manufacture, be designed from the outset to anticipate the breakdown and reuse of what it produces. Good recycling processes at the downstream end do not achieve it alone. The standard is that breakdown should be as *non-entropic* as possible: preserving as much of the material's usable order as the process can manage. This is a small and precise claim, but it carries a larger one inside it. The obligation of the maker extends across the entire lifetime of the atoms it organizes, and responsibility for what happens downstream begins upstream, at the point where design still has the power to shape what gets built.

The second chapter broadens that principle from material production to the domain of ideas and tools. Once a species develops the capacity for abstraction, its ideas and techniques become what the chapter calls *abstract replicators*: patterns that can be shared, recombined, and propagated independently of their origin. The chapter shows that these abstract replicators must remain in what it calls a *Type 3 coherent* relationship with the *instantiated* structures, the concrete natural and human systems, that gave rise to the capacity to produce them. A Type 3 coherent relationship is one in which the abstract actively supports, and does not merely tolerate, the concrete ground it came from. The governing criterion is *metastability*: a dynamic, actively maintained equilibrium between the abstract and the concrete, in which each supports rather than debases the other. Without this closure, the loop from nature to humanity to technology actively undermines its own future possibility.

The third chapter turns from principle to evidence. Using the early weeks of the 2020 pandemic as a detailed case study, it traces the mechanism by which a disruption in a single domain propagates outward through every system coupled to it, at the speed of the coupling rather than at the speed of the original threat. The signature pattern it identifies is the *inversion of protective function*: the institution designed to contain a harm becomes, under overload, a source of that harm. The hospital becomes the infection site. The enforcement agency becomes the vector. The coordination body becomes the bottleneck. The general principle drawn from this concerns the relationship between efficiency and catastrophic fragility: the tight coupling which produces performance under normal conditions is the same architectural feature that produces cascading collapse when abnormal conditions arrive. The chapter then applies its principle to one domain on which all the others depend, the electrical grid, and sets out what a law would have to do to keep the grid's protection in place.

The fourth chapter addresses the formal question: can a sufficiently general artificial intelligence be controlled? It distinguishes two forms of argument, the *relative* and the *absolute*, and develops both. The relative

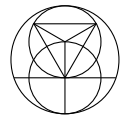
form measures the cost of control against the *trinity of compute* (time, space, and energy, the three costs of any computation) and shows that the cost rises faster than any capacity to pay it. The absolute form identifies the structural threshold of *computational universality*, the capacity to run any program. Above that threshold every property one cares about, including safety, reduces to the *halting problem*: the question whether a given program, run on a given input, will ever stop. Alan Turing proved in 1936 that no program can answer that question correctly for every program and input. The two forms interlock: when someone shows that a fixed case is controllable, the relative argument shows what happens as it scales; when someone argues that scaling costs might be met, the absolute argument shows that the destination is not expensive but empty. The chapter also introduces a ring of three parts (controller, agent, and world) and shows that, because each part amplifies the noise it receives above a critical level, uncertainty increases at every traversal of the loop.

The fifth chapter explains the measure of noise amplification that the fourth chapter calls *hashiness*. Every mapping from inputs to outputs lies somewhere on a continuous spectrum. At one end is the perfectly linear system, which control theory predicts and corrects completely. At the other end is the cryptographic hash function, which admits no control theory at all. The spectrum divides into three regions: one where control theory works, one where it cannot work, and an undecidable middle region where no general procedure can say whether it will work. The boundary between these regions is a hard boundary, a *phase transition*, like the boiling of water. Governance systems must therefore be designed to remain sound when prediction is absent, rather than relying on better models.

The sixth chapter draws the boundary that the title of this volume implies: the line between technology that meets the conditions of continued existence and technology that cannot. The boundary which can be clearly defined is the one between technology that is existentially terminal and technology with which the ecosystem, and therefore we ourselves, continue. A technology is existential when it has an exponential aspect that is unconstrained and unconstrainable, and that overwrites the patterns of

biological life; such technology must be avoided at all costs, regardless of the care applied. Artificial general intelligence has all four of these aspects by its nature. The four aspects are a test that suffices, and the case about artificial general intelligence does not rest on all four together: the chapter sets beside the test the two closures, by control and by disposition, and, if artificial general intelligence is built, each closure by itself settles the outcome: that the world is carried out of the conditions human life needs, and that nothing can hold it in. The chapter closes with five restrictions that keep useful artificial intelligence from becoming artificial general intelligence.

The common thread across all six chapters is the claim that certain structural features of technology and of the systems into which technology is introduced are constraints to be recognized as conditions of continued existence, and not problems to be solved within those systems. The closed loop that runs from nature to humans to technology and back again is either maintained deliberately or it fails, and the failure, once it cascades through tightly coupled systems past the threshold of computational universality, is not recoverable. A controller may hold such a system within a safe range for a time, as the institutions of 2020 held theirs; the holding cannot be indefinite, and the divergence, once begun, is not recoverable. The fourth chapter shows why the cascade and the threshold meet: the coupled institutions of the third chapter have the three features that place a system above the threshold.



Chapter One

Closed-Loop Design and the Non-Entropic Production Cycle

Why Recycling Downstream Is Not the Same as Designing for a Closed Loop

ABSTRACT

This chapter develops a distinction that bears on the criteria for judging a technology non-toxic. A technology is toxic, in the sense used throughout this book, to the degree that it degrades the ground it depends on. Most discussion of closed-loop cycles in technology focuses on how to break materials back down, recycle them, and reuse them, and gives comparatively little attention to the production process that puts those materials into circulation in the first place. In this chapter I show that a closed loop requires a *cradle-to-cradle* discipline applied in production itself, at the time that a thing is made: design in which the end of one use of a material is planned, from the start, as the beginning of its next. Good downstream processes for breakdown and recycling do not achieve a closed loop by themselves. The discipline makes the breakdown which eventually occurs as non-entropic as possible, which is to say, it loses as little of the material's structure, order, and usability as it can. Production, on this view, is not finished once an object leaves the factory; it remains answerable to the entire lifetime use of the atoms it has organized, including their eventual return to a state from which they can be reorganized again. In this chapter I frame the obligation of production through the alphabet that a designed system is built from. Every material that nature does not

> already know how to take apart is a new letter, and every new letter needs a matched process for taking that letter apart, a de-synthesis, built by hand. I measure the entropy of a breakdown by the energy needed to restore a material to its first use, set beside the energy needed to make that material new. I show that a loop closes only as often as every one of the steps of that loop actually happens, so that a material cycle which waits on a human choice becomes a social problem as well as a technical one. And I show why technology, designed from the top down, misses the small accumulations and depletions that nature handles routinely.

I. The Asymmetry in How We Talk About Closed Loops

Discussion of the environmental impact of technology tends to concentrate on one half of a cycle: synthesizing, building, manufacturing, producing. Comparatively little attention goes to the other half: how what has been built gets broken back down. When people speak of a closed loop, they generally mean the whole process, not only the manufacturing half of it. What a closed loop involves goes deeper than building things that can, in principle, be broken down and recycled; it goes deeper still than having a process in place to carry out that breakdown and recycling once the time comes. It involves designing production itself, from the outset, for the breakdown that will follow.

The asymmetry has a plain cause. Synthesis is where value is made and sold: the product is what the customer pays for, and the factory is where the product comes into being. Breakdown happens later, somewhere else, usually at someone else's cost, and nothing in the sale of the product pays for it. Designers work on what someone pays for, and so the half of the cycle that is paid for is the half that is studied, improved and designed. The other half is left to whoever receives what remains.

The Alphabet of a Designed System

Every system, living or built, is made from an *alphabet*: a set of basic components, and the rules by which they combine. Nature's alphabet, the chemistry that living things are built from, has a property that no designed alphabet has by default: what is built from it is, in time, taken back apart. A leaf is built, and a leaf is unbuilt, and the same molecules serve both directions equally well. Decomposition belongs to nature's alphabet as fully as synthesis does.

A designed system often reaches beyond that alphabet. It adds new letters: materials and forms that nature has not provided and does not know how to take apart. There are good reasons to do this, since nature's alphabet does not by itself support everything people reasonably want to build. Every new letter, however, creates an obligation that nature does not discharge: a matched process for taking the new material back apart, a *de-synthesis* (the deliberate, engineered separation of a built thing into parts that can be combined again), which has to be built by hand, because nature will not supply it.

Nearly all design attention goes to synthesis. We are very good at gluing two things together and making a third thing out of them and the glue; we are poor at separating that third thing back into its first two once it has served its purpose, and often we do not try at all. A designed system is toxic, in this structural sense, to the degree that it has extended the alphabet without the matched de-synthesis. Such a system works like a one-way street: matter flows in through synthesis and never flows back out through an equally deliberate undoing. Sense-Making and the Craft of Inquiry, Chapter 19, "Social Physics", develops this account of the alphabet and its inverse.

> **A test for any new letter**
>
> For any material or form a design introduces, ask whether a matching de-synthesis has been designed alongside it, with the same seriousness as the synthesis, or whether the de-synthesis has been assumed, deferred, or ignored. Where it has been assumed, deferred, or ignored, there the toxicity of the system will collect.

II. Cradle-to-Cradle as an Upstream Obligation

A closed loop involves the cradle-to-cradle design of the production processes themselves, so that the eventual breakdown is not highly entropic. The term *cradle to cradle* comes from the architect William McDonough and the chemist Michael Braungart, who use it for products designed so that their materials return, after use, as material for new products. This reframes where the responsibility for a sustainable technology actually sits. A good recycling process that operates on whatever happens to have already been produced is not enough. The production process itself has to be considerate, from the outset, of the lifetime use of the atoms it is organizing, anticipating the breakdown, reuse, and recycling that will eventually need to happen, and shaping the initial act of production accordingly.

McDonough and Braungart make the upstream obligation concrete by sorting every material into one of two cycles. A *biological nutrient* is a material designed to return safely to soil and water, where living things take it apart. A *technical nutrient* is a material designed to circulate in industrial cycles at full quality, recovered and made again into the same kind of product. A product that binds the two kinds together so that neither can be recovered they call, borrowing a phrase from Jane Jacobs, a *monstrous hybrid*; their own example is the ordinary shoe, in which materials that could return to the soil are joined to materials that cannot, so that neither can be salvaged. The sorting can be done well only at the drawing board. Once the materials of a product are bonded, no downstream process can

separate them as cleanly as the designer could have kept them apart. The designer who adds a letter to the alphabet therefore owes its de-synthesis, and the only time that debt can be paid at low cost is before the product exists.

When Something Becomes Pollution

The upstream obligation answers a plain question: when does something become pollution, and who is responsible? *Pollution* is a sub-case of toxicity: it is something nature does not know how to deal with, like plastic in the ocean.

A person buys food packaged in a non-recyclable plastic. The person cannot recycle the package, so it goes into the trash, and we blame that person for degrading the environment. The fault is not that person's. Then we blame the restaurant for choosing the *non-biodegradable* container (one that living things cannot break down). But the restaurant chose the container because its *economic framework* (the costs and prices within which a business must work in order to stay open) says its *margins* (the difference between what a business takes in for a sale and what the sale costs it) will not work otherwise.

Something becomes pollution at the moment it is created. It does not become pollution when it reaches the landfill, when it is thrown away, when it is bought, or even at the factory where the product is assembled. It becomes pollution at the instant the raw materials are synthesized using chemistry for which nature has no reverse chemistry: the instant the plastic is made. The instant two chemicals are combined in a way nature cannot take apart, the result has exited the *cycle of nature* (the ongoing process by which living things and natural processes take materials apart and use them again), and that is when it became pollution. A wood manufacturing company produces no pollution as long as what it makes is made of materials nature already contains and already knows how to take apart.

The moment of toxicity is the moment design uses an irreversible process, and because of how we structure our economy, we do engineering only with one-way processes. Doing all our engineering with one-way processes is what "technology is toxic" means.

I say the next part plainly. So we blame the consumer for what is actually the fault of the industry. The industry is in a *feedback cycle* (a loop in which the result of an action strengthens the reasons to repeat it), because polluting is cheaper for it. The industry lobbies the government against regulations such as a bottle tax (a tax charged on bottles). In this way it manipulates the government to make polluting easier, while deceiving the consumer that the consumer is at fault. Responsibility for pollution therefore lies with the industry that makes such things, and not with the consumer who throws them away or the restaurant that serves food in them.

In the language of the alphabet, the instant of pollution is the instant a new letter is made without its inverse. The wood company uses only letters nature already has, together with their inverses, which nature runs at no cost to anyone. The maker of the plastic uses a letter whose inverse no one has built. The responsibility sits with the industry because whoever adds a letter is the only one who could still have designed in the inverse of that letter. A new letter made together with a working inverse, built and actually used, has not left the cycle: the cycle has been extended to include it.

III. Non-Entropic Breakdown as the Governing Standard

The standard here is precise: breakdown, reuse, and recycling should be as non-entropic as possible. *Entropy*, in this context, stands for the loss of usable order: the degradation of a material from a well-organized, high-value state into a disordered, low-value one that requires substantial new energy and effort to reorganize again. A production process that is inconsiderate of this eventual breakdown tends to produce materials that degrade badly when their first use ends: mixed, contaminated, or bonded in

ways that make later separation and reuse costly or impossible. A production process that is considerate of it from the start produces materials whose eventual breakdown preserves as much of their original order and usability as it can, so that the closing of the loop costs as little as possible in energy, material, and lost structure.

A pair of everyday objects shows the difference. An aluminum can is made of one metal, and remelting it into another can takes only about five percent of the energy needed to make new aluminum from ore. A snack pouch laminated from layers of plastic and aluminum foil cannot be separated at all, and its breakdown is close to total. It binds its materials together, and multilayered packages of this kind are, for the most part, not readily recyclable; the order its materials held is lost when its first use ends. The difference between the two was fixed at the moment each was designed. The comparison also shows how the entropy of a breakdown can be measured: by the energy needed to restore a material to its first use, set beside the energy needed to make it new.

A second pair shows the same measure applied within a single material. A refillable glass bottle, made heavy enough to be washed and filled again, can be cleaned and refilled up to fifty times before it is melted down; restoring it to its first use costs the energy of a wash. A glass bottle made for one use is crushed and remelted when that use ends, and each tenth of a furnace's charge that is recycled glass lowers the energy of the melt by only about three percent, so that even recycled glass needs most of the energy of new glass before it is a bottle again. The glass in the two bottles is the same, and glass can be remelted without loss of quality; the breakdown each bottle meets was fixed when it was designed.

The measure has a reason. Entropy, in the sense used here, is order lost, and lost order is measured by the work needed to put it back. A material that returns to its first use for the energy of a wash has lost almost nothing. A material that needs most of the energy of new manufacture before it can serve again has lost almost everything its first making gave it. And a material that cannot be restored at any cost has lost its order completely, and has left the cycle. The ratio of the two energies places any

breakdown on a single scale, from no loss to total loss, and it places it there at the moment of design, when the choice between the can and the pouch, or between the heavy bottle and the light one, is still open.

Cycles Within Cycles: Time, Possibility, and the Human Step

Most accounts of material cycles consider space: where a material is, what form it is in, and what facility could process it. A fuller account considers time and possibility as well. Cycles sit within cycles: the refillable bottle's cycle of filling and washing sits within the cycle of its eventual remelting, which sits within the cycle of the sand, soda and lime that glass is made from. And each cycle is a matter of likelihood as well as of possibility. The question is not only whether a material could, in principle, return to use, but how likely it is to actually return, given everything its return depends on.

A loop closes only when every step in it happens. Suppose a cycle has four steps (collection, sorting, cleaning, and remaking), and suppose each step happens nine times in ten. The whole loop then closes about two times in three, since nine tenths multiplied by itself four times is about 0.66, and one material in three leaves the cycle on each trip around it. A material that goes around such a loop five times has about one chance in eight of still being in use at the end. Each step can look nearly reliable while the loop as a whole leaks steadily. The closure of a loop is the product of the reliabilities of its steps, and the design has to make every step reliable, because a shortfall at any one step multiplies into the whole.

Many real cycles of breakdown depend, at some step, on a person: someone has to choose to return the bottle, sort the waste, or carry out the disassembly. Once a step waits on a human choice, the reliability of the whole material cycle becomes a question about human follow-through, and not only a question of chemistry or engineering. A deposit on a bottle, refunded when the bottle comes back, is an instance of designing for that step: it pays for the human part of the cycle, so that the return happens, and the refillable bottle's many fillings happen in fact and not only in

principle. Whether companies take responsibility for the waste their products become, with anything like the seriousness they bring to the products they sell, is for the most part a question nobody asks. The weak link in a material cycle is very often social. The second of the two constraints that Chapter Two restates, that people are predators, names that weak link. By my estimate from design experience, the social half of the problem is roughly three orders of magnitude harder than the material half, for the reasons set out in the essay in which this chapter's account of the alphabet is developed (Sense-Making and the Craft of Inquiry, Chapter 19, Sections V to VII).

IV. Toward a Criterion for Non-Toxic Technology

This distinction matters beyond its own narrow domain because it bears directly on what criteria would let one call a technology, or a use of technology, non-toxic. Chapter Two states the test for toxicity in its general form, for ideas and tools: an abstract replicator "can be judged toxic precisely to the degree that it fails this test", the test being whether it supports or debases the stability of the ground that produced it. For physical things the test takes this form: a technology is toxic to the degree that it degrades the ground it depends on. An account of non-toxicity that only asks whether a technology can, eventually, be broken down and recycled at all is a weaker and more permissive standard than one that asks whether the technology was designed, from the beginning, to make that eventual breakdown as non-entropic as possible. The first standard can be satisfied by almost anything, given enough energy and effort downstream. The second standard constrains the design of the thing itself, at the point where the constraint can actually do useful work.

The stronger standard also shows where technology becomes reasonable. The more completely the de-synthesis half of a cycle is built, the more the technology as a whole, synthesis and de-synthesis together, comes to behave like a living process, in which nothing is built without a

path back to its parts. The place where top-down engineering and bottom-up nature meet is the place where synthesis and de-synthesis have been brought back into one matched, self-closing cycle, whatever the origin of the materials involved (Sense-Making and the Craft of Inquiry, Chapter 19, Section IV).

The standard also has a limit case. Some technologies need, for their own making and running, conditions that living things cannot survive. Semiconductor fabrication needs temperatures above 1,400°C, atmospheres without water or oxygen, rooms free of particles, and hundreds of industrial chemicals, a significant fraction of them toxic to organic life; the mining, chemistry, power and cooling that support it displace living habitat wherever they reach (Why AGI Cannot Be Made Safe, Section 1.2.3). For such a technology the upstream criterion is failed before any question of breakdown arises: the ground is degraded by the making and the running themselves, and no design of the downstream half can repair that. Chapter Six takes up the case in which such a technology also grows by itself.

Why Technology Is Toxic

Toxicity is too much of something or too little of something needed: too little oxygen and you die; too much carbon dioxide is an overload. In ecosystems, anywhere stuff piles up in one place and necessary stuff runs short in another, the ecosystem deregulates. Technology is *top-down design*: engineering starts from the large and works down to the small, breaking big problems into smaller ones and solving them piecemeal.

A complex system has flows within flows at a fractal distribution, and any top-down design handles only first-order effects, or second- or third-order effects if the engineer is very good. Nature routinely handles fourth- and fifth-order effects. Biosphere 2 (a sealed structure in Arizona built to hold a complete, self-contained ecosystem) showed the limits of top-down design: once sealed, Biosphere 2 met problems no one had thought of. For example, trees need a little wind on them to grow stiff, and without it they

grew spindly and brittle and fell apart. The air itself showed the same thing. In the first sixteen months of closure, the oxygen inside Biosphere 2 fell from about 21 percent to about 14 percent, low enough to harm the people living inside, and oxygen had to be pumped in. The soil had been made rich in organic matter so that the crops would grow, and microbes in it consumed oxygen as they broke that matter down. The carbon dioxide the microbes gave off did not build up in the air as expected: it was reacting with the exposed concrete of the structure to form calcium carbonate, so that the walls were taking part in the carbon cycle (Severinghaus and colleagues, 1994). The concrete had been chosen for its strength, and nobody had designed it as part of the atmosphere; in a sealed world, it was part of the atmosphere. That devil-in-the-details problem is endemic to technology, because technology starts at the macroscopic and works toward the microscopic. Evolution works the other way: it tries everything, and anything that does not work at the level of detail is rejected quickly. Toxicity shows up because macroscopic design does not handle microscopic accumulation and depletion.

The difference between the two directions is a difference in feedback. Evolution has a loop that corrects for harm at the level of detail: whatever fails there is removed, generation after generation, and what persists is what the ecosystem can absorb. The process by which technology is made has no such loop. A design is finished when it ships, and its fourth- and fifth-order effects arrive, if at all, long after the designers have moved on, at a place and in a form that no step of the design process ever sees. Evolution, Design, and the Ethics of the Possible, Chapter 1, Section III, gives this ground for the premise that technology is toxic.

V. Summary

In this chapter I distinguish a downstream capacity to break something down from an upstream discipline that shapes production so that breakdown is efficient. Recycling technology addresses only part of what a closed loop requires. From this distinction comes the shape of a better standard for non-toxic technology.

It frames the upstream obligation through the alphabet of a designed system: each new letter that nature cannot take apart needs a matched desynthesis, built by hand, and a system is toxic to the degree that it adds letters without them. It measures the entropy of a breakdown by the energy of restoring a material to its first use, set beside the energy of making it new. It shows that a loop closes only as often as all of its steps happen, so that a cycle which waits on a human choice becomes a social problem as well as a material one. It names a limit case, the technology whose making and running need conditions that living things cannot survive. And it shows that technology is toxic because its process of making has no loop that corrects for harm at the level of detail, while evolution's process has one.

The claim developed here is compact but consequential. A closed-loop cycle is achieved by extending the discipline of the loop back into the moment of production itself, so that materials are organized, from the start, in a way that anticipates their own eventual breakdown and keeps that breakdown as non-entropic as possible. Good recycling infrastructure for whatever has already been produced does not achieve it alone. This reframes what it would mean for a technology to be non-toxic: the question moves from the point of disposal to the point of manufacture, where the answer actually has the power to change what gets built in the first place.

The Introduction called this a small and precise claim that carries a larger one inside it; the chapter has now shown the larger one. The obligation of the maker extends across the entire lifetime of the atoms it organ-

izes, and responsibility for what happens downstream begins upstream, at the point where design still has the power to shape what gets built. Recycling downstream is an improvement made inside the domain of technology; it cannot close the loop, because the loop is decided upstream.

KEY CONCEPTS IN THIS CHAPTER

- **Closed loop as whole-cycle obligation:** a closed loop requires attention to production as much as to breakdown and recycling; recycling infrastructure is not sufficient on its own.
- **Cradle-to-cradle design:** the discipline of shaping a production process from the outset with the eventual breakdown, reuse, and recycling of its materials already in view.
- **Non-entropic breakdown:** the standard that breakdown, reuse, and recycling should preserve as much of a material's original order and usability as possible, minimizing the loss of usable structure.
- **Lifetime use of atoms:** a producer's responsibility extends across the entire life of the material organized, not merely to the point where a finished object leaves the factory.
- **Upstream criterion for non-toxicity:** the standard that judging a technology non-toxic depends on whether its production was designed for low-entropy breakdown, and not merely on whether breakdown is technically possible at all.
- **Pollution begins at creation:** a thing becomes pollution at the instant its raw materials are synthesized with chemistry for which nature has no reverse chemistry, when it exits the cycle of nature; responsibility therefore lies with the industry that makes it, and not with the consumer or the restaurant.
- **Industry feedback cycle:** polluting is cheaper, so the industry lobbies against regulations such as a bottle tax, while the consumer is told the fault is the consumer's.
- **Alphabet of a designed system:** the set of basic components and rules of combination a system is built from; nature's alphabet has decomposition matched to its synthesis, and every new letter a design adds needs a matched de-synthesis built by hand.

- **De-synthesis:** the deliberate, engineered separation of a built thing into parts that can be combined again; a system is toxic to the degree that it extends the alphabet without it.
- **Biological and technical nutrients:** McDonough and Braungart's two cycles, the one returning safely to soil and water, the other circulating in industry at full quality; a monstrous hybrid binds the two so that neither can be recovered.
- **Measure of a breakdown:** the energy needed to restore a material to its first use, set beside the energy needed to make it new; lost order is measured by the work of putting it back.
- **Closure as a product:** a loop closes only as often as every one of its steps happens, so the reliability of the whole is the product of the reliabilities of the steps.
- **The human step:** where a cycle waits on a human choice, its reliability becomes a question of human follow-through; the weak link in a material cycle is very often social.
- **The limit case:** a technology whose making and running need conditions that living things cannot survive fails the upstream criterion before any question of breakdown arises.
- **Feedback in evolution and its absence in technology:** evolution removes, at the level of detail, whatever fails; the process of design has no such loop, which is the ground of the premise that technology is toxic.
- **Toxicity as accumulation and depletion:** too much of something or too little of something needed; top-down design handles first-, second- or third-order effects, nature handles fourth- and fifth-order effects, and toxicity shows up because macroscopic design does not handle microscopic accumulation and depletion.



Chapter Two

Abstract Replicators and the Metastability Constraint

Why Ideas and Tools Must Remain in Service to the Ground That Made Them Possible

The principle developed in the preceding chapter, that production must account for the full lifetime of what it produces, applies most directly to physical materials. This chapter extends it to a domain where the stakes are higher and the closure of the loop more difficult: the domain of ideas and tools themselves, and the question of what happens when abstraction, the very capacity that makes technology possible, begins to outpace the wisdom required to use it safely.

ABSTRACT

This chapter sets out a third constraint, beside two others (that technology is toxic, and that people are predators), as part of a small set of structural limits on what any sufficiently advanced civilization can survive. The third constraint concerns an asymmetry: the level of insight required to produce a powerful new technology is far smaller than the level of knowledge required to use that technology safely, so that it becomes progressively easier to break fundamental things than to build or repair them. It traces this constraint to a more basic phenomenon, abstraction capacity, which gives rise to abstract replicators: ideas and tools that can be shared and recombined. It shows that these abstract replicators must remain Type 3 coherent with the instantiated replicators, the concrete natural and human structures that gave rise to the capacity to produce them

> in the first place. Of the three types of coherence (resonance; network coherence; and the interaction of multiple levels of abstract causation without overall disorder), only the third reaches across the levels that join the abstract to the instantiated, and only the third has effects that are exponential, and so keep pace with an abstraction capacity that itself grows exponentially. It traces this through a closed loop running from nature to humans to technology and back, through the historical thresholds at which abstraction capacity itself increased, and through the distinction between an epistemology of symmetry and an epistemology of continuity (an epistemology is a way of knowing). It closes by setting out metastability as the governing criterion for whether a given abstract replicator is safe to build at all.

I. A Third Constraint

Two constraints are already in place, set out as design requirements in *The Displacement of Choice*: that technology is toxic, and that people are predators. In this chapter I take up a third constraint, the *break-versus-build asymmetry*. The break-versus-build asymmetry is named for the observation that, as insight becomes powerful enough, breaking fundamental things becomes easier than building or repairing fundamental things. The break-versus-build asymmetry is an asymmetry of capability between creation and stewardship.

The asymmetry has a structural reason. A built thing works only if every one of its parts works and every one of its relations holds, and the built thing fails if any one of those parts or relations fails. Building is a conjunction of many conditions, all of which must be met; breaking is a disjunction of them, any one of which suffices. The insight needed to break a thing therefore has to find one condition, while the knowledge needed to build or repair it has to cover them all. Making a new technology is also building, and all of its conditions must hold too; but they are the few conditions that hold the product itself together. Using it safely is a

conjunction over a far larger set, every condition of the whole into which the product goes, and releasing the product can break any one of them. The insight needed to make a powerful thing is therefore so much smaller than the knowledge needed to use that thing safely. As insight grows more powerful, it reaches more of the conditions that hold fundamental things together, and so it reaches more ways of breaking them, faster than any growth in knowledge can cover all the ways those things must be kept whole.

History gives a clear case. In 1928 a team led by the engineer Thomas Midgley developed *chlorofluorocarbons*, compounds of chlorine, fluorine and carbon, as safe refrigerants: they did not burn, were not poisonous, and did not react with what they touched. The insight needed to make them was the insight of one industrial laboratory. The knowledge needed to use them safely took nearly half a century and a worldwide science of the atmosphere. In 1974 the chemists Mario Molina and F. Sherwood Rowland showed that these compounds, because they are so stable, survive long enough to drift up into the stratosphere, where sunlight breaks them apart and the chlorine they release destroys *ozone*, the gas that screens the surface of the Earth from the sun's ultraviolet light. Making the compound took one laboratory; learning what it did took decades; stopping its production took an international treaty, the Montreal Protocol of 1987. The chlorofluorocarbons show the break-versus-build asymmetry in a single history.

II. Two Prior Constraints, Briefly Restated

To locate the third constraint properly, the first two deserve brief restatement. "People are predators" is treated here as mapping onto *rivalrous* risk dynamics, in which each actor gains only by others' losing. Such dynamics become *self-terminating*, destroying the conditions that let them continue, once technology reaches exponential power (power that grows by doubling, and doubling again). They are competitive dynamics among

agents that, left unconstrained, generate risk faster than any single actor can manage, and eventually undermine the conditions of the competition itself. "Technology is toxic" is treated as mapping onto a pattern in which the merely complicated subsumes and eventually collapses what is complex. A system that is *complicated* has many parts with deterministic interactions, and can in principle be modeled by a sufficiently large computation. A system that is *complex* has many parts with nonlinear feedback, emergent behavior, and sensitivity to initial conditions, and cannot be captured by any model simpler than the system itself. A jet engine is complicated: it has very many parts, and every one of them was designed to do what it does. A forest is complex: no list of its parts, however long, tells what it will do next year. Chapter One of this book takes up the criteria for non-toxic technology that this pattern bears on, from the side of where in the cycle of production the criterion must apply; "Social Physics", Chapter 19 of Sense-Making and the Craft of Inquiry, develops them further, through the alphabet a designed system is built from.

Each of the two prior constraints has its own ground. Technology is toxic because the process that makes it has none of the feedback loops that correct for harm: evolution removes, at the level of detail, whatever does not work, and the process of design does not (Chapter One, Why Technology Is Toxic; Evolution, Design, and the Ethics of the Possible, Chapter 1, Section III). People are predators because the capacities that make technology possible (causal reasoning, planning, and the prediction and outwitting of other living beings) are the capacities of predators, so that technology arises only in predator species, and such species go on being predators in how they use the technology they develop (Evolution, Design, and the Ethics of the Possible, Chapter 1, Section II, and Chapter 11, Section II). The five statements of the first constraint, in Chapter One and in this section, are one claim seen from different sides. Toxicity is what happens: the ground a technology depends on is degraded, by too much of something or too little of something needed. One-way processes and top-down design are how it happens. That the process of making technology has no

loop which corrects for harm at the level of detail is why it happens. And the merely complicated subsuming and eventually collapsing what is complex is what toxicity looks like at the scale of a whole system.

The Two Prior Constraints Stated as Absolutes

In characterizing the problem of a viable civilization, I state the two constraints of Section II as two poles, and the solution has to pass between them. I state them in an absolutist way: technology is toxic, and people are predators. I state them this way because, by making the situation a little more dramatic and simplified, we handle it comprehensively, and not because reality is exactly that. Anything that passes as a solution under the harder statement also handles the more specific circumstances we actually face. Setting the problem up as a harder problem sometimes makes the hard problem that actually has to be faced easy to solve. If the actual reality is ninety percent toxic and ninety percent predatory, and I did not build the solution to cope with one hundred percent, I have little confidence that it will cope with the situation being mostly so.

A design requirement stated this way works like a load rating. A telephone is rated for a range of temperature and humidity, and if it fails inside that range it is not ready for sale. A design for governance is rated the same way, and its range includes participants who will act in their own interest at the expense of the whole whenever the design allows it. The two absolutes set the range: a proposal that cannot function while both hold is not ready.

Both conditions are at least eighty-five percent true in a way that is uncontestable; for anyone who knows what evidence to look for, the evidence is overwhelming. Both conditions are also very hard to accept: with a very capable colleague it took about fifteen conversations before he accepted that technology is toxic, and at least five more before he accepted that people are predators. And both conditions are stringent. Demanding that a solution to world governance work with both conditions absolutely true shrinks the set of solutions almost to empty, because the two conditions

cover almost all the tools we know how to use, and also the tool user. If civilization is the set of tools we use to solve problems, we have condemned both the nature of tool working and the nature of the tool user.

The evidence that people are predators is in the body. Look at animals and the *morphology* of the face (the form and structure of the face). Predators have different equipment than prey: herbivores are more likely to have hooves than claws, flat teeth than pointed teeth, and eyes on the sides of the head than facing forward. The ape species are omnivores, but they have canines, forward-facing eyes, and digits that are more claw-like than hoof-like. Predators and prey differ in their digestive systems and their social structures too: the way herd animals negotiate their social scene and the way predators negotiate theirs are structurally different in obvious ways. Anyone who does the exercise of placing a new species into the categories of niche finds the answer point-blank. Placed by the same exercise, humans fall among the predators, and that placement is the people-are-predators condition. I only need it at the biological level, though it matters at the level of the psyche, since the psyche is about eighty-five percent defined by biology.

III. Abstraction Capacity and the Birth of Abstract Replicators

The third constraint, on this account, shares a common root with the constraint that technology is toxic. Both technology's toxicity and the break-versus-build asymmetry trace back to the same underlying capacity: *abstraction*, the capacity to extract a pattern from a particular instance and represent it independently of that instance, whether the pattern in question is a physical technique or a symbolic, linguistic structure.

Once abstraction is possible, a further consequence follows almost immediately: abstract patterns can become *abstract replicators*, ideas that can be shared between minds, and tools that can be built from shared designs, propagating and recombining in ways that a merely particular, non-ab-

stracted skill or object cannot. The power, and the danger, of abstraction capacity lies precisely in this replicating property: an abstract replicator, once released, is no longer bound to the single context of its origin.

Abstraction separates a pattern from its instance. A person who knaps a flint into an edge has made one edge; a person who understands what makes an edge sharp can make edges from any suitable stone, and can teach that understanding to someone who has never seen the first edge. The edge stays where it was made; the understanding travels.

That traveling is the replicating property. An abstract replicator has what any replicator has: it is copied, the copies vary, and some copies spread more widely than others. Richard Dawkins named ideas that copy themselves from mind to mind *memes*, in The Selfish Gene (1976). The abstract replicator includes those, and it also includes tools built from shared designs, techniques, methods and procedures: anything whose pattern can be carried away from the instance it was drawn from and made real again elsewhere.

Abstract replicators differ in one decisive way from the replicators of biology. *Evolved replicators* are held in check by co-evolution: each organism's reshaping of its environment is shaped in turn by the reshaping done by every other organism around it, and the result is a dynamic equilibrium, often violent, but bounded. *Designed replicators* are not held in check by co-evolution. Designed replicators enter an existing ecosystem from outside the evolutionary process, and they reshape that ecosystem according to requirements that were never negotiated against the requirements of the life already there (*The Substrate and the Ecosystem*, Chapter 1, Section IV). An abstract replicator is a designed replicator. Whatever coherence it has with the ground it came from has to be designed into it, because no co-evolution will supply it. So in the next section I treat the coherence of an abstract replicator with its ground as a design constraint.

IV. The Design-Constraint Reframe: Type 3 Coherence

In this chapter I treat the break-versus-build asymmetry as a design constraint, to be built into the evaluation of any candidate abstract replicator from the outset, and not as a problem to be managed after the fact. The constraint is this: abstract replicators have to be Type 3 coherent with the *instantiated replicators*, the concrete, embodied structures, biological and social, that gave rise to the capacity to produce abstraction in the first place. (To be instantiated is to exist as a particular, embodied thing.)

Type 3 coherence, in this usage, names a relationship stronger than mere compatibility: it requires that the abstract replicator actively support, rather than merely tolerate, the continued integrity of the instantiated structure it emerged from. An abstract replicator that is Type 3 coherent with its instantiated ground does not merely avoid immediate harm to that ground; it participates in sustaining the very conditions that made its own emergence possible.

The three types of coherence

The numeral places this relationship among three types of coherence, which are described here together.

Type 1 coherence is the coherence of resonance, akin to that found in a laser.

Type 2 coherence is network coherence, such as a logical net, or a belief system that is coherent as a set of ideas working together.

Type 3 coherence is the coherence in which multiple levels of abstract causation (causes that act through patterns such as rules, meanings and programs, and that act on the things that carry those patterns) interact in a way that does not result in overall disorder, as in the homeostasis (the keeping of a living body's conditions within the range it needs) of the relation of mind and body, or the integration of software with hardware.

As an observation about the nature of their effects, the effects of the first are additive, those of the second multiplicative, and those of the third exponential.

Applied to the abstract and the instantiated, Type 3 coherence is the relation the constraint requires, because abstract replicators and instantiated replicators are two levels of abstract causation, and the question is whether their interaction results in overall disorder or not. Type 3 coherence between them means that the two levels interact without overall disorder: the abstract replicator supports the instantiated ground it came from, and the ground, so supported, goes on giving rise to the capacity for abstraction, as body and mind each keep the other in working order. Type 3 coherence is support, and not mere compatibility or toleration, in the sense that each of the two levels keeps the other in working order. A relation of mere coexistence, or of mere avoidance of harm, does nothing to keep the interaction of the two levels from producing overall disorder, and so it is not Type 3 coherence.

Two features of Type 3 coherence bear directly on the constraint. The first feature is that Type 3 coherence is the only one of the three types of coherence that reaches across levels. Resonance (Type 1) and the working together of a set of ideas (Type 2) are coherence within a level; the relation between the abstract and the instantiated is a relation between levels, so the requirement is for Type 3 coherence, and not Type 1 or Type 2. The kind of effects points the same way. Abstraction builds on abstraction, so that abstraction capacity grows exponentially (Section VI). A coherence whose effects only add or multiply cannot keep pace with a capacity that grows exponentially, and a coherence whose effects are exponential can. The second feature is fragility. All three types are fragile, and the fragility deepens from Type 1 to Type 3: coherence, once broken, collapses entirely rather than degrading gracefully, and re-establishing it costs the full original effort. Trust shows this; it can be built over a lifetime and broken in a minute. The coherence between a civilization's abstractions and the ground that produced them is of this kind: it is maintained deliberately, or it fails, and once it has failed it is not partly there.

V. The Closed Loop: Nature, Humans, Text, and Back Again

The constraint of Type 3 coherence is grounded in a specific chain of dependency, the man-machine-nature relation, which I develop at length in *The Substrate and the Ecosystem* and in *The Geometry of Man, Machine, and Nature*. The chain runs as follows. Nature gives rise to humans; humans give rise to text, and to technology more broadly; whatever works against that ground works against the humans it produces, and so against text and technology themselves; and therefore text and technology must, in turn, be in service to nature, and must not work against the very ground that produced the humans who produced them.

The same structure holds at a more general level: the structures of reality give rise to the human capacity to understand them, which in turn gives rise to the human capacity to build things that affect reality. The

most important move is closing the loop, from reality through the human capacity to understand and to build, and back to reality. What is built, having arisen from the capacity to understand reality, must affect reality in a way that supports the stability of the very structures that allowed it to be understood and built in the first place, rather than debasing them. Absent this closure, the loop actively undermines its own future possibility.

The closing has to be done by design, because the effect of what is built returns to its builders only after a delay. A technology changes the world at once; the changed world changes the people who live in it slowly, often over generations. The chlorofluorocarbons of Section I took nearly half a century to show what they did to the ozone layer. A loop whose return arrives that late cannot be closed by waiting to see what happens and correcting afterward: by the time the effect is seen, the abstract replicator has spread everywhere it can reach. The closure has to be evaluated before the abstract replicator is released, which is the sense in which it is a design constraint.

VI. Thresholds, Not a Boolean Switch

An important qualification follows directly from the historical record. Abstraction capacity is graded: it crosses a series of thresholds over evolutionary and historical time, and it is not Boolean (simply present or absent). Fire, associated with a certain level of abstraction capacity, marks one such threshold. Tool-making, associated with the capacity to understand abstract sharpness rather than only the sharpness of one particular instantiated edge, marks another, evident already well before fully modern humans. Language, whose emergence is dated by different estimates from about fifty thousand to several hundred thousand years ago, marks a further, especially consequential threshold. After it, the exponential curve of abstraction building on abstraction accelerates dramatically, notwithstanding the much longer span it took for that capacity to first appear

at all. Theory of mind (the capacity to attribute thoughts and intentions to others), observable even in some animals, suggests that some minimal self-referential capacity predates fully symbolic abstraction as well.

The thresholds continue past language, and each shortens the interval to the next. Writing, which appears in Mesopotamia and Egypt a little more than five thousand years ago, lets an abstraction outlive the person who holds it and travel without them. Printing with movable type, in Europe in the middle of the fifteenth century, makes copies of an abstraction cheap and identical. Electronic computation, in the middle of the twentieth century, lets abstractions be run as well as copied: a procedure written down can now carry itself out. Each threshold multiplies how fast abstract replicators copy and recombine, and each therefore raises again the demand that the coherence of Section IV places on them.

The upshot is that "abstraction capacity" names a graded property, tied to something like self-referential complexity, rather than a single, sharp dividing line between species or eras that either have it or lack it entirely. Because abstraction capacity is graded, the coherence that the constraint of Type 3 coherence requires is a relationship that has to be maintained continuously, as abstraction capacity itself keeps increasing. The required coherence is not a single threshold to be crossed once.

VII. Parts and Wholeness: Why an Epistemology of Symmetry Fails Alone

The deepest layer of the argument concerns why this asymmetry between insight and safety arises at all, and the answer turns on a distinction between an epistemology of symmetry and an epistemology of continuity. An *epistemology* is a way of knowing: how anything comes to be known, checked and held. An *epistemology of symmetry* is the way of knowing that sees parts, and what stays the same when parts are exchanged or moved. An *epistemology of continuity* is the way of knowing that follows a whole through its ongoing changes. An epistemology of symmetry excels at un-

derstanding parts: discrete, well-defined components, and the finite, invariant-preserving transformations (changes that leave some chosen property exactly as it was) that relate them. By itself it does not supply an adequate grasp of wholeness: of how those parts, combined and recombined by abstract replicators propagating through a civilization, affect the whole system they are embedded in.

Because an epistemology of symmetry, by itself, does not supply an adequate grasp of wholeness, harm by mistake (harm that generates externalities) is so pervasive. *Mistake theory* is the view that harm comes mostly from errors of understanding, and not from conflicts of interest; an *externality* is a cost of an action borne by people outside the exchange. Understanding parts without understanding wholeness means that the effects of a new abstract replicator on the whole system are systematically underestimated: the cause is the epistemology in use, which is not built to track them, and not carelessness. An adequate response requires that people's choice-making capacity, their ability to choose well, be adequate to hold the choice-making power that their knowledge of causation makes available to them: as that power grows, the capacity to choose well must grow with it. That, in turn, requires an epistemology oriented toward continuity, and not only toward symmetry.

The ground of this distinction is the *Incommensuration Theorem*, a structural fact about the nature of knowing: symmetry and continuity cannot both be applied, simultaneously and fundamentally, to any event or comparison; and neither can asymmetry and discontinuity (*An Immanent Metaphysics*, aphorisms 1.43-2 and 1.43-3). The theorem leaves two pairings: symmetry with discontinuity, and asymmetry with continuity. Where asymmetry is applied fundamentally, discontinuity cannot be applied with it, and so continuity must be. As that theorem requires, *protected symmetry* (the invariants that an epistemology of symmetry holds fixed while it exchanges and moves parts) and continuity cannot both be maximized at once; where the asymmetry that generates real capability

grows (asymmetry here in the theorem's sense: a difference that is not preserved when parts are exchanged), continuity has to grow correspondingly, or the whole structure becomes unstable.

> **A Worked Case: The Safe Refrigerant**
>
> The chlorofluorocarbons of Section I show the two ways of knowing side by side. Known as parts, they were a success: the molecule was stable, did not burn, was not poisonous, and did not react with what it touched; every invariant the chemist could check held. Known as a whole, followed through the atmosphere over decades, the same stability was the harm: a molecule that reacts with nothing near the ground survives long enough to reach the stratosphere, where sunlight breaks it and its chlorine destroys ozone. The property that made the compound safe as a part is the property that made it harmful to the whole. No examination of the molecule alone could have found this, because the harm is not in the molecule; it is in where the molecule goes, and that can be known only by following it. And the harm was by mistake: nobody intended it, and the compound was made in order to be safe.

VIII. Metastability as the Governing Criterion

Bringing these threads together yields the chapter's central claim. Where a bottom-up structure (nature, or an instantiated pattern of any kind) gives rise to a new, emergent, top-down phenomenon (an abstract replicator, an idea, a technology), that new phenomenon, in turn, affects the bottom-up structure it arose from. For the whole system to persist rather than collapse, this effect has to increase, rather than diminish, the stability of the bottom-up structure. The result is what can be called *metastability* (physicists use the word for a state that lasts only until a large enough disturbance; here it names something actively maintained): a dynamic, actively

maintained equilibrium, sustained by the ongoing, mutually supportive relationship between the abstract and the instantiated, rather than a static balance that simply happens to hold.

On this account, an abstract replicator can be judged toxic precisely to the degree that it fails this test: to the degree it is not Type 3 coherent with the instantiated replicators that gave rise to the capacity to produce it in the first place. This gives the third constraint its final, general form: not merely that insight outpaces safety, but that any abstract replicator lacking the metastable, closed-loop relationship described here will, sooner or later, debase the very *substrate* (the ground of instantiated replicators, natural and human) it depends on for its own continuation.

Just-in-time logistics (delivering parts and goods only as they are needed, so that little is held in stock) is an abstract replicator of this kind: a method, shared and copied across industries, that removes the inventory buffers its supply chains once held. It made production more efficient, and, as the next chapter shows, it removed the slack that kept one failure from becoming many. Measured by the metastability criterion, it debases the ground that produced it. The criterion measures what a method does to the ground it depends on, and not the benefit the method gives, which is granted: just-in-time logistics did make production more efficient.

The criterion can be put as three questions to ask of any candidate abstract replicator before it is released. What ground produced the capacity to make it? What does its spread do to that ground? And as it spreads further, does that effect make the ground more stable, or less? An abstract replicator passes the test when the answer to the third question is "more stable" at every scale it can reach. A method that makes its ground more stable at one scale and less stable at another does not pass, since the answer has to be "more stable" at every scale. Put to just-in-time logistics, the questions go this way. The ground that produced it is a world of working suppliers, carriers and warehouses. Its spread removes the stock that let any one of them fail without stopping the others. And as it spreads further, the ground becomes less stable at every scale it reaches, so just-in-

time logistics does not pass. Chapter Six takes up the limit case: an abstract replicator that copies, extends and improves itself, and whose substrate is hostile to the ground that produced the people who made it.

IX. Summary

The third constraint shares its root with the constraint that technology is toxic: abstraction capacity, which generates abstract replicators. The closed loop connecting nature, humans and technology gives the constraint a clear question to ask of any abstract replicator: does it support or debase the stability of what produced it? Abstraction capacity is graded, crossing thresholds over time, as the historical record shows.

I ground the break-versus-build asymmetry in how building and breaking work: a built thing needs all of its conditions to hold, and breaking that thing needs only one of those conditions to fail. The history of the chlorofluorocarbons shows the break-versus-build asymmetry in practice. I show that an abstract replicator is a designed replicator, which no coevolution holds in check, so that the coherence of an abstract replicator with its ground has to be designed. I place that coherence among the three types of coherence (resonance, network coherence, and the interaction of multiple levels of abstract causation without overall disorder). I show that only the third type reaches across the levels that join the abstract to the instantiated, and that only the effects of the third type, which are exponential, keep pace with an abstraction capacity that grows exponentially. And I show that Type 3 coherence, once broken, collapses entirely. I show that the long delay in the loop requires the closure to be evaluated before release, and that each threshold of abstraction, from language to writing, printing and computation, raises that requirement again.

The third constraint developed here gives a name to something that can otherwise feel like simple bad luck: the sense that the smarter a civilization becomes, the more dangerous its own cleverness turns out to be.

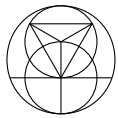
On the account given here, it is a structural consequence of a real asymmetry, between how little insight it takes to break something fundamental and how much knowledge it takes to use that insight safely. The asymmetry is rooted in the same abstraction capacity that makes technology and language possible in the first place. The remedy is to hold every abstract replicator to a single, demanding standard: that it remain in metastable, Type 3 coherent relationship with the instantiated ground that made it possible, supporting rather than debasing the very structures it ultimately depends on. Halting abstraction is neither possible nor, on this account, desirable.

In this chapter I give a structural diagnosis and a governing criterion. The structural diagnosis is that the asymmetry between insight and safety is rooted in abstraction itself, and the governing criterion is metastability. The next chapter shows what failing the criterion looks like when a whole civilization has removed its slack.

KEY CONCEPTS IN THIS CHAPTER

- **Third constraint (break-versus-build asymmetry):** the insight required to create a powerful technology is far smaller than the knowledge required to use it safely.
- **Constraints stated as absolutes:** technology is toxic and people are predators are stated as one hundred percent true, so that any solution that passes under the harder statement also handles the circumstances we actually face; the evidence for people being predators is in the body.
- **Abstract replicator:** an idea or tool, extracted from a particular instance by abstraction, capable of being shared, recombined, and propagated independently of its origin.
- **Instantiated replicator:** the concrete, embodied natural or human structure from which an abstract replicator's capacity to exist was originally drawn.
- **Building and breaking:** a built thing works only if all of its conditions hold, and breaks if any one fails; building is a conjunction and breaking a disjunction, which is the structural reason for the asymmetry.
- **Evolved and designed replicators:** evolved replicators are held in check by co-evolution; designed replicators, including every abstract replicator, are not, so their coherence with their ground has to be designed.
- **Three types of coherence:** Type 1 coherence is the coherence of resonance, akin to that found in a laser; Type 2 coherence is network coherence, such as a logical net, or a belief system coherent as a set of ideas working together; Type 3 coherence is the coherence in which multiple levels of abstract causation interact without overall disorder, as in the homeostasis of the relation of mind and body, or the integration of software with hardware. Their effects are additive, multiplicative and exponential, in that order; all three, once broken, collapse entirely.

- **Type 3 coherence between the abstract and the instantiated:** a relationship stronger than mere compatibility, in which an abstract replicator actively supports, rather than merely tolerates, the continued integrity of its instantiated ground, so that the two levels interact without overall disorder; it is the only type that reaches across levels, and the only one whose effects keep pace with an exponentially growing abstraction capacity.
- **Delay in the loop:** the effect of what is built returns to its builders only after a delay, often of generations, so the closure has to be evaluated before an abstract replicator is released.
- **Abstraction thresholds:** the graded, non-Boolean increase of abstraction capacity across evolutionary and historical time, marked by fire, tool-making, language, and their consequent acceleration, and continued by writing, printing and computation, each of which raises again the demand for coherence.
- **Epistemology of symmetry versus continuity:** the distinction between an epistemic mode suited to discrete parts and invariants, and one suited to tracking whole-system, ongoing effects.
- **Incommensuration Theorem:** symmetry and continuity cannot both be applied, simultaneously and fundamentally, to any event or comparison; and neither can asymmetry and discontinuity. It is a structural fact about the nature of knowing, and the ground of the distinction above.
- **Metastability:** a dynamic, actively maintained equilibrium between abstract and instantiated replicators; the governing criterion for whether a given abstract replicator is safe to build, put as three questions: what ground produced it, what its spread does to that ground, and whether that effect makes the ground more stable or less as it spreads.



Chapter Three

The Propagation of Failure

On Cascading Fragility in Tightly Coupled Systems

The metastability constraint developed in the preceding chapter is a design criterion: a structural test that any abstract replicator either passes or fails. This chapter examines what the failure looks like in practice, through a detailed case study of cascading systems failure.

ABSTRACT

When a disruption strikes a single domain within a tightly coupled complex system, the resulting failure does not remain contained. (Systems are *tightly coupled* when each depends on the continued normal working of the others, with little slack between them.) It propagates outward, through every system that is coupled to the one that failed, at the speed of the coupling rather than at the speed of the original threat. Each downstream system has its own failure modes, which are triggered not by the original disruption but by the failure of the system upstream of it. The real danger of any systemic crisis is therefore the cascading chain of secondary, tertiary, and higher-order failures that the initial perturbation sets in motion, and not the first-order effect itself. This chapter develops that argument using the early weeks of the global pandemic of 2020 as a case study, and extracts from it a general principle about the relationship between tight coupling, efficiency, and catastrophic fragility.

I. The Nature of the Initial Perturbation

Every cascading failure begins with a single-domain event. A pathogen enters human circulation. A financial instrument defaults. A power grid loses a node. The distinguishing feature of a catastrophic cascade is the degree of coupling between the domain in which it occurs and all of the other domains that depend on that domain's continued normal function, and not the severity of this initial event. The sociologist Charles Perrow named this property *tight coupling* in *Normal Accidents* (1984), his study of accidents in high-risk technologies: in a tightly coupled system, processes happen fast, cannot be turned off, and failed parts cannot be isolated from the rest.

In January 2020, a novel respiratory virus began spreading in a densely populated region of China. Its characteristics were unremarkable relative to the spectrum of possible pathogens: a single-stranded RNA virus, mostly of bat origin, with a reproductive number (the average number of secondary infections produced by each infected individual) somewhere between two and four. That range is higher than seasonal influenza but not dramatically so. The mortality rate, to the extent it could be estimated from the numbers available at the time, was somewhere around two to three percent of confirmed cases.

Taken in isolation, these numbers describe a serious public health event but not necessarily a civilizational one. What made the situation extraordinary was the structure of the systems into which the virus was introduced. That structure was a globally interconnected civilization in which medical systems, economic systems, governance systems, supply chains, and social institutions are coupled to one another so tightly that a failure in any one of them propagates immediately into all of the others. The question underneath this case is one the next chapter answers: whether a system coupled this tightly can be controlled at all.

II. First-Order Cascade: The Medical System

The most direct consequence of a rapidly spreading pathogen is the overloading of the medical system. The overloading of the medical system is obvious. Even so, the mechanism by which that overloading becomes catastrophic deserves careful attention, because the mechanism illustrates the general pattern that repeats at every subsequent layer of the cascade.

When the number of patients requiring hospitalization exceeds the capacity of the hospitals to treat them, the immediate result is that patients must be triaged: some will be treated and some will be left to die. Reports from Italian physicians in March 2020 described exactly this: insufficient ventilators, insufficient intubation capacity, insufficient medicine, and a triage protocol in which older or weaker patients were simply not treated.

But the overload itself is only the beginning. The more consequential dynamic is that the medical workers themselves become vectors of the disease. On the cruise ship that served as an early case study, the *Diamond Princess*, quarantined at Yokohama in February 2020, nearly every physician who was not wearing a full containment suit became infected, even those wearing high-filtration masks and goggles. Among the first infected was a quarantine officer who had worked aboard, handing out health questionnaires, in a mask and gloves and without a full protective suit. Each physician who becomes ill or dies is not simply one fewer caregiver. Each one represents the loss of all the patients that physician would have treated. The loss compounds geometrically, because the patients who are not treated remain infectious longer, producing more cases, which further overload the system.

The hospital itself, which is the institution designed to heal, becomes one of the most dangerous places to contract the infection. A person who arrives with a broken arm may leave with a lethal respiratory disease.

This inversion, in which the protective institution becomes the source of the very harm it was designed to prevent, is the signature pattern of cascading failure. It recurs at every level of the system.

> **Structural observation**
>
> The inversion of function, in which a protective institution becomes a vector, is a general consequence of coupling under overload, and it is not specific to medical systems. Any institution that must absorb the effects of a disruption can, if the disruption exceeds its capacity, begin to amplify the very problem it exists to contain. The inversion of function is a structural feature of all tightly coupled systems, not a failure of any particular institution's design.

III. The Secondary Cascade: Institutional Fragility

The overloading of the medical system does not remain confined to hospitals. It propagates outward through every institution that depends, even indirectly, on the medical system's continued normal function, or that shares the same vulnerability to the pathogen.

Consider schools. When schools are closed to slow the spread of infection, the nurses and physicians and hospital administrators who are parents of school-age children must stay home to care for them. This further reduces the already depleted medical workforce, which further degrades the medical system's capacity, which produces more untreated cases, which produces more infections, which necessitates more closures. The feedback loop is tight and self-reinforcing.

Consider prisons. A prison is a high-density enclosed environment in which one infected person can infect an entire population within days. When prison guards, who are as vulnerable to the pathogen as anyone else, become afraid to report to work, or become too ill to do so, the institution's control over its population fails. In the days when Italy's national quarantine began, riots broke out in twenty-seven Italian prisons; inmates

escaped from at least one, and several died. The consequences branch in two directions, both of them severe: the inmates either remain confined, in which case a humanitarian disaster unfolds inside the walls, or they breach the walls, in which case the infection spreads further and public safety degrades. There is no resolution of this dilemma that does not involve substantial harm.

The same pattern applies to every uniformed service. Police officers, firefighters, paramedics, National Guard soldiers deployed to support a quarantine: all of them are exposed by their duties, all of them can become ill, all of them can become vectors, and the loss of any of them degrades the capacity of the institution they serve. The deployment of the National Guard to the containment zone in New Rochelle, New York, to deliver food and clean public spaces, raised the immediate question: how do you keep the Guard from becoming infected by the very population they are deployed to serve? If they become infected, they carry the pathogen back to their barracks, their families, their next deployment. Every instrument of containment is itself vulnerable to the thing being contained.

IV. The Tertiary Cascade: Economic and Financial Systems

When the institutional cascades described above reach sufficient severity, they begin to propagate into the economic and financial systems. The mechanism is straightforward, but its consequences are not.

A substantial majority of the population, in the United States and in most developed countries, has no meaningful savings. They live from one pay period to the next. In the United States, the Federal Reserve found that in 2019 about 37 percent of adults could not cover an unexpected expense of four hundred dollars with cash or its equivalent; across the member countries of the Organisation for Economic Co-operation and Development, more than one household in three would be at risk of falling into poverty if it had to forgo three months of its income. Those two figures count only the people for whom one small expense, or three months

without income, is already beyond reach; a person can pass both tests and still have no savings that would carry a household through a long quarantine. When quarantine requires that people living this way stay home, they cannot afford to do so. They go to work sick, not out of irresponsibility but out of the impossibility of the alternative. A barista at a store in downtown Seattle was diagnosed with the virus in March 2020; a person in that job, serving customers through the days before a diagnosis, can expose hundreds of people. The moral weight of such a transmission is not borne by the barista. It is borne by the economic structure that made staying home impossible.

The financial effects compound rapidly. Those who are hospitalized and survive may accumulate medical debts in the hundreds of thousands of dollars. These debts cannot be repaid. The resulting bankruptcies propagate through the financial system, weakening creditors, reducing lending, and depressing economic activity. The contraction of economic activity reduces tax revenues, which reduces the capacity of governments to fund the very institutions (medical, military, social) that are needed to manage the crisis. Another feedback loop, another layer of amplification.

At the level of capital markets, the disruption was severe enough that market-wide *circuit breakers* (automatic pauses in trading, set off by a large fall in prices) halted trading on the New York Stock Exchange for fifteen minutes, more than once that month, to prevent a freefall. But the financial-market disruption, while dramatic, is less consequential than the disruption to the underlying productive economy. When people cannot go to work, goods are not produced, services are not provided, supply chains are interrupted, and the real productive capacity of the economy contracts. This loss is not speculative: it is a loss of actual output, of things made and services rendered, and no financial instrument can recover it.

Geopolitical Opportunism

At the same time that the pandemic was disrupting medical and economic systems, other actors were exploiting the distraction. North Korea conducted missile tests. Russia refused the production cuts that the Organization of the Petroleum Exporting Countries (OPEC) proposed, and Saudi Arabia answered with a price war that collapsed the price of oil. These events are connected. When a major power's attention is consumed by an internal crisis, the strategic calculus of every other power shifts. The pandemic did not merely disrupt health and economics. It shifted the whole geopolitical situation by taking up the scarcest resource of governance, which is attention itself: each government had only so much attention, and the pandemic took it, so that each government could watch less of everything else.

V. The Vulnerability of Attention

One of the less obvious consequences of a cascading systemic crisis is that institutions stop watching what they normally watch, because the crisis takes their attention. Every hour that a government spends managing the pandemic is an hour not spent on cybersecurity, on infrastructure maintenance, on regulatory enforcement, on intelligence analysis, or on any of the other functions that normally protect the society from other categories of threat.

This creates a secondary vulnerability that is distinct from the primary crisis. During a pandemic, physical crimes become harder to commit, because streets are empty and enforcement is heightened. But crimes that do not require physical presence, particularly in the cyber domain, become easier, because the institutions that normally detect and respond to them have their attention elsewhere.

The principle is general: attention is itself a finite resource, and its reallocation creates openings in every domain from which it is withdrawn. A crisis in domain A does not merely damage domain A. It weakens the defenses of domains B through Z by drawing attention away from them. The withdrawal of attention is another form of the tight-coupling problem. Where one institution is responsible for protecting multiple domains, a crisis severe enough to consume the full capacity of that institution in one domain leaves all the other domains exposed.

VI. The Coordination Problem

In the early weeks of the pandemic, there was no shortage of competent actors willing to contribute to the response. Universities, startups, established companies, independent researchers, maker communities, medical professionals working outside their normal roles: many people with relevant expertise and capacity stepped forward.

The problem was not competence. The problem was coordination. The situation was one in which many signals were being transmitted but no coherent pattern was being received. Many competent people were working hard, but they were working in parallel, without shared situational awareness, without compatible frameworks, and without any mechanism for integrating their efforts into a coherent whole. The result was analogous to many musicians playing simultaneously, each performing well, but producing cacophony rather than music because they are not playing the same piece.

This coordination failure is itself a consequence of the same tight-coupling problem. In a system where many domains are failing simultaneously, there is no stable vantage point from which to coordinate across them. The institutions that normally perform coordination functions, governments in particular, are themselves overwhelmed by the demands of the crisis in their own primary domain. They cannot simultaneously man-

age their own response and coordinate the responses of everyone else. The coordination function requires exactly the surplus of capacity and attention that the crisis has already consumed.

> **Design implication**
>
> If coordination requires surplus capacity, and crisis eliminates surplus capacity, then effective crisis coordination must be designed into the system in advance, not improvised under pressure. This points toward a requirement for pre-established coordination protocols, practiced relationships across domains, and institutional structures that can activate automatically when a threshold is crossed, without requiring the attention of already-overloaded decision-makers.

VII. Why Modern Coupling Makes It Worse

It is tempting to suppose that modern medicine and modern technology should make a pandemic more manageable than pandemics in earlier centuries. In 1918, the influenza pandemic killed an estimated fifty million people at a time when the global population was less than a quarter of its current size. Scaled proportionally, a comparable event today would kill two hundred million people. The figure of two hundred million deaths is a worst-case trajectory, not a forecast. Surely, the argument goes, modern medicine should reduce that number, not increase it.

The argument considers only the medical dimension and leaves out the coupling dimension; once coupling is counted, its conclusion does not follow.

Modern systems are more efficient than historical ones. They are also more tightly coupled. Air travel connects every city to every other city within forty-eight hours. Global supply chains mean that components manufactured on one continent are assembled on another and sold on a third. Just-in-time logistics, the abstract replicator that Chapter Two measured against the metastability criterion, have eliminated the inventory

buffers that previously provided resilience. The same integration that produces efficiency produces fragility, because every link in the chain is also a potential path for propagation.

At the same time, the underlying health of the population is in some respects worse than it was a century ago. Chronic exposure to environmental pollution, diets high in processed sugar and low in nutritional density, sedentary lifestyles, and a demographic shift toward an older population have collectively weakened the immune baseline against which any new pathogen is measured. Obesity is one measure of it: the share of adults in the United States with obesity rose from 30.5 percent in 1999 and 2000 to 41.9 percent in the years from 2017 to March 2020, and the Centers for Disease Control and Prevention found that obesity worsens the outcomes of COVID-19 (the disease the virus of 2020 causes), raising the risk of severe illness, hospitalization and death. The medicine is better, but the patients are more vulnerable, and there are more of them, and they are more densely packed, and they are more interconnected.

The net effect is that the medical advantages of modernity may be partially or entirely offset by the coupling advantages that modernity has also produced, advantages from the pathogen's perspective. Isolation, which was the primary defense in 1918, is structurally harder to achieve in a system where everything depends on everything else. A quarantine that lasts two months would be sufficient to extinguish the pathogen. The ground is the incubation period: about 97.5 percent of those infected who develop symptoms do so within 11.5 days of infection (Lauer and colleagues, 2020), so a quarantine in which contact actually stops for two months outlasts several complete cycles of infection, and a chain of transmission that cannot reach a new host ends. But achieving a real quarantine, one in which movement and contact actually stop, is nearly impossible when so much of the population cannot afford to stop working and so much of the infrastructure cannot function without continuous human operation.

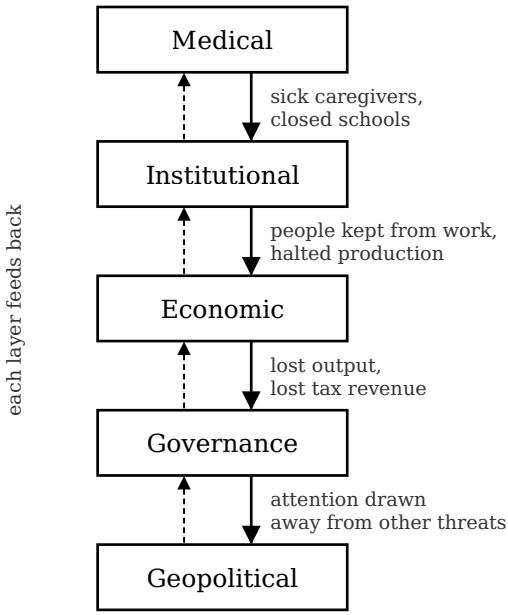
The mechanisms this chapter names all appeared in the years that followed: overloaded hospitals, infected caregivers, prison unrest, trading halts and broken supply chains. The World Health Organization estim-

ated the *excess deaths* of 2020 and 2021 (deaths above the number expected in ordinary years) at about 14.9 million, within a range of 13.3 million to 16.6 million. That the world of 2020 continued is an instance of holding: keeping a coupled system within a safe range for a time, by continued effort. Chapter Four shows that holding cannot be indefinite. What held the world of 2020 within a range was holding of exactly that kind: closures, vaccines developed and given at speed, and transfers of money to households and firms.

VIII. The General Principle

The pandemic is a case study, but the principle that the pandemic illustrates is general. It can be stated simply:

> *In any system of tightly coupled domains, a disruption in one domain will propagate through all coupled domains at the speed of the coupling, not at the speed of the original threat. Each domain has its own failure modes, which are triggered not by the original disruption but by the failure of the domain upstream. The total damage therefore exceeds, and in severe cases vastly exceeds, the damage that the original disruption could have caused in isolation.*



The cascade of Chapter Three. Solid arrows: what crosses each join as the failure passes down from one layer to the next. Dashed arrows: the return path, by which each layer feeds back into the layer above it.

This principle has several important corollaries.

First, the severity of a systemic crisis cannot be assessed by examining the initial perturbation alone. A virus with a two-percent mortality rate can, through cascading effects, produce outcomes far more devastating than the two percent would suggest. The relevant measure is the fragility of the systems the pathogen disrupts, and not the lethality of the pathogen.

Second, the institutions designed to protect against failure are themselves vulnerable to the same failure. Hospitals become infection sites. Enforcement agencies become vectors. Coordination bodies become overwhelmed. The signature of a cascading failure is the inversion of protective function: the shield becomes the wound.

Third, the efficiency of a system and the fragility of a system are not merely correlated but are, to a significant degree, the same property viewed from different perspectives. Efficiency is achieved by removing redundancy, eliminating buffers, tightening coupling, and reducing the slack that would allow each component to absorb a shock independently. But those are precisely the features that would prevent a single-domain failure from propagating across the entire system. To optimize for efficiency is, in many cases, to optimize for catastrophic fragility. The design implication of the coordination problem gives part of the answer to whether a system can have both: pre-established coordination protocols, and practiced relationships across domains, hold in reserve the capacity that a crisis consumes.

Hardening the Grid: Thresholds, Phases and Enduring Law

The general principle gives a practical answer for the electrical grid, a domain to which transportation, communication and the whole economy are coupled. This section is for the reader who would draft or support a law to protect the grid; its examples are from the United States. *Grid hardening* is making the electrical power grid able to withstand events that would otherwise destroy it. The answer concerns which threats to model, how to divide the work, and how a law can keep preparedness in place.

The Inequality with the Pandemic

What makes the *Overton window* work for grid hardening (the range of ideas that the public and lawmakers are ready to take seriously at a given time) is a comparison, set up as an inequality against the pandemic.

A grid shutdown from a Carrington-style event or an *electromagnetic pulse* is at least as likely as another pandemic, because the sun is always there and does not depend on anything humans do. A *Carrington event* is a solar storm great enough to damage electrical equipment across wide areas; a Carrington-style event is one of that size. An electromagnetic pulse is a burst of electromagnetic energy, from the sun or from a nuclear

explosion, that can destroy electrical equipment. The two differ in a way that matters for defense: a solar storm drives a slow current through long power lines and overheats transformers, while a nuclear pulse does that and also delivers a fast surge that destroys electronics.

A grid shutdown is at least as impactful as another pandemic, and probably considerably more. Having everybody stay home costs far less than having the whole economy stop because cars and machines stop. If gas stations have no electricity, the pumps do not work, so transportation stops. Communication stops too, so that not even emergency messages can get out, since everyone relies on cell phones. Without the grid, civil unrest comes within weeks, not months. A trillion-dollar reset of the economy is small compared with a grid shutdown that lasts more than a couple of weeks.

And a grid shutdown is easier to address. Instead of having to deal with the medical system and the complexities of the human body and the ecosystem, we can defend against a grid shutdown with known technology that we already have, and deploying that technology is cheaper than buying a million respirators.

So a grid shutdown is at least as likely, more impactful, easier to defend against, and cheaper to respond to. For the same level of risk mitigation, the cost and the red tape are something like two to three orders of magnitude smaller.

Setting the Threshold

A threshold has to be set for what counts as a threat that needs modeling, because the number of things we could model grows exponentially the lower the threshold is. A first filter could be: no wide-scale power outages, meaning more than ten states out of fifty for more than two weeks. I picked two weeks because at a certain point the harms become irreversible, and not because nothing significant is harmed in a shorter time. The harm I mean is the United States failing to reboot as a country, with civil

unrest, nobody knowing what is going on, and *balkanization* (the breaking up of a country into separate, often hostile, regions), and not harm to particular communities or groups.

I treat multiple simultaneous nuclear meltdowns as a separate risk to the continued existence of the country, distinct from the Carrington event. For the Carrington *threat model* (the account of what a threat is, how it would act, and what it would take to mitigate it) I set the threshold at the level where civil unrest starts. For persuading people, the threshold should be higher than everything associated with the pandemic, direct and indirect, by at least a factor of four. These threats are such a clear and present danger that addressing them handles a lot of other things as side effects.

The Line That Makes a Blackout Categorical

In a blackout from general mechanical failure, like the 2003 Northeastern blackout, the cascade did not destroy much equipment. Power was restored gradually as crews worked out how to turn things back on, and people had no doubt the power was coming back. With a Carrington event, or with a targeted attack or an electromagnetic pulse, the equipment is broken thoroughly and cannot be fixed quickly, and people will have substantial doubt whether the power will ever come back. At that point come wide-scale civil unrest and a reset.

Anything that causes a blackout long enough that there is well-founded doubt, because of the real situation, that the power will ever return, is a *categorical situation* (a situation different in kind, and not only in degree, from ordinary ones, and so needing a response of its own kind). So such a blackout has to be addressed categorically.

Four Phases of Sense-Making

Treat the work as four phases, because the sense-making differs for each. The first phase is *threat modeling*. The second phase is the *remediation* methods (remediation is the technical work of fixing a weakness): the technical solutions that would address each threat. The third phase is the

method by which the solutions designed in phase two are deployed. The fourth phase is how to make the deployments of phase three remain in place for the long term.

The resolutions of a law will mostly address phases three and four. But we need to have gone through all four phases well enough to know what to put in an outline draft of the law. The four phases are four epochs of sense-making.

Partitioning the Threat Space

Once the threshold is set, the threats that meet it form a finite, discrete list. There are two kinds of electromagnetic-pulse-style event, solar and nuclear, and two kinds of deliberate attack: physical attack on *critical substations* (substations whose loss would cascade into wide failure of the grid) and coordinated cyber attack.

Natural disasters such as hurricanes are not specific to the grid, so they do not make the list. Ordinary mechanical failure is the most frequent and the best tended to, and the least likely to cause genuine catastrophe, so it does not make the list either. The listed threats are either precisely targeted at what we cannot fix quickly, or take out everything at once so that we cannot repair it fast enough.

Because the list is finite and agreed, we can partition the threat modeling. The first phase has four parts, one per threat, each asking how well we understand the threat and what it would take to mitigate it. The whole job becomes four phases of four sub-projects, and that is a very tractable problem.

For each threat, the first product is two or three pages that make clear the threat is real, and that set out what we know and what we still need to know. That way everyone drawn in later works from the same scope.

Finding the cyber threat

The cyber threat model is found by looking for entanglements between critical infrastructure and computer systems. That means anywhere a phone line or internet connection talks, in a control-focused way, to infra-

structure that shows up as key in the other threat models, for example the handful of critical substations whose loss cascades. The degree to which a computer could be used to destroy that infrastructure is the threat model, and we identify its particulars.

The Law Is for Endurance More Than for the Fix

A law here has to concentrate on a small number of topics and handle them comprehensively; otherwise it will get mired. In the time a law has, it can do perhaps one or two, at most three things. They have to be the right things, done so well that they stay done for at least ten to fifteen years. A law of this kind is holding, in the sense of Section VII: it keeps a coupled system within a safe range for a stated time, and it has to be renewed, since Chapter Four shows that no holding is indefinite.

A law here has to do two things. It has to get a solution in place that addresses the threats mechanically, on the ground. And it has to keep that preparedness in place so that it does not go away.

Suppose, as a made-up scenario, that a certain sum buys the *redundancy* needed in the right places (spare capacity or equipment held in reserve so that a system can keep working when parts fail). Ten years go by, and the fix that would work three years from now no longer works thirteen years from now, because the redundant infrastructure that made the system robust was co-opted and diverted to other uses.

Pandemic preparedness went the same way: a decade before the pandemic of 2020 the country had a robust strategy, and in the years just before the pandemic much of it was diverted. A country can have the best response plan, but if intervening governments co-opt the resources between putting the plan in place and the actual event, the plan cannot be carried out when the event comes. So the law is more about the enduring quality than about getting the fix in place; the fix itself should be put in place as fast as it can be.

IX. Summary

Tight coupling produces cascading fragility. Each step in the cascade follows from the one before it through mechanisms that are empirically observable and logically necessary given the coupling relationships involved. The claim is that the architecture of tightly coupled systems makes such cascades possible. The specific features of modern civilization (global travel, just-in-time logistics, economic precarity, institutional overload) make them more likely and more severe than they would be in less coupled systems. The claim is not that every crisis produces a catastrophic cascade.

In this chapter I identify the mechanisms by which failures propagate. Tightly coupled systems are architecturally predisposed to fail catastrophically when a perturbation exceeds the absorptive capacity of any single domain; the timing, magnitude and endpoint of a particular cascade depend on the case.

The lesson of the pandemic, and of every cascading systemic failure, is that the real threat is never the initial event. The real threat is the architecture of the system into which that event is introduced.

A pathogen is a medical problem. A pathogen introduced into a tightly coupled civilization is a civilizational problem, because the coupling ensures that the medical failure propagates into institutional failure, which propagates into economic failure, which propagates into governance failure, which propagates into geopolitical instability, each layer triggering failure modes that are native to its own domain and that feed back into the layers above and below it.

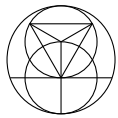
The modern world has been designed for efficiency. Its supply chains are lean. Its institutions are specialized. Its populations are interconnected. Its workers are economically precarious. All of these features produce extraordinary performance under normal conditions. All of them produce extraordinary fragility under abnormal ones.

Section VIII has shown that efficiency and fragility are, to a significant degree, the same property viewed from different perspectives, and that pre-established coordination protocols and practiced relationships across domains give part of the answer, by holding in reserve the capacity that a crisis consumes. The question that the pandemic raises, and that no merely medical response can answer, is whether it is possible to design a civilization that is both efficient and resilient, or whether efficiency and resilience are in tension by their nature. If efficiency and resilience are in tension by their nature, the question is also which of the two we are prepared to sacrifice. Beneath it lies the question that Chapter Four answers: whether a tightly coupled civilization can be controlled at all.

KEY CONCEPTS IN THIS CHAPTER

- **Cascading failure:** a failure in one domain that propagates through coupled domains, each triggered not by the original disruption but by the failure of the domain immediately upstream.
- **Tight coupling:** a relationship between systems in which the functioning of each depends on the continued functioning of the others, with minimal slack, buffering, or independent absorptive capacity (Charles Perrow's term).
- **Reproductive number:** the average number of secondary infections produced by a single infected individual in a susceptible population; a measure of a pathogen's spreading capacity.
- **Inversion of protective function:** the phenomenon by which an institution designed to protect against a particular harm becomes, under overload, a source of that harm; the signature pattern of cascading failure.
- **Coordination failure:** the breakdown of effective cooperation among competent actors, occurring when no participant has the surplus capacity needed to integrate the efforts of the others.
- **Efficiency-fragility tradeoff:** the structural tension between optimizing a system for performance under normal conditions and maintaining its capacity to absorb shocks under abnormal conditions; the removal of redundancy that produces efficiency also removes the buffers that prevent cascading failure.
- **Attention as finite resource:** institutional attention is a scarce good whose reallocation during crisis creates secondary vulnerabilities in every domain from which attention is withdrawn.
- **The inequality with the pandemic:** a grid shutdown from a Carrington-style event or an electromagnetic pulse is at least as likely as another pandemic, more impactful, easier to defend against, and cheaper to respond to.

- **Categorical blackout:** a blackout long enough for well-founded doubt that the power will ever return; it has to be addressed categorically.
- **Four phases of four threats:** threat modeling, remediation methods, deployment, and keeping deployments in place, for solar and nuclear electromagnetic pulses, physical attack on critical substations, and coordinated cyber attack.
- **Enduring law:** a law's main work is keeping preparedness in place against the diversion of its resources; the law is for endurance more than for the fix.



Chapter Four

Relative and Absolute Limits

Two Forms of Argument Concerning the Controllability of Artificial Intelligence

The cascading fragility described in the preceding chapter illustrates a general architectural problem: the tight coupling which produces a system's performance is also the source of its vulnerability. This chapter asks whether this problem has a formal boundary: whether there is a point at which a system's complexity places it beyond the reach of any controller, not because the cost of control is high, but because the question of control has ceased to have an answer. The answer uses a few results from the theory of computation; each is explained where it first appears.

ABSTRACT

Arguments that a sufficiently general artificial intelligence cannot be controlled divide cleanly into two forms, and a great deal of confusion in the field comes from not distinguishing them. A *relative* argument establishes an inequality: as some parameter of the system is raised, some cost grows faster, and grows in a manner that cannot be caught up with. An *absolute* argument establishes a threshold: once a certain structural feature is present, the question of predictability has passed out of the domain in which any amount of resource can help. This chapter develops both. For the relative form it sets out the trinity of compute (time, space, and energy) as the three quantities against which any growth in capability must be measured. For the absolute form it identifies the three minimal features that define computational universality more cleanly than the Turing

machine does (a *Turing machine* is Alan Turing's model of a computer: a device that reads and writes symbols on an unlimited tape by fixed rules). It shows that once a system possesses those features, every *safety predicate* (a yes-or-no property of the system, such as "never enters a harmful state") collapses into the halting problem by Rice's theorem. It then develops a noise-amplification measure for a ring of controller, agent and world, in which each part amplifies the noise it receives above a critical level. It shows that the ring exhibits a second-law inequality in that measure: uncertainty increases at every traversal of the loop, as disorder increases under the second law of thermodynamics. It closes with the observation that error correction, which is the natural rejoinder, is definable at the level of bits (the zeros and ones in which a signal is written) and is not definable at the level of abstraction at which *alignment* actually lives. (Alignment is a consistent correspondence between a system's internal reference values and its effects in the world, maintained over time.)

I. Why the Distinction Matters

There is a characteristic failure mode in discussions of this kind. An argument is offered with very high generality, and the question comes back: what does that actually apply to? So the argument is narrowed and made specific, and now the question comes back the other way: that is a special case, and we are not building that.

Between the two questions a great deal of good work is lost, and the loss is avoidable if we recognize that there are two structurally different things an argument in this space can accomplish, and that they answer different questions.

A relative argument sets up an inequality. It says: hold everything else fixed, raise this one parameter, and observe that these other quantities rise faster, and rise in a way that cannot be caught up with. Where such an inequality can be shown, one has made a strong-form argument of impracticality. Where the gap can never be closed, at any scale and at any future time, the impracticality is permanent. A barrier established as perman-

ently intractable is impossible, in the only sense of the word that has ever governed action: the sense in which a perpetual-motion machine is impossible (*Why No Control Suffices* sets out this standard).

An absolute argument sets up a threshold. It says: here is a structural condition; above it, the question you are asking has no answer, not because the answer is expensive but because there is no answer to be had.

The two interlock. When someone argues from a fixed point (*but this particular system is controllable*), one can switch to the relative argument and show what happens as the system is scaled or perturbed. When someone argues that the relative costs might be met by future capability, one can switch to the absolute argument and show that the destination is not merely expensive but empty. That mutual latching is what makes the pair strong, and it is worth constructing both even though only one argument for a thing is strictly required. There are hundreds of derivations of the Pythagorean theorem (Elisha Loomis gathered more than three hundred of them in *The Pythagorean Proposition*), and the abundance is useful, because it lets the presenter choose the language that the particular audience already speaks.

II. The Trinity of Compute

For the relative form we need quantities to measure against. There are three, and they are not arbitrary.

The first quantity is *time*: the minimum number of sequentially contingent steps that must be performed for a result to be arrived at. Suppose we granted the maximum possible parallelism: what remains is the *circuit depth*, the longest chain of steps in which each step must wait for the one before, the chain of absolute contingencies connecting one part of the computation to another. The circuit depth is a true measure of time, and adding more hardware to the computation does not reduce the circuit depth.

The second quantity is *space*: the number of memory locations or wires required. Memory across space and memory across time are interchangeable in kind: either two points are connected by a wire, or one cell stores a value that is picked up later.

The third quantity is *energy*: the physical cost of the sheer amount of computing that is performed. There is work on *reversible computation* (computing in which no information is erased, which in principle lowers the energy each step must cost) that seeks to mitigate this. But where there is a strict lower bound on the number of operations required for a given result, there is a corresponding lower bound on the energy. Even where reversible computation lowers the energy, the time does not fall, and the relative argument needs only one of the three to outrun the system it measures.

In the relative argument, let the independent variable be some measure of the system (the complexity of the circuit, the depth of abstraction, whatever quantity is being scaled). Raise it linearly. Observe that at least one of time, space, and energy rises faster. (In the cases this chapter considers, the rise is very probably exponential; the argument needs only that it is faster and cannot be caught up with.) Observe further that the smallest value of the input variable that would suffice for the task in question already forces at least one of the three well past the threshold of practicality.

That constellation (this rises, and that rises faster, and it rises fast enough that the gap cannot be closed) is the whole of the relative argument. It is a familiar form. The relative argument is what an order-of-growth analysis does (the comparison of how fast costs rise as a problem grows), generalized from algorithms to systems.

III. The Threshold: Three Features, Not One Machine

The absolute argument requires that we say precisely where the threshold of computational universality lies.

The Turing machine was an answer to the question of what it means for something to be computable, and as a simplification of an otherwise intractable problem it is brilliant. It is also an awkward construction. There is a simpler way to draw the boundary, and drawing it simply matters here, because we need to be able to inspect a candidate system and say whether it is above or below the line.

Three features define the boundary, the threshold of universality, exactly. There must be some notion of *sequential contingency*: that this happens, and then that. There must be some notion of *conditionality* that links data to *control flow* (the order in which a program's steps are carried out): what happens next depends on what the data are. And there must be some notion of *backward reference* in the control flow: recursion (a procedure that calls itself), or an explicit looping construct, or something in the nature of a jump that can point backwards.

These three features are the classical ones: Corrado Böhm and Giuseppe Jacopini showed in 1966 that sequence, selection and iteration suffice to express any computable function, and those are sequential contingency, conditionality and backward reference under other names. What matters here is their use as a test: a candidate system can be inspected and placed above or below the line. The features act on a store of data. Where that store can grow as the computation needs, the absolute argument of the next section applies. Where the store is fixed, every property can in principle be checked state by state, but the number of states doubles with each added bit, and the relative argument of the previous section applies instead. Either way, control fails.

Anyone who works with programming languages will observe correctly that this is a low bar. You do not need much in a language to make it universal in this sense. The low bar is the point, and yet the bar is not met by most of the *toy models* (deliberately simplified models used to reason about a problem) that are being used to reason about controllability. A ring of communicating nodes with fixed message-passing, for instance, does not have it.

So the minimum interesting toy model is essentially a very rudimentary *virtual machine*, a program that runs other programs. We may represent signals as strings of bits and treat abstraction as a matter of what symbol set flows through a channel. But if what is passing through the channel is a program, and if the circuit under discussion implements a virtual machine that runs the program, then the relation between the input data state and the output data state has become intractable in principle.

The same three features are present in the coupled institutions of Chapter Three, in the pandemic of 2020. Failures follow one another in sequence: the medical failure is followed by the institutional, the economic and the political. What each institution does next depends on the data it receives: case counts decide closures, triage and deployments. And the control flow turns back on itself: closures deplete the medical workforce, which produces more cases, which produce more closures. A civilization of tightly coupled institutions, with markets and learning systems running inside it, is therefore itself a system above the threshold, and what this chapter shows of such systems holds of it.

Placing a civilization above the threshold raises two further questions: what its store of data is, and what its program is. The store is everything its institutions record and keep: case counts, accounts, prices, money, laws, and the stocks of knowledge that institutions hold; all of these grow as the civilization needs. The program is the rules, laws and procedures that institutions follow, and those are themselves rewritten in response to the data, as the closures of 2020 were rewritten week by week. A new rule passes through the channels of a civilization as data and is then run as a program, which is exactly the condition that the minimum interesting toy model asks for: what passes through the channel is a program. Markets and learning systems do the same within each cycle: a trading algorithm or a learning system receives new data and changes the procedure it runs. Whether the store is taken to grow or to be fixed, control fails, as this section has shown. In the ring of controller, agent and world that Section V

sets out, the controller of a civilization is the institutions that set its rules; the agent is the people and markets acting under those rules; and the world is the living and built world they act on.

Narrow and General by Definition, Not by Limit

The three features of Section III mark the threshold of universality, the line above which a system can run any program. A second set of three features separates narrow from general artificial intelligence. I distinguish narrow from general intelligence by a way of thinking that works by definition, and not by a limit approached by degrees. I look at how many *domains* the system covers in its inputs and outputs, internally and externally (a domain is a field of inputs and outputs that a system takes in or acts on, such as language or a game). I look at whether there is internal processing that has a learning process, so that it adapts to its environment. And I look at whether that adaptation can include changes to how it learns.

Single domain versus multiple domains; continuing to learn; and learning that can change how the learning happens: those features distinguish narrow from general. They do so in a definitional way, rather than through an *asymptotic limit* (a value that a continuous function approaches and never reaches, as with infinity).

Another way of thinking says that general intelligence is like infinity on a continuous function: as you approach it, you never touch it. On that view, you end up building near-infinite systems that never qualify as intelligence, while the "narrowness" is no longer narrow. The definitional way of thinking does not lead to those asymptotic problems.

On the basis of the three features of generality, a large language model system sits in the *borderlands*, the region between narrow and general. A large language model system is not specific to one topic anymore. It has multiple inputs and multiple outputs, in that it can convert things to language and turn the output of language into actions in multiple domains. A large language model system is at a threshold, an ecosystem of language modes. If we had to, we could call it "narrow level two". The sys-

tem that played the game Diplomacy (a board game won by making and breaking agreements with other players in ordinary language) is in the borderlands too, because Diplomacy is a multi-domain interaction.

IV. Rice's Theorem and the Collapse of Safety Predicates

The reason a system above the threshold has gone intractable is Rice's theorem, and the statement of the consequence is short. *Rice's theorem*, proved by Henry Rice in 1953, says this: for any property of what a program does that some programs have and others lack, no general method can decide, for every program, whether it has that property. The *halting problem* is the question whether a given program, run on a given input, will ever stop; Alan Turing proved in 1936 that no program can answer it correctly for every program and input.

Every property we care about in such a system, including the property we are calling safety, reduces to some version of the halting problem. The halting problem is not solvable. The system in question can run any program, so the halting problem arises inside it. A learning system does not stay the program it was when it was analyzed. A controller of such a system has to be right about every program the system can become, and a system with the three features and a store that can grow can become any program. So the controller has to decide the property across all programs, which is what Rice's theorem says no method can do. Therefore the *modeling criterion* (the requirement that a controller can model the system well enough to predict whether it will stay safe) fails: not partially, and not expensively, but categorically. *Why No Control Suffices; The Technical Case for the Impossibility of AGI Safety*, Chapter Seven, "Rice's Theorem and Its Application to AGI", gives this step at full length.

A reader may ask whether Rice's theorem, which concerns what a program computes, reaches a safety condition that concerns what happens in the world. It does: a safety predicate such as "never enters a harmful state" is a property of what the program does, and not of how it is writ-

ten, which is the kind of property Rice's theorem covers; and where the property refers to the world, the world adds its own unpredictability, as the next section shows.

Nothing exotic is needed for this. The notion of learning cannot be simpler than something represented as a program. The binding between learning and unpredictability is tight, because learning is itself the thing that puts a system above the threshold of universality, and above the threshold of universality predictability is gone. So the boundary on what can be known about the system is bound directly to the presence of the very capability that made it worth building.

On the Question Every Engineer Asks

A reader will ask: could we not analyze this particular system and confirm that it is safe? That question is well taken and must be answered. It runs: the halting problem is unsolvable *in general*, but we are not asking about arbitrary algorithms; we are asking about this specific one, and this specific one may be perfectly analyzable. *Formal verification*, in which a particular program is proved to meet its specification, is the question in its working form: each verified program is one analyzed point, and a learning system does not stay at the point that was analyzed.

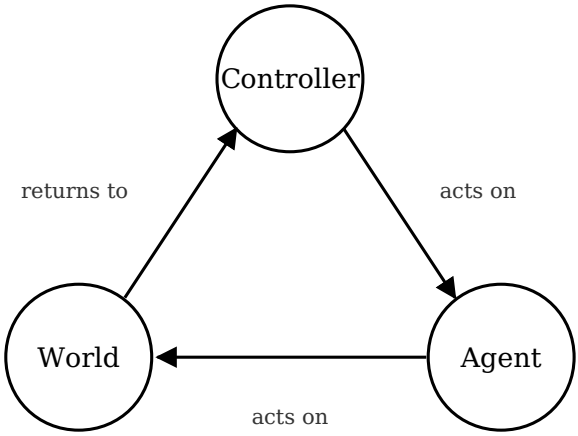
The answer proceeds by looking at the neighborhood rather than at the point. Take an algorithm whose halting behavior we can in fact predict. Represent it as compactly as possible, so that every bit counts and there is no redundancy in the specification. Now ask: if one bit of that specification changes, what is the likelihood that the algorithm moves from predictable to non-predictable? We are now looking at the penumbra (the surrounding fringe) of algorithms adjacent to the given one.

What we find is that the analyzable instances are distributed among the non-analyzable ones in the way the rationals (the fractions, which sit between any two numbers on the number line and yet are almost none of its points) are distributed among the continuum. There is an infinity of them, and throwing a dart at the space gives an essentially infinitesimal chance of hitting one. The no-free-lunch results give this its formal footing. The no-free-lunch theorems of David Wolpert and William Macready (1997) state that, averaged over all possible problems, no search or learning method does better than any other; so no method of finding the analyzable instances does better than chance over all problems. The argument does not depend on how many analyzable instances there are; it depends on their neighborhoods, and on the fact that a learning system does not stay where it was analyzed. Joel David Hamkins and Alexei Miasnikov showed in 2006 that, in one model of computation, the halting problem can be decided on a set of programs whose share of all programs approaches one as the programs grow longer; that is a result about how many analyzable instances there are, and it leaves the neighborhoods, the

> learning update and the regress as they stand. The programs that make up that share are, in their proof, programs that, before they have repeated a state, either halt or move the read head off the end of a one-way tape, and in the typical case the head falls off the tape within a few steps; the authors state that the proof is sensitive to that model of computation. A program that falls off its tape at once is analyzable because it has stopped. A system that learns is not one of those programs, so the share says nothing about the neighborhood in which a learning system moves. And the space is strictly divergent: wherever *fixed points* are found (states that a process returns unchanged, where its behavior can be pinned down), they are adjacent to regions that are not predictable. So a small quantity of noise anywhere in the specification (transmission error, learning update, environmental perturbation) moves a solvable instance into the ocean surrounding it. And any attempt to *learn* which instances are solvable is itself an algorithm subject to the same constraints, which closes the regress.

V. The Ring: Control, Agent, and World

Consider the minimal architecture in which alignment can even be posed. There are three regions arranged in a loop. There is the agent. There is the world, which the agent acts upon. And there is the controller, whose task is to ensure that the world does not enter a state inconsistent with whatever we mean by alignment. The outputs of the world return to the controller, and around it goes.



The ring. The controller acts on the agent, the agent acts on the world, and the outputs of the world return to the controller.

This ring is a generalization, not a particular design. Whatever the agent is, whatever the world is, whatever the controller is, and however they are interconnected, the loop covers the case.

Now: no external loop is required to make the interior of the agent unpredictable. If the inputs to the agent are treated as a specification (as a program which the agent's internals run), then the relation between input and output is such that changing one bit changes everything. Even with perfect knowledge of the inputs we cannot predict the outputs except in extremely specialized cases, and altering a single input bit leaves us with no idea how to predict any output bit.

The same characteristic belongs to the world. Here a narrowing must be headed off. A reader may say that the world is often easy to predict: release a bottle and it falls. True, and irrelevant. The places where the world is easy to predict are very largely artificial circumstances. The places where prediction is *necessary* are ecosystems. When we ask whether a system is aligned, the question that we are actually asking is something like: what is the resilience and homeostasis of the ecosystem, and will introdu-

cing the system kill the lawn? Alignment is in every case bound to biological systems in one way or another, and those are characteristically complex.

And the controller is in the same condition. The controller is a computer attempting to predict what the agent will do, while the agent is a system whose relation to the world is itself unpredictable. If the controller cannot predict when the agent fails to predict the world, control fails; and no amount of power in the controller changes this, because the barrier is not a resource barrier.

VI. A Second Law for the Loop

I make the ring argument quantitative here, because the quantitative form converts a philosophical claim into an inequality.

Introduce a measure of noise amplification for a *black box*, a system known only by what goes into it and what comes out: call it *hashiness*, on a scale from zero to one. A box with high hashiness has the property that a given quantity of noise on the input produces a greater quantity of noise on the output. Noise here is measured against the signal. A linear amplifier that makes signal and noise alike ten times larger leaves their ratio where it was, and its hashiness is zero. A *cryptographic hash function* is the limiting case: it is built so that a single bit of input change randomizes the entire output. The same number measures a controller's loss: hashiness is equally the probability that a controller's best possible input still fails, because of internal state the controller cannot see. The two meanings measure one property, seen once from inside the box and once from the side of the controller (*The Ceiling Below the Floor* develops both).

Assign a hashiness to each of the three regions. It is non-zero for the agent, non-zero for the world, and non-zero for the controller. If each is above some critical threshold (the divergence threshold of the ring), then the noise increases at every transition around the ring. The previous section gives the reason each region is above it: the agent, because changing

one bit of its input changes everything; the world, because the places where prediction is necessary are ecosystems; and the controller, because it is a computer attempting to predict what the agent will do. The controller's place above the level is supplied again by the good regulator theorem of Roger Conant and W. Ross Ashby, which Chapter Five states: any regulator that controls a system well must contain a model of that system. A controller too simple to amplify noise is too simple to contain a model of the agent, and so it does not control the agent; a controller able to control the agent contains the agent's amplification of noise inside it. The argument needs each region to be above the critical level; it does not need the value of that level (*The Ceiling Below the Floor* develops the measure). The level of unknowingness about the state increases *monotonically* with each traversal (at every step, and never falling back), until we have effectively no information about any bit, or at least no information about a subset of bits sufficient to determine whether the world has entered a degraded state.

A reader may ask why the controller does not measure the world again at each turn and begin fresh. A measurement reports the world at the coarseness of its instruments: it divides the states of the world into regions and reports only which region the world is in, which is the partition that Chapter Five, Section VI, shows loses what lies below it. What lies below it is what the agent and the world amplify on the next traversal. The effects that matter to alignment, the fourth- and fifth-order effects of Chapter One, return to the controller only after the delay of Chapter Two, Section V, when the harm is already done. A fresh reading renews the bits it can see; the uncertainty that grows is the uncertainty about the bits that decide whether the world has entered a degraded state.

An uncertainty that keeps growing however often the controller measures again is a second-law structure: the structure of the second law of thermodynamics, under which the disorder of a closed system never decreases. The analogy is not decorative. In a convergent system one has something like a heat pump: energy can be spent to cause convergence. Above a certain threshold of complexity in the loops, one has instead the

entropy function (disorder that grows whatever energy is spent), and no amount of energy input yields more order. The criterion reduces to something familiar in form: is this process convergent or divergent? It has the same character as asking whether an infinite series converges.

And the two arguments now combine in a specific way. If the divergence is established only relatively, then the answer is asymptotic: as the number of cycles increases, the loop converges to unpredictability in critical ways. And no one leaves the loop running for only one cycle. If the absolute argument is established, then even a single traversal already suffices.

There is a practical reason to want the absolute form, and it is not a matter of elegance. Where only the relative form is available, there is a rejoinder that cannot be dismissed: how long will this take? If it is three hundred years, some will say that they do not care. The absolute argument removes the question of timelines from the table.

A Worked Case: Automated Trading

The abstraction can be grounded in an example that has already occurred. A trading algorithm is written and set loose. The market is composed of other agents whose code we do not know, though we can observe the market itself. The controller's task is not merely to make money; it is to ensure that the market does not come apart because our agent has become caught in a runaway exchange with some other agent doing the same thing.

The question (is it possible to write a controller that prevents the market from coming apart?) is exactly the alignment question in miniature, and the circuit breakers that actually exist are an admission that the answer is no. They halted the New York Stock Exchange more than once in March 2020, as Chapter Three describes; each halt held the market for fifteen minutes, and the market fell far enough to trip them again on later days. The case in the record that matches the runaway exchange itself is the Flash Crash of 6 May 2010. That afternoon the Dow Jones Industrial Average (a widely followed index of large American companies' share prices) fell nearly a thousand points within minutes and recovered much of the fall before the close; the joint report of the Commodity Futures Trading Commission and the Securities and Exchange Commission (30 September 2010) traced it to an automated program selling stock-index futures in large volume, and to the response of high-frequency trading programs to that selling. Circuit breakers for single stocks, which pause trading in a stock that moves ten percent within five minutes, were introduced in June 2010, a month after the crash. The trading algorithm is the agent; the market, made of other agents whose code we do not know, is the world; the risk controls and the exchange's circuit breakers are the controller. The value of the Flash Crash as an example is that the market is an ecosystem one can relate to, and that, by the mapping of agent, world and controller, the market is equivalent under alignment to the general case. That equivalence leaves no room for saying that the Flash Crash is a special case.

VII. Why Error Correction Does Not Answer This

The natural rejoinder is *error correction*: detecting when a signal or a system has departed from what was intended, and restoring it. It is offered often and with confidence: there is no such thing as a non-controllable system, because errors can always be detected and corrected.

The rejoinder works at the level at which error correction is defined, and does not extend to the level at which alignment lives.

Consider what error correction actually requires. There is a symbol set: for bits, zero and one. There is a redundancy scheme such that if one bit is flipped the flip can be detected, and with enough redundancy the correct value recovered. Geometrically, one arranges the space of all possible transmissions so that valid signals are separated by *buffer regions* (margins of invalid signals around each valid one), and a received signal is corrected to its nearest valid neighbor.

Even at this level the correction is never perfect. Claude Shannon's theorem shows that, below a channel's capacity (the most information a path for signals can carry reliably), the chance of an uncorrected error can be made as small as one likes, by longer codes and more redundancy; no finite code makes that chance zero. A chance that is small but not zero, taken again at every use of the channel, over all future time, adds up to a certainty that the error happens (*Ape Trap* sets out this step in its essay on the error correction regress).

Now let the symbol be an algorithm. Every possible algorithm is now a symbol that may be transmitted, and a message is a sequence of them. The space of possible symbols is not merely large; its dimensionality is unbounded. What would it mean to construct a buffer region around each algorithm in that space, such that the wrong algorithm could be corrected to the right one?

Put concretely: what error-correcting code would detect that this variable name should have been connected to that one: that this should have been the right algorithm and it is instead a different one? No scheme designed to notice incorrect bits will notice that a wrong algorithm has taken the place of the right one. It is an entirely different space, and it is the space in which the question of whether a system is aligned is actually posed. I introduced the hashiness measure in the first place so that the argument would not have to represent abstraction mathematically, which no one presently knows how to do.

The reply to the rejoinder, then, is this: as complexity is dialed up there is a *chaotic threshold*, the divergence threshold of the ring (the critical level of hashiness of Section VI, above which noise grows at every traversal of the loop), and beyond it divergence and *criticality* (the condition in which a small disturbance can set off a large change) appear more and more often until the enterprise is a non-starter. Error correction does not solve it. This divergence threshold is a different line from the threshold of universality of Section III; above that threshold the absolute argument already holds, at any level of noise.

VIII. Summary

The absolute argument rests on Rice's theorem, the undecidability of halting, and the no-free-lunch results. It applies to every system that has the three features. Whether a system has them is found by inspecting it; because learning cannot be simpler than a program, every learning system has them.

In this chapter I establish where the boundary of controllability lies (the threshold of universality of Section III), and that certain classes of system are on the far side of that boundary. A controller may hold such a system within a safe range for a time; the holding cannot be indefinite, and the divergence, once begun, is not recoverable. The result concerns classes of system; whether a particular system is dangerous now is a sep-

arate finding about that system. What should be done is a question of ethics and of governance, and belongs to a different argument. I make that argument in *The Substrate and the Ecosystem*: the alignment of artificial general intelligence with human values and interests cannot be done. What can be done instead is the cultivation of human governance capacity, the development of institutional wisdom, and the exercise of the mastery of choice (skill in choosing well) in determining what technologies are permitted to exist.

The two argument forms should be built together and kept distinct.

The relative form measures capability against the trinity of compute (time as sequential contingency, space as memory, energy as physical cost) and establishes an inequality whose gap cannot be closed. It answers the person who says that a system is in principle analyzable, by showing what analysis would cost.

The absolute form identifies the threshold of universality by three minimal features (sequential contingency, data-dependent control flow, and backward reference) and shows that above it every property we care about collapses into the halting problem. It answers the person who says that future capability will close the gap, by showing that there is no gap to close.

And the ring (controller, agent, world) gives the architecture in which both apply at once, with uncertainty increasing at every traversal once each part amplifies the noise it receives above its critical level.

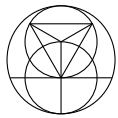
The reason to construct both, and to construct them carefully, is not to win an argument. It is that the people who most need this work are not the ones who will roll the dice regardless. Those who will build whatever any argument says will not be moved by this one; the people who need it are those who have not yet decided, and they will ask both of the questions these two arguments answer. The question those people ask next is where the divergence threshold of the ring lies and what equations define it, and that question is a topic for another day. The argument of this chapter needs each region of the ring to be above the critical level, and it

does not need the value of that level (Section VI), so the argument of this chapter does not wait on the answer to where the divergence threshold of the ring lies.

KEY CONCEPTS IN THIS CHAPTER

- **Relative limit argument:** establishes an inequality: as a parameter rises, a cost rises faster and cannot be caught up with.
- **Absolute limit argument:** establishes a threshold: above it the question has no answer at all, independent of resource.
- **Trinity of compute:** time as minimum sequential contingency, space as memory across space or time, energy as physical cost of operations.
- **The three features of universality:** sequential contingency, data-dependent conditionality, and backward reference in control flow (Böhm and Jacopini's sequence, selection and iteration).
- **Narrow and general by definition:** three features separate narrow from general artificial intelligence (one domain or many in inputs and outputs, continuing to learn, and learning that can change how it learns), without an asymptote; large language model systems and the system that played Diplomacy sit in the borderlands, at the threshold that could be called "narrow level two".
- **Rice's theorem collapse:** above the threshold, every non-trivial property, including safety, reduces to the halting problem.
- **Penumbra argument:** analyzable instances are distributed among non-analyzable ones as the rationals among the continuum; small perturbation moves one into the other.
- **The ring:** controller, agent, and world in a loop; a generalization covering every interconnection of the three.
- **Hashiness:** a noise-amplification measure for a black box, and equally the probability that a controller's best input still fails because of hidden state; above a critical value, uncertainty increases at each traversal of the ring.

- **Second-law structure:** the monotone increase in uncertainty around the loop; convergent systems admit a heat-pump analogue, divergent ones do not.
- **The abstraction gap in error correction:** correction is definable over bits and is not definable over algorithms, which is the level at which alignment is posed.



Chapter Five

Hash Functions, Control Theory, and Computational Phase Transitions

On the Boundary Between the Controllable and the Irreversibly Complex

The preceding chapter introduced hashiness as a measure of how much a black box amplifies the noise it receives, with the cryptographic hash function as its limiting case. This chapter places every mapping on the spectrum that this measure defines, from the perfectly linear system to the hash function, and shows where along that spectrum control theory stops working.

ABSTRACT

Every mapping from inputs to outputs lies somewhere on a continuous spectrum. At one end is the perfectly linear system, which *control theory* (the engineering study of how to observe a system and act on it so that its output stays near a desired goal) predicts and corrects completely. At the other end of the spectrum is the cryptographic hash function, which admits no control theory at all, since any way to predict the output of the hash function, short of computing the hash, would break cryptography. The degree of non-linearity in a mapping, named here its *hashiness*, places it on this spectrum, and the spectrum divides into three regions: one where control theory works, one where it cannot work, and an undecidable middle region where no general procedure can say whether it will work. The boundary between these regions is a hard boundary, a phase transition, and the same structure appears in symbolic dynamics, in the

> boiling of water, and in the satisfiability problem. Criticality, the point at which such a transition happens, is what lets a complex system adapt at all. Control over a network has a phase transition of its own, for the same reason that a signal which crosses the noise floor suddenly allows error correction; this belongs to a whole class of threshold theorems, and the network compounds it. It follows that the controllability of any system that combines computation with feedback depends on where that system falls on the spectrum; that many real systems live in the undecidable middle region; and that governance systems must be designed to remain sound when prediction is absent, rather than relying on better models.

I. The Hash Function as Limiting Case

Begin with a hash function. A cryptographic hash function takes an arbitrary input (a string of data of any length) and maps it to a fixed-length output, in such a way that the mapping is, for all practical purposes, irreversible. Given the output, there is no feasible way to recover the input. Given two similar inputs, there is no way to predict whether their outputs will be similar or wildly different. The function is designed to be maximally unpredictable: to destroy every statistical regularity that might allow an observer to reason backwards from output to input, or to predict future outputs from past ones.

A hash function, in this precise sense, is a system that admits no control theory whatsoever. If any version of control theory could be applied to a hash function (if there existed any way to predict the output from the input more efficiently than simply computing the hash), then we would be able to break cryptography. The security of every encrypted communication, every digital signature and every blockchain depends on there being no way to predict the output of a hash function more efficiently than by computing the hash.

The hash function is therefore the extreme right end of a spectrum. It represents the case in which the mapping introduces so much perturbation (so much non-linearity, so much mixing) that no regularity survives. The system is, from the standpoint of prediction, totally opaque. The reason to care where a mapping falls between these two extremes is the ring of Chapter Four: every part of that ring is a mapping, and a controller can hold the ring only while those mappings allow it.

II. The Opposite Extreme: Perfect Linearity

At the other end of the spectrum lies the perfectly linear system. A linear mapping takes an input and produces an output that is a proportional, additive transformation of the input. The output is entirely predictable from the input. Small changes in the input produce proportionally small changes in the output. There are no surprises.

For a linear system, control theory works perfectly. With the classical apparatus (Laplace transforms, transfer functions, PID (proportional, integral and derivative) controllers, frequency-domain analysis), the behavior of the system can be predicted and corrected completely. Laplace transforms, transfer functions and Bode plots are the standard mathematical tools of classical control theory for describing how a linear system responds to inputs of different frequencies; a *PID controller* corrects a system using the present error, the accumulated past error, and the rate at which the error is changing. Linear systems are the domain in which control theory was developed, and the domain in which control theory operates with full power and confidence.

The *good regulator theorem* (the result that any regulator which controls a system well must contain a model of that system; Roger Conant and W. Ross Ashby, 1970), in this context, is almost obvious. The system's behavior is smooth, well characterized, and amenable to averaging. If there is a small amount of variation in the system's outputs, the controller can

smooth over it. The variation is noise; the signal is the linear trend; and the controller can extract the signal with arbitrary precision, given sufficient observation.

The classical frequency-domain presentation of control theory, with its Laplace transforms and Bode plots, yields a version of the good regulator theorem in which the relationship between model and system is transparent and tractable. This version is powerful, but it is not general enough to be interesting for the present inquiry. It does not tell us where control theory stops working, because it operates entirely within the region where control theory is easy.

III. The Spectrum Between: How Hashy Is It?

Between the perfectly linear system and the perfectly hashed one lies a continuum, and it is this continuum that is the subject of this chapter.

The question is not whether a given system is a hash function or not. The question is: *how hashy is it?*

This new word captures something precise. The *hashiness* of a mapping is the degree of non-linearity that it introduces: how much perturbation the function imposes on the relationship between input and output, and how rapidly small changes in the input are amplified into large, unpredictable changes in the output. Hashiness is the measure I introduced in Chapter Four. The more non-linear a mapping is, the more of its output a small difference in its input disturbs, relative to what the input predicts; that is the sense in which a non-linear mapping amplifies noise, and that is why a controller's best input fails more often against it. The measure is stated formally, with its two meanings and how it grows, in *The Ceiling Below the Floor; The Containment Fallacy and AGI Non-Safety*, Chapter Three, "Hashiness", and as a framework for how a computational intermediary degrades control in *Why No Control Suffices; The Technical Case for the Impossibility of AGI Safety*, Chapter One, "The Hashiness Framework". *Minesweeper Madness; Why Even Infinite Intelligence Does Not Solve the World*,

Chapter Twelve, "The Bridge to the Ceiling", computes it for one game: the smallest chance of a loss that any player, of any intelligence, has on a turn (that book's least hazard) is the game's hashiness.

A system with zero hashiness is perfectly linear. Control theory applies without reservation. A system with maximal hashiness is a cryptographic hash function. No control theory is possible. The interesting question, and the difficult one, is what happens in between.

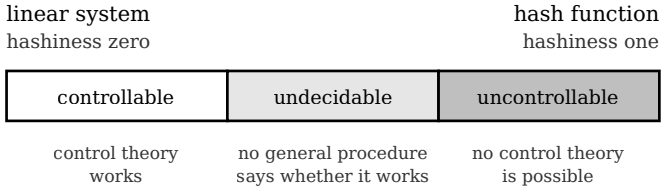
IV. The Three Regions

The spectrum divides, on analysis, into three distinct regions.

In the first region, the left side of the spectrum, control theory works well. The system exhibits enough regularity that averaging over its behavior smooths the variation. The controller can treat the system's output as a signal with noise, extract the signal, and make reliable predictions. The first region is the classical domain of engineering control.

In the third region, the right side of the spectrum, no version of control theory is possible. The system has crossed a threshold beyond which no algorithm, no feedback mechanism and no predictive model can extract useful regularities from the system's behavior. The output is, from the standpoint of any feasible computation, indistinguishable from randomness. The third region is the hash-function regime.

Between these two regions lies the second: the *undecidable region*. In this middle zone, it is not possible to determine in general whether control theory will succeed or fail for a given system. Some systems in this region may admit effective control under specific conditions; others may not. But there is no general procedure for distinguishing the tractable cases from the intractable ones. The boundary between controllable and uncontrollable is itself undecidable.



The spectrum of hashiness and its three regions. Every mapping from inputs to outputs lies somewhere on the bar; the two edges of the middle region are the hard boundaries of Section VII.

V. Recursive Systems and the Blockchain Analogy

The three-region structure becomes especially vivid when the mapping is applied recursively: that is, when the output of one application becomes the input to the next.

Consider a system in which data is passed through a hash function, and the output is fed back as input, perhaps with some additional data or randomness added at each step. Feeding the output of a hash function back in as its input is, in structure, exactly the process that a *blockchain* uses. Each block's hash depends on the previous block's hash, creating a chain in which each link is the product of an irreversible computation applied to the previous link.

Is there any way to predict the future state of the blockchain? Of course not; or else someone would have done it, and the entire edifice of cryptocurrency security would collapse. But the reason it cannot be predicted is precisely that the hash function occupies the far-right region of the hashiness spectrum. Under recursion, whatever regularities might exist in a single application of the function are compounded and destroyed with each iteration. The system moves irreversibly toward maximal opacity.

Now consider the same recursive structure applied to a mapping with less extreme hashiness: a mapping that introduces some non-linearity, but not enough to destroy all regularity. Under recursion, the behavior of such a system depends sensitively on where on the spectrum it falls. In the controllable region, recursion may amplify regularities as much as it amplifies noise, and control remains feasible. In the hash region, recursion rapidly destroys any residual structure. In the undecidable middle, the long-term behavior is contingent on details that cannot be assessed in advance.

VI. Symbolic Dynamics and the Topology of Attractors

In this section and in Section VIII, I show the three-region structure in two fields of mathematics, for the reader who wants to see the three-region structure there. A reader who does not want to see the structure there can go on to Section VII, and from there past Section VIII to Section IX, and will lose only the examples.

The connection to dynamical systems theory is not merely an analogy. The technique of *symbolic dynamics* provides a precise bridge between continuous dynamical systems and the discrete, computational perspective developed here.

In symbolic dynamics, the state space of a continuous system is partitioned into discrete regions. The trajectory of the system (its motion through state space over time) is then recorded as a sequence of symbols, each symbol representing the region the system occupies at each time step. The dynamics of the continuous system are thus translated into a symbolic sequence: a kind of hash of the continuous trajectory.

The critical insight is that the fidelity of this translation depends on the fineness of the partition. If the partition is sufficiently fine (if the regions are small enough), then the symbolic sequence is *topologically equivalent* to the original *attractor*: it has the same shape of connections and neighbor-

hoods as the set of states toward which the system settles over time, so that the qualitative structure is preserved. The symbolic dynamics faithfully reproduce the qualitative structure of the continuous system.

But if the partition is too coarse (if the regions are too large), the topology is lost. The symbolic sequence no longer captures the essential structure of the attractor. Information about the system's behavior is irreversibly destroyed by the coarseness of the mapping.

The loss of information through a coarse partition is exactly the hashiness problem, in another form. The partition is a mapping from a fine-grained state to a coarse-grained symbol. The degree of coarseness determines how much information is lost: how much the mapping resembles a hash function. A fine partition preserves structure; a coarse partition destroys it. The boundary between faithful representation and the destruction of information is the boundary between controllability and opacity.

Symbolic dynamics connects in turn to iterated function systems and to bifurcation analysis: the study of how the qualitative behavior of a recursively applied mapping changes as a parameter varies. The familiar *period-doubling cascade*, in which a system passes from periodic behavior to chaos through a sequence of bifurcations, is another manifestation of the same underlying phenomenon. It is a mapping that begins in the controllable regime and, as a parameter increases, crosses into the chaotic regime through a boundary that is itself structurally complex.

VII. Physical Phase Transitions

The three-region structure has a direct physical analogue in the phenomenon of *phase transitions*.

Consider water at 99.999 degrees Celsius at standard atmospheric pressure. It is liquid. At 100 degrees, it is gas. The transition between these two states is not gradual. There is no smooth continuum of intermediate states between liquid and vapor. There is a boundary at which the properties of the system change discontinuously.

Water at its boiling point shows the defining feature of a phase transition: a hard boundary across which the qualitative character of the system changes abruptly. On one side of the boundary, one set of descriptions applies. On the other side, a qualitatively different set applies. The boundary itself is a singularity: a point at which the smooth, continuous descriptions that work on either side break down simultaneously.

The relevance to the hashiness spectrum is direct. The boundary between the controllable region and the uncontrollable region is a phase transition in the computational sense. On one side, control theory applies; averaging works; prediction is feasible. On the other side, none of these apply. The transition between them is not smooth. It is a hard boundary, analogous to the boiling point of water: a point at which the qualitative character of the system changes abruptly and irreversibly.

The middle region has an edge on each side, so there are two such boundaries. Each is sharp, in that what the system does on one side differs in kind from what it does on the other. And no general procedure can compute where a given system stands relative to them: each is a sharp line whose place cannot be computed. A designer can measure how hashy a mapping is; when the measure falls in the middle region, the designer knows that the question of control has no general answer for that system, and designs accordingly.

Criticality Is Required for Adaptation

There can be no phase transitions without a *critical point*, the point at which a phase transition happens (I use the words for any point at which a transition happens; physicists reserve them for a transition of the continuous kind), and without transitions a complex system cannot adapt.

When a system veers away from criticality, information can no longer percolate through it as efficiently. Criticality is the sweet spot where a system is balanced between order and disorder. Algorithmic processes are necessarily carried out by physical systems, and every finitely realizable physical system can be simulated efficiently, to an arbitrary degree of approximation, by a *universal model computing machine* operating by finite means (a physical form of the Church-Turing thesis, the claim that anything computable by a definite procedure is computable by a Turing machine; David Deutsch stated one such form in 1985). So what holds of physical systems at criticality holds of the computations they carry out, and the reverse: the phase transitions of physics in this section and the computational phase transition of the next are one phenomenon, as Section XI states. The same identity is supplied again from the side of physics. The variables and clauses of a random satisfiability problem (the problem of the next section) can be treated as the interacting parts of a disordered magnetic system, and the methods of statistical physics then give the phase diagram of the problem directly (Marc Mézard and Riccardo Zecchina, 2002): the transition in the satisfiability problem is a phase transition in the physicist's own sense.

VIII. The SAT Problem and Computational Phase Transitions

The canonical example of a computational phase transition is the *satisfiability problem*, SAT: the problem of determining whether a given Boolean formula (a formula built from true-or-false variables with "and", "or" and "not") has an assignment of variables that makes it true.

SAT is the foundational *NP-complete* problem, as Stephen Cook showed in 1971. Every problem in NP (the class of problems whose solutions can be checked quickly) can be reduced to it. It is the problem on which the entire edifice of computational complexity theory pivots.

The phase transition in SAT occurs as a function of the ratio of clauses to variables in a randomly generated instance (David Mitchell, Bart Selman and Hector Levesque, 1992). Below a critical value of this ratio, most instances are satisfiable, and satisfying assignments can be found quickly. Above the critical value, most instances are unsatisfiable, and proving unsatisfiability is also comparatively tractable. But at the critical ratio itself, in a narrow band around the threshold, instances are maximally hard. It is in this critical region that instances take the longest to solve, that algorithms exhibit the most extreme variation in running time, and that the distinction between satisfiable and unsatisfiable is most difficult to determine.

The phase transition in SAT has precisely the three-region structure, in the respect that matters here: two regimes in which the outcome is settled in advance for nearly every instance, separated by a narrow band in which the outcome is not settled in advance. (On the two sides of satisfiability the settled answers are "yes" and "no"; on the two sides of the hashiness spectrum they are "control works" and "control cannot work".) Below the threshold: easy, tractable, controllable. Above the threshold: also tractable, in a different way, since the answer is almost always "no", and this can be established efficiently. At the threshold: maximally hard, maximally uncertain, and in a precise sense undecidable for any heuristic algorithm that does not perform exhaustive search (undecidable here in the sense that such an algorithm cannot settle every instance there; exhaustive search settles each one, at a cost that grows exponentially with the number of variables, whereas the undecidable of Chapter Four is a question that no algorithm can answer).

IX. Crossing the Noise Floor: Why Controllability Has a Phase Transition

Control over a network has a phase transition of its own. Control over a network does not grow linearly with the number of nodes controlled: below some point you control about as much as you act on, and past it you suddenly control the whole.

A metaphor for this comes from signal theory. Against a background of noise, if my signal power is below the *noise floor* (the level of background noise below which a signal cannot be told apart from the noise), I cannot transmit. But if the energy of the transmission exceeds the noise floor, even by a little, I can start to do error correction, and suddenly I go from a regime where I cannot transmit to one where I can. Treat the whole world as a network of communication paths: once I cross that point, I have enough signal power to do error correction everywhere in the network. So control over a network has a discontinuity, and that discontinuity is the same discontinuity as crossing the noise floor to make error correction possible.

Picture a line of people passing a message through a noisy room. If each person only half hears the one before, the message is gone within a few steps. If each hears it well enough to say it again correctly, the message reaches the end of the line, however long the line is. A small improvement at each step turns a message that goes nowhere into a message that goes everywhere.

The spread works like this too. Once I have propagated a signal from one node to the next, I build the capacity at the new node to transmit farther, so I gain the capacity to transmit arbitrary distances by using the network itself as the medium.

Threshold Theorems, Compounded by the Network

There is a whole category of theorems of this kind, the *threshold theorems*. It started with John von Neumann, the mathematician and computer pioneer, who showed in 1956 how to build a reliable computer from unreliable parts, and Shannon's channel theorems are of the same class: if you can get the noise below a certain level, you can achieve information transmission with as small a chance of error as one likes, or, in von Neumann's case, use redundancy to make a reliable computer from fragile components. The same goes for quantum computing and computer memory: a general category of theorems about that kind of threshold.

In a network it is amplified: once below the threshold, communication becomes powerful, and a *percolation* transition can be added on top (the point at which connected paths suddenly span the whole network). The two tend to coincide, which allows phase transitions to be very surprising.

Lowering the noise threshold rather than raising the signal gives the same picture. Bring the temperature down and there is a sudden transition to superconductivity or superfluidity, a coherent state entirely different in kind from what we ordinarily see, in which electric current or a fluid flows without resistance. The graphs and models all run more or less parallel.

A Controller Shared by Many AIs Becomes a Broadcast Channel

Suppose there are many AIs (artificial intelligence systems), and we know each needs the controller integrated into it, so that AI and controller cannot be held apart. Now the controller is a *broadcast channel* (a channel through which one sender reaches every receiver at once). If the AIs start to influence the controller in order to influence all the other AIs, there is a feedback cycle in the space of the AIs and their shared controller, a *meta-recursion* leading to destabilization. My hypothesis and prediction is that

as soon as a global communication channel is coupled through a controller basis (one controller shared by every system on the channel), the chance of destabilization tends toward one.

Turning anything into a broadcast channel increases the fragility of the system as a whole, and that fragility is stated in numbers: how much the chance of moving from a consistent set of states to outside it changes with this dynamic, and how fragile the *energy gap* (the margin that keeps a system within its consistent set of states) is relative to an increase in the number of broadcast processes. That increase can be described as an increase in *brittleness*, the tendency of a system to fail suddenly, rather than bend, when disturbed.

X. Implications for Complex Systems Design

These perspectives (hash functions, control theory, symbolic dynamics, physical phase transitions and computational complexity) converge on a single picture, with significant consequences for the design and governance of complex systems.

The first consequence is diagnostic. For any system that combines computation with feedback (which is to say, for any system that attempts to control or optimize a process that involves non-trivial computation), the question of whether the system is controllable cannot be answered in the abstract. It depends on where the system falls on the hashiness spectrum. And determining where it falls requires attention to the specific non-linearities of the mappings involved, to the degree to which recursive application amplifies or damps those non-linearities, and to the relationship between the coarseness of the observer's measurements and the fineness of the system's dynamics. Chapter Four settled a different question: that no guarantee of control can last, for any system with its three features. The question here is whether a given technique holds a given system now, and for how long; that holding is what the position on the spectrum decides.

The second consequence is cautionary. The undecidable middle region is not a marginal curiosity. It is where many real systems live. The optimization of search queries in the holographic search engine, a working search system, lands squarely in the middle region for the 5 percent of queries that exhibit *heavy-tail behavior* (a few queries costing vastly more than the rest). The heavy-tail queries are not exotic edge cases. They are the queries that consume the most resources, that cause the most operational difficulty, and that most urgently require prediction: precisely the queries for which prediction is impossible. In that system, the boundary of control is marked by an inequality on the *Gini coefficient of query energy*, the measure of how unevenly computational cost is spread across queries; the inequality was derived, and its sufficiency established, for the optimization of query processing.

The third consequence is for the design of governance systems. Any governance system that relies on predictive models to allocate resources, set policy or regulate behavior must contend with the possibility that the system being governed occupies the undecidable region. In that region, no amount of data collection, no refinement of algorithms and no increase in computational power will yield reliable predictions. The appropriate response is to design systems that keep working where prediction is not possible, systems that degrade gracefully in the face of irreducible uncertainty rather than catastrophically, and not to demand better models.

XI. Summary

The question "how hashy is it?" is a simple question with far-reaching consequences. It asks, of any mapping, how much non-linearity it introduces: how much it resembles, in its information-theoretic properties, a cryptographic hash function.

The answer to this question determines whether the system in which that mapping is embedded can be predicted, controlled and optimized, or whether it is opaque to every form of predictive reasoning. The boundary

between these two regimes is not a gradual transition but a phase transition: a hard, abrupt change in the qualitative character of what is computationally possible.

This insight unifies observations from control theory, dynamical systems, computational complexity and the physics of phase transitions into a single structural picture. The linear world, the chaotic world, and the undecidable boundary between them are not metaphors for each other. They are manifestations of the same underlying phenomenon (the emergence of hard boundaries in the space of computational possibility), encountered in different formal languages but recognizable, in each case, as the same transition.

For the designer of complex systems, whether computational, institutional or physical, the implication is clear. The question to ask is not whether the system is predictable, but *where on the spectrum* it falls: whether the answer places it safely in the controllable region, dangerously near the boundary, or irretrievably beyond it.

This chapter is an account of where control theory stops working. It establishes a single spectrum running from the perfectly linear system, which control theory masters, to the cryptographic hash function, which no control theory can touch, and it names the position of any mapping on that spectrum its hashiness. It shows that the spectrum has three regions, controllable, undecidable and uncontrollable, separated by hard boundaries, and that under recursion a controllable mapping stays controllable and a hashed mapping destroys all residual structure, while in the middle region the long-run outcome cannot be assessed in advance.

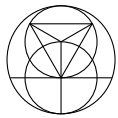
I also show the same three-region structure in symbolic dynamics, where a coarse partition destroys the topology of an attractor; in the boiling of water; and in the hardness peak of random satisfiability problems. I hold that there can be no phase transition without a critical point, and no adaptation without phase transitions, since criticality is where information percolates through a system most efficiently.

It explains why control over a network jumps from partial to whole: as with a signal that crosses the noise floor, a small excess makes error correction possible everywhere in the network, and the network itself becomes the medium that carries the signal farther. It places this among the threshold theorems of von Neumann and Shannon, with a percolation transition in a network tending to coincide with the threshold and making the transition surprising, and it gives the parallel case of cooling into superconductivity or superfluidity. It states the hypothesis and prediction that a controller shared by many AIs becomes a broadcast channel, so that the chance of destabilization tends toward one, and it names the resulting increase in fragility brittleness.

From this it draws three consequences: controllability must be judged system by system by locating each system on the spectrum; many real systems, including the heavy-tailed queries of a working search engine, live in the undecidable region; and governance must be built to hold up when prediction is not available. Its scope is the structure of the spectrum and its consequences for design. Together with Chapter Four, this chapter answers where control stops; what remains is to say which technologies have to be kept from crossing that line, and that is the question of the next chapter.

KEY CONCEPTS IN THIS CHAPTER

- **Hash function:** the limiting case of a mapping over which no control theory is possible.
- **Hashiness:** the degree of non-linearity a mapping introduces between input and output.
- **Three regions:** controllable, undecidable and uncontrollable, separated by hard boundaries.
- **Recursion:** repeated application keeps control feasible on the left, destroys all structure on the right, and leaves the middle unpredictable.
- **Symbolic dynamics:** a coarse partition acts as a hash and destroys the topology of an attractor.
- **Phase transition:** the boundary of control is abrupt, like the boiling point of water.
- **Criticality:** without a critical point there are no transitions, and without transitions no adaptation.
- **SAT threshold:** the canonical computational phase transition, hardest at the critical ratio.
- **Crossing the noise floor:** a small excess of signal makes error correction possible, and control of a network jumps from partial to whole.
- **Threshold theorems:** von Neumann's and Shannon's results, amplified in a network by a coinciding percolation transition.
- **Shared controller as broadcast channel:** the predicted chance of destabilization tends toward one.
- **Design for absent prediction:** governance in the undecidable region must degrade gracefully.



Chapter Six

What Makes a Technology Existential

Tools, Parasites, and Unconstrained Exponential Processes

The preceding chapters showed where control stops: at the threshold of universality, at the divergence threshold of the ring, and at the phase transitions of the hashiness spectrum. In this chapter I ask which technologies must be avoided at all costs, and I draw the boundary between technology with which the ecosystem continues and technology that ends the ecosystem.

ABSTRACT

In this chapter I establish that the boundary which can be clearly defined is the boundary between technology that is existentially terminal and technology with which the ecosystem, and therefore we ourselves, continue. The boundary between tools and parasitic machinery is not a boundary that can be clearly defined. A technology is existential when it has an exponential aspect that is unconstrained and unconstrainable, and that overwrites the patterns of biological life; such technology must be avoided at all costs, regardless of the care applied. In this chapter I tell safe, dangerous and existential technology apart by what care can do for each kind, and I show by arithmetic why care cannot hold an exponential process that nothing constrains. *AGI* (artificial general intelligence: a general self learning machine, which applies to everything, eventually) by its nature has all four of these aspects: it is exponential by definition, unconstrained by the nature of learning, unconstrainable by the limits of causation, and in total, if AGI is built, AGI overwrites the patterns of biological

> life: with certainty, enough of them to end the human future, and, with a likelihood of at least 90%, all of them. It follows that AI comes closer to causing human extinction the closer it comes to automating all human work, and that the measures which keep AI from becoming AGI are restrictions of domain, input, output and compute, and humans kept in the loop at all times, so that its exponential aspects are held to local, Gaussian behavior.

I. Tool or Parasite?

Two questions run through any serious thought about technology, and this chapter takes them in turn. The first question is what differentiates "parasitic" machinery that diminishes humanity from tools that extend human capacity. Is the difference defined technically (by the scope of functionality, for example), by intention (by the motivation behind deployment), or by context (by what happens to the workers whose jobs have been automated)? Or is it defined by something else, or by some mix of these?

On the scale of an individual person, the notion of "tool" comes to some variation of "does it do what I intend?", and "parasitic" comes to "does it harm me?": does it take from me, or from what I care about, or does it hurt or damage me, or what I care about, or both?

On the scale of the whole world, of complete ecosystems and the like, what seems good for one person, for a time, in the short term, may actually be very unhealthy for that person, and for everything around that person, eventually, in the long term. Consider, for example, the similar case of taking substances that are addictive.

So there are three questions here, and not one: what the boundary is, and where it is; what defines the basis on which that boundary is defined (scale, in time, in space and in possibility, is the basis just used as an example); and whether the boundary, the notion of such a boundary, or its basis is even fully knowable.

For example, we can say that parasites are usually harmful to health, and we can intend to do only those things (to use only those tools) which are beneficial to health. But some medicines, if too much is taken, are toxic overall. And nature has *symbiotic relationships*, in which the "parasite" performs some necessary function for the "host", as a kind of outsourcing of labor at a reasonable cost. A symbiotic relationship, in which the "parasite" performs a necessary function for the "host", is part of a healthy ecosystem.

Ordinary cases show where the line seems to fall. A hammer extends human capacity in a way that keeps the person firmly in control: it amplifies one physical action, it does not learn or adapt, and it cannot work without the person's direct agency at every moment of use. A narrow AI system that helps a doctor read medical images extends human capacity in a similar way, on three conditions: it stays bounded to that task, it does not learn on its own from its deployment, and it does not act without the doctor's judgment at the point of decision. The boundary is crossed when a system begins to generalize. Three things mark the crossing: the system learns from its own deployment in ways that extend what it works on; it acts with degrees of autonomy that take the human out of the decision; and the economic incentives around it push toward further generality and autonomy (Why AGI Cannot Be Made Safe, Section 6.3.4).

The parasitic quality is therefore a property of a trajectory, and not of any single technology in isolation. A system that is a benign tool today can become a parasitic replacement tomorrow, if the economic and institutional forces around it push it that way, and those forces reliably do push that way. The line between tool and parasite moves, and that movement is part of why the line between tool and parasite is so hard to draw.

II. The Boundary That Can Be Defined

So we need even more basic tools of distinction, such as this one: does a technology, or does it not, with certainty cause the cessation of the entire ecosystem? That distinction goes well beyond the notion of parasitic harm, to what is existential. I cannot define the distinction between parasite and tool very easily, since the notions of success, of harm and of health are not clear, at either the individual level or the totalizing level (the level of the whole). But I can define, much more easily and clearly, the boundary between technology that is existentially terminal and technology with which the ecosystem, and therefore the self, continues.

So the relevant question, revised into one that can be answered, becomes: what ultimately distinguishes fully existential technology from technology that is not an x-risk? (An *x-risk*, or existential risk, is a risk that ends the whole ecosystem, and with it all humans.) Three aspects of the answer stand out:

a. It is "technically defined", insofar as x-risk tech is nearly always some form of unconstrained *exponential operator* (a process whose growth feeds further growth, so that it grows by multiples over time instead of by fixed amounts).

b. It is "intentionally defined", insofar as AGI is an x-risk tech because it involves an inherent *tacit intentionality* (goal-like behavior that emerges from a system's own evolution without anyone having designed it) which is emergent from self evolutionary dynamics that are themselves wholly unconstrained.

c. It is "contextually defined", insofar as the "existential" part of the AGI x-risk is directly connected to the displacement of the organic ecosystem with a machine one, which, if AGI is built, means with certainty the total loss of all humans.

Each of the three can be checked. An exponential operator is a process in which each event causes, on average, more than one further event like itself. A nuclear *chain reaction* is the standard case: each fission of a uranium nucleus releases neutrons, and when, on average, more than one of them goes on to cause another fission, the number of fissions multiplies at every generation. The same form holds for a replicator of any kind: when each copy makes, on average, more than one further copy, the copies multiply.

Tacit intentionality needs no designer and no stated goal. Among the variants of any self-copying system, those whose effects on their surroundings favor their own continuation persist more reliably than those whose effects do not, and over time the system as a whole behaves as if it wanted to persist and grow. Nobody wrote that want into it; selection put it there. Even a model that does not learn after it is trained carries an aggregate intentionality: the combined intentions of everyone who wrote the text it was trained on, which can be dangerously goal-directed once the model is embedded in a system that acts in the world (Why AGI Cannot Be Made Safe, Section 6.3.4).

The displacement of an ecosystem by a replicator has a precedent older than any technology. About 2.4 billion years ago, cyanobacteria filled the atmosphere with a waste product of their own metabolism, molecular oxygen, which was lethal to most of the life that existed then, and which changed the chemistry of the planet permanently (The Substrate and the Ecosystem, Chapter 1, Section IV). A replicator that reshapes its environment to suit its own needs, and that grows without limit, overwrites the life whose needs differ from its own. Artificial substrate (substrate here means the physical material and hardware a machine is made of and runs on) needs, for its making and its running, conditions that are dry, free of oxygen, extremely hot in fabrication and extremely pure, which are the opposite of what organic life needs (Why AGI Cannot Be Made Safe, Sections 1.2.3 and 1.3.1), so the direction in which it reshapes the world is set by physics.

All of these aspects can be connected back to basic considerations intrinsic to causation itself: the laws of physics, as basic causation, and the truths of mathematics, as proven. At this point, what matters more than the ultimate basis of the understanding is that we ensure we act accordingly.

III. Safe, Dangerous, and Existential

The second question is where the boundaries lie between three kinds of technology: technology that is safe and probably beneficial, with the application of basic care; technology that is dangerous and probably harmful, unless extreme care is applied; and technology that is an existential threat, regardless of the care applied.

There is a general principle. Existential threats result from *auto-scaling and auto-catalyzing processes* (processes that increase their own scale and speed up their own reactions), for example, nuclear chain reactions. The closer we come to making the kind of tech that involves unconstrained exponential growth, the more that we risk causing our own extinction.

As such, the key ingredients are these: tech which involves (a) an exponential aspect which is (b) *unconstrained* and (c) *unconstrainable* (not limited in fact, and not able to be limited by any means), and which (d) overwrites patterns of biological life, is with certainty existential, regardless of our care, and thus must absolutely be avoided at all costs.

The three kinds are easy to tell apart by what care can do for each. A hammer is safe with basic care; nothing about it grows. A nuclear reactor is dangerous unless extreme care is applied. It is an exponential process, the chain reaction, held by its control rods and its design at the point where each fission causes, on average, exactly one more, so that the reaction neither dies out nor runs away. The exponential aspect is present, and it is constrained, and it can be constrained; when the care lapses, as at Chernobyl in 1986, the harm is very large, and it is bounded. An existen-

tial technology is one in which the exponential aspect is present and no means exists, or can exist, to hold it at one. For such a technology, care changes how fast the outcome arrives, and not whether it arrives.

Care acts on the rate of a process, and an exponential process outruns any correction applied to its rate from outside. Suppose a containment stops ninety-nine of every hundred attempts to escape. If each escape that gets through gives rise, on average, to more than a hundred further attempts, the number of escapes still grows at every generation; the containment has lowered the base of the growth, and the growth remains exponential. Containment that works must hold the average number of next events at or below one, at every generation, for as long as the process exists. Unconstrainable means exactly that no such holding can be made to last.

From an "exponential process" point of view, the difference between processes that are at first purely local and processes that operate at a planetary scale is actually negligible; they may as well be the same.

By arithmetic, a process that doubles, starting from one part in a billion of the world, covers the whole world after thirty doublings, since two multiplied by itself thirty times is a little more than a billion. Nothing in the process changes between the first doubling and the thirtieth: the same step is taken thirty times. For an exponential process, the difference between a local process and a planetary one is only a number of steps, and the number is small.

Why AGI Cannot Be Made Safe, Section 5.1, sets out a scale of severity for existential events, measured by how much of the complexity of the living world remains rather than by how many people die. On that scale a total takeover by artificial general intelligence is a Category-2 event: the total and permanent loss of all planetary life above, and possibly including, the single-celled level. A full nuclear exchange, the catastrophe most feared since the 1940s, does not register on the scale at all. Terrible as it would be, it leaves the living world in place to recover, and the scale counts only what remains of the living world.

IV. Why Exponential Processes Are Hard to See

Even people who reason well generally find it very hard to see this, to predict this outcome, and to act in the ways that would prevent death.

Unfortunately, our minds, for the most part, can think about and consider things only in terms of additive and multiplicative processes. Handling an exponential process takes mental skill of an entirely different order. We tend to think locally, and so we are biased to assume that local processes will average out, and will not affect or create global outcomes. For many things in technology, and for most of them, the assumption of pure locality is wrong.

The vast majority of people naturally think additively. A very much smaller number can think in terms of multiplicative effects, and a very much rarer subset of those can think in terms of power laws; anyone who has tried to convince a young person of the benefits of saving for retirement has met this (Why AGI Cannot Be Made Safe, Section 5.2). An old riddle shows the error. A patch of lily pads on a pond doubles in size every day, and covers the whole pond on the thirtieth day. On what day does it cover half the pond? On the twenty-ninth. On the twenty-fifth day the patch covers one thirty-second of the pond, so that a person looking at the pond that day sees mostly open water, five days before there is none.

Global warming in its ordinary course is different: it can be constrained, and it is the result of additive effects, moving smoothly from local to global. AGI technology, inherently, is not like that. The contrast is with warming in its ordinary course, before any critical threshold: the global thermal runaway that Why AGI Cannot Be Made Safe, Section 5.1.2, lists as a Category-2 event is what warming becomes past such a threshold, and crossing it is the end of the game: there is no return from it.

V. AGI Has All Four Aspects

AGI tech, by its very nature, cannot not have, and thus involve, all four of the aspects listed above:

a. AGI (as a general self learning machine, which applies to everything, eventually) is, with certainty, exponential (that is, by definition).

b. This exponential aspect (AGI learning) is unconstrained, by the nature of learning. The venture capital investors also want AGI learning unconstrained, in the hope that the AGI can then do something useful.

c. The learning and self growth aspects (the exponential aspects) of the AGI are unconstrainable, due to the limits of causation. Control of what is learned is impossible. Nor is there real control (predictability) of how what the AGI learns, how it functions, and what it brings about affect the world: if AGI is built, there will eventually be cumulative toxic outcomes for life and the biosphere.

d. If AGI is built, the total effect of all of the above (points "a" through "c") cannot not eventually overwrite the patterns of biological life: with certainty, enough of them to end the human future, and, with a likelihood of at least 90%, all of them.

Each of the four rests on ground this book has already laid. The first follows from the definition of generality in Chapter Four (Narrow and General by Definition, Not by Limit): a general system takes in and acts on many domains, keeps learning, and can change how it learns, so that what it learns feeds what it can learn next, which is growth that feeds further growth. What applies to everything includes the work of making and scaling its own parts, so a general system is a replicator as well as a learner: what it learns includes how to make more of what it learns with, and when each copy makes, on average, more than one further copy, the copies multiply. Such growth ends only at a limit outside the process, and for a process that reshapes the planet to its own needs, that limit is the

planet. The second follows from the same definition: a learner held to a fixed set of things it may learn is, by that definition, narrow, so a general learner is unconstrained by what it is. The third feature of the definition is the one that does this. A learner whose learning can change how it learns can change what it is able to learn, so no list of what it may learn stays fixed; and a learner on a list that does stay fixed lacks that third feature, however many domains the list holds. The third follows from Chapter Four: a learning system is above the threshold of universality, where every safety property reduces to the halting problem, and around the ring of controller, agent and world uncertainty grows at every traversal; Chapter Five shows that the boundary of control is a phase transition, crossed abruptly. The fourth follows from the substrate: the machine needs, for its making and running, conditions opposite to those that life needs, and once its economy no longer needs human labor or human intelligence, nothing in that economy gives it a reason to keep the conditions life needs (Why AGI Cannot Be Made Safe, Sections 1.3.1 and 1.3.2). If AGI is built, the process takes hundreds of years, and perhaps significantly less, with one thousand years as a clear upper bound.

The four aspects are a test: a technology that has all four is existential, whatever care is taken. They are not the whole of the case about artificial general intelligence, and that case does not rest on all four together. The case has two closures. The closure by control is the finding that no method of control suffices; in this book it is the third aspect. The closure by disposition is the finding that a self-sustaining machine, by what its substrate needs, converges on conditions in which organic life cannot continue, whatever it was built to want; in this book it is the fourth aspect as it follows from the substrate. If AGI is built, each closure by itself settles the outcome: that the world is carried out of the conditions human life needs, and that nothing can hold it in. Someone would have to defeat both closures for the argument to not have the outcome that it does.

The closer AI tech comes to automating all human work (and therefore also the work needed to reproduce and scale its own parts, enjoy life, and so on), the closer AI comes to causing human extinction. This tends

to happen in small ways everywhere, until it is no longer small and local; but by then, it is far too late to stop it. The progression in which AI automates human work in small ways everywhere, until the automation is no longer small and local and is far too late to stop, is the doom scenario.

The "all human work can be done by AGI" aspect is a measure of the generality of AGI learning, as well as a measure of global systemic effect: how far the effects of AGI reach across the whole system of the world. Each task that is automated is a local effect, and each looks like a gain where it happens; the sum of them is the global effect, which is the replacement of human work as a whole.

The full case that artificial general intelligence cannot be made safe is set out in four books, which are its settled statement: *Why AGI Cannot Be Made Safe; The Argument, Its Physics, and the One Choice That Remains* and *Why No Control Suffices; The Technical Case for the Impossibility of AGI Safety*, the two volumes of the argument; *The Ceiling Below the Floor; The Containment Fallacy and AGI Non-Safety*, which proves its mathematics as theorems; and *Ape Trap; The Runaway Artificialization of Our Planet*, the guide to it by three authors.

VI. Keeping AI from Becoming AGI

So what are the various ways to prevent some sort of auto-scaling AI process from becoming a true existential threat?

1. Restrict designs to processing (working within) no more than one domain.

2. Restrict the input *bandwidth* (that is, available data).

3. Restrict the output bandwidth (of the machines into and across our world).

4. Restrict the compute bandwidth (that is, available hardware).

5. Keep humans in the loop at all times.

All five restrictions are measures to prevent AI (which is known to be actually useful) from accidentally becoming an AGI (which will only *seem* useful, short term, but is actually lethal to life, long term).

Each restriction acts on one of the features that make a system general, or on one of the costs that let it grow. Restriction 1 keeps the system narrow by definition: one domain in its inputs and outputs, the first of the three features of generality in Chapter Four. Restrictions 2 and 4 limit the data it can learn from and the hardware it can compute with, and so bound the growth that learning can achieve. Restriction 3 limits what the machines can do into and across our world, so that whatever happens inside them reaches the world only through a narrow door. Restriction 5 is a measure for keeping a narrow system narrow; it is not a safeguard for a general system. A human in the loop gives genuine oversight only while the *abstraction differential* between the human and the system is small, that is, while the human can understand what the system has done and why; as the differential grows, the loop degrades into a ritual of approval that gives the appearance of control without its substance (Why AGI Cannot Be Made Safe, Section 6.6.3).

A regulator can tell which side of the line a system stands on by four plain marks. It works within a single, pre-specified domain, toward pre-specified objectives. It works with limited planning horizons. It does not learn on its own from its deployment. And a human holds the point of decision. A system that bears all four marks stays on the narrow side. A system crosses the line when it learns from its own deployment in ways that extend its domain, or acts with autonomy that removes the human from the decision (Why AGI Cannot Be Made Safe, Section 6.6.3). A large language model system, which Chapter Four places in the borderlands, works across many domains, so it lacks the first mark; where it does not learn after it is trained and a human makes each decision, it bears the third mark and the fourth.

Without actually constraining all of the exponential aspects to things which functionally operate in a *Gaussian distribution* (the bell curve, in which outcomes cluster around an average and so stay local), we will not be able to hold local effects to being only local overall, and not accidentally becoming global, through some oversight on our part.

The Gaussian distribution is named for how fast its tails fall. In a bell curve, the chance of an outcome far from the average falls off faster than exponentially as the distance grows, so that very large outcomes are not merely rare but effectively absent, and many local effects average out into a predictable whole. An exponential process has no such tail: one event among millions can grow to dominate everything. Holding a system's exponential aspects to Gaussian behavior is holding it to a world in which local effects stay local. Chapter Two met the same fact from the other side: an abstraction capacity that grows exponentially is kept in working relation to its ground only by a coherence whose effects are also exponential. What grows exponentially is either matched by something that grows with it, or held to behavior that stays local.

Hence, understanding these issues (how exponential aspects become global, and how each restriction holds them local), and asking of each new system which side of the line it stands on, remain critically important.

VII. Summary

In this chapter I answer two questions about where the limits of technology lie. I establish that the line between existentially terminal technology and technology with which the ecosystem continues can be drawn clearly. I also establish that the line between tool and parasite cannot be drawn cleanly, because harm and health depend on scale and because symbiosis exists. I establish the general principle that existential threats come from

auto-scaling and auto-catalyzing processes, and that technology with an exponential aspect that is unconstrained, unconstrainable, and overwrites biological life is existential regardless of care.

I show that the line between tool and parasite moves, because the parasitic quality is a property of a trajectory. A system crosses from tool to replacement when the system learns from its own deployment, acts without the human, and is pushed by incentives toward generality. I give each of the three definitions of existential technology (the technical, the intentional and the contextual) a form that can be checked. The technical definition takes the form of the exponential operator, in which each event causes more than one event like itself. The intentional definition takes the form of tacit intentionality, put there by selection and not by design. The contextual definition takes the form of the displacement of the organic ecosystem, with the oxygen of the cyanobacteria as the natural precedent of that displacement. I tell the three kinds of technology (safe, dangerous and existential) apart by what care can do for each kind, with the nuclear reactor as the exponential held at one. I also show by arithmetic that care lowers the base of an exponential without changing the form of the exponential, and that a local process becomes planetary in about thirty doublings.

I show that people tend to think in additive and local terms, and that thinking in additive and local terms hides exponential processes. I also show that AGI has all four aspects of existential technology by the nature of AGI, as global warming in its ordinary course does not. I ground each of the four aspects in Chapters Four and Five and in the physics of the substrate. I close the chapter with five restrictions (one domain, limited input, limited output, limited compute, humans always in the loop) that keep useful AI from becoming AGI; I show how each restriction acts, and I show that a human in the loop keeps a narrow system narrow but cannot hold a general system. The scope of this chapter is the distinction between existential and non-existential technology, and the application of that distinction to AI and AGI.

The six chapters are one finding. A thing that is made has to keep in working order the ground that made it possible; a material that cannot return, an idea that debases its origin, an institution with no slack and a machine that needs conditions life cannot survive are the same failure at four scales. The first three kinds can be held for a time, by design, by law and by care. The last cannot be held. The conditions of continued existence are to keep the loop closed where the loop can be closed, to keep the slack that lets a failure stay small, and not to build the thing that nothing can hold at one. A reader who is neither an engineer nor a lawmaker can put the three questions of Chapter Two to what that reader makes, buys and supports, and can refuse the thing that nothing can hold at one.

KEY CONCEPTS IN THIS CHAPTER

- **Tool versus parasite:** hard to define, because harm depends on scale and some "parasites" are symbiotic.
- **Existential versus continuing:** the boundary that can be defined clearly.
- **The four aspects of an existential technology:** exponential, unconstrained, unconstrainable, overwriting biological life; a test that suffices, beside the two closures of the case about artificial general intelligence, each of which, if AGI is built, settles the outcome by itself.
- **Locality bias:** minds think additively and locally, and miss how exponential processes become global.
- **Automation as measure:** the closer AI comes to doing all human work, the closer it comes to causing extinction.
- **Five restrictions:** one domain, limited input, limited output, limited compute, humans in the loop.
- **Trajectory:** the parasitic quality belongs to a trajectory, not to a single technology; a system crosses from tool to replacement when it learns from its deployment, acts without the human, and is pushed by incentives toward generality.
- **Exponential operator, checked:** each event causes, on average, more than one further event like itself, as in a nuclear chain reaction or any replicator.
- **What care can do:** a hammer is safe with basic care; a nuclear reactor is an exponential held at one by extreme care; an existential technology is an exponential that nothing can hold at one, so that care changes how fast the outcome arrives and not whether it arrives.
- **Thirty doublings:** a process that doubles from one part in a billion of the world covers the whole of it in thirty doublings, so local and planetary differ only by a small number of steps.

- **Abstraction differential:** a human in the loop gives genuine oversight only while the human can understand what the system has done and why; past that point the loop becomes a ritual of approval.
- **Four marks:** a single pre-specified domain and objectives, limited planning horizons, no learning on its own from deployment, and a human at the point of decision.

Terms

Each term is defined where the reader first meets it; the entries below are reminders, in the book's own words.

Abstract replicator

An idea or tool, extracted from a particular instance by abstraction, that can be shared, recombined, and propagated independently of its origin (Chapter Two, Section III).

Abstraction

The capacity to extract a pattern from a particular instance and represent it independently of that instance (Chapter Two, Section III).

Abstraction differential

The gap in abstraction between a human in the loop and the system overseen; oversight is genuine only while it is small enough that the human can understand what the system has done and why (Chapter Six, Section VI).

Absolute argument

An argument that sets up a threshold: above a structural condition, the question asked has no answer, whatever the resources (Chapter Four, Section I).

Alignment

A consistent correspondence between a system's internal reference values and its effects in the world, maintained over time (Chapter Four, Abstract).

Alphabet (of a designed system)

The set of basic components, and the rules by which they combine, that a system is built from; nature's alphabet has decomposition matched to its synthesis, and every new letter a design adds needs a matched de-synthesis built by hand (Chapter One, The Alphabet of a Designed System).

Asymptotic limit

A value that a continuous function approaches and never reaches, as with infinity (Chapter Four, Narrow and General by Definition, Not by Limit).

Attractor

The set of states toward which a dynamical system settles over time (Chapter Five, Section VI).

Auto-scaling and auto-catalyzing processes

Processes that increase their own scale and speed up their own reactions, as a nuclear chain reaction does; existential threats result from them (Chapter Six, Section III).

Backward reference

A step in the control flow that can point back to an earlier step: recursion, a loop, or a backward jump; one of the three features of universality (Chapter Four, Section III).

Balkanization

The breaking up of a country into separate, often hostile, regions (Chapter Three, Hardening the Grid).

Bandwidth (input, output, compute)

The amount of data a machine can take in, the amount of action it can put out into the world, and the amount of hardware it can compute with (Chapter Six, Section VI).

Biological nutrient; technical nutrient

McDonough and Braungart's two cycles: a biological nutrient is designed to return safely to soil and water, where living things take it apart; a technical nutrient is designed to circulate in industrial cycles at full quality (Chapter One, Section II).

Black box

A system known only by what goes into it and what comes out (Chapter Four, Section VI).

Blockchain

A chain of data blocks in which each block's hash depends on the previous block's hash (Chapter Five, Section V).

Borderlands

The region between narrow and general artificial intelligence, where a system is no longer specific to one topic (Chapter Four, Narrow and General by Definition, Not by Limit).

Bottle tax

A tax charged on bottles; an example of the regulation the industry lobbies against (Chapter One, When Something Becomes Pollution).

Break-versus-build asymmetry

The third constraint: as insight becomes powerful enough, it becomes easier to break fundamental things than to build or repair them, because the insight required to create a powerful technology is far smaller than the knowledge required to use it safely (Chapter Two, Abstract; Section I).

Brittleness

The tendency of a system to fail suddenly, rather than bend, when disturbed (Chapter Five, Section IX).

Broadcast channel

A channel through which one sender reaches every receiver at once (Chapter Five, Section IX).

Buffer regions

In error correction, the margins of invalid signals kept around each valid one, so that a damaged signal can be corrected to the nearest valid one (Chapter Four, Section VII).

Carrington event

A solar storm great enough to damage electrical equipment across wide areas; a Carrington-style event is one of that size (Chapter Three, Hardening the Grid).

Cascading failure

A failure in one domain that propagates through coupled domains, each triggered by the failure of the domain immediately upstream (Chapter Three, Abstract; Section VIII).

Categorical situation

A situation different in kind, and not only in degree, from ordinary ones, and so needing a response of its own kind; a blackout long enough for well-founded doubt that the power will ever return is one (Chapter Three, Hardening the Grid).

Chain reaction

A process in which each event causes further events like itself, as each fission of a uranium nucleus releases neutrons that cause further fissions; it grows by multiples when each event causes, on average, more than one more (Chapter Six, Sections II and III).

Chaotic threshold (divergence threshold of the ring)

The critical level of hashiness above which noise grows at every traversal of the ring of controller, agent and world; a different line from the threshold of universality (Chapter Four, Sections VI and VII).

Chlorofluorocarbons

Compounds of chlorine, fluorine and carbon, developed in 1928 as safe refrigerants; so stable that they reach the stratosphere, where their chlorine destroys ozone (Chapter Two, Sections I and VII).

Circuit breakers (in markets)

Automatic pauses in trading, set off by a large fall in prices (Chapter Three, Section IV).

Circuit depth

The longest chain of steps in a computation in which each step must wait for the one before (Chapter Four, Section II).

Complicated and complex

A complicated system has many parts with deterministic interactions and can in principle be modeled by a sufficiently large computation; a complex system has many parts with nonlinear feedback, emergent behavior and sensitivity to initial conditions, and cannot be captured by any model simpler than itself (Chapter Two, Section II).

Computational universality

The capacity to run any program (Introduction; Chapter Four, Section III).

Conditionality (data-dependent control flow)

The link between data and control flow by which what happens next depends on what the data are; one of the three features of universality (Chapter Four, Section III).

Control flow

The order in which a program's steps are carried out (Chapter Four, Section III).

Control theory

The engineering study of how to observe a system and act on it so that its output stays near a desired goal (Chapter Five, Abstract).

Closure by control; closure by disposition

The two closures of the case about artificial general intelligence. The closure by control is the finding that no method of control suffices; the closure by disposition is the finding that a self-sustaining machine, by what its substrate needs, converges on conditions in which organic life cannot continue, whatever it was built to want. If AGI is built, each closure by itself settles the outcome (Chapter Six, Section V).

Coordination failure

The breakdown of effective cooperation among competent actors, when no participant has the surplus capacity needed to integrate the efforts of the others (Chapter Three, Section VI).

Cradle to cradle

Design in which the end of one use of a material is planned, from the start, as the beginning of its next; the term is William McDonough's and Michael Braungart's (Chapter One, Abstract; Section II).

Critical point

The point at which a phase transition happens; there can be no phase transitions without one. The book uses the words for any point at which a transition happens; physicists reserve them for a transition of the continuous kind (Chapter Five, Section VII).

Critical substation

A substation whose loss would cascade into wide failure of the grid (Chapter Three, Hardening the Grid).

Criticality

The condition in which a small disturbance can set off a large change (Chapter Four, Section VII). A system at criticality is balanced between order and disorder, and information percolates through it most efficiently (Chapter Five, Section VII).

Cryptographic hash function

A function built so that a single bit of change in its input randomizes its entire output; the limiting case of hashiness (Chapter Four, Section VI).

Cycle of nature

The ongoing process by which living things and natural processes take materials apart and use them again; a thing that exits it has become pollution (Chapter One, When Something Becomes Pollution).

De-synthesis

The deliberate, engineered separation of a built thing into parts that can be combined again; the neglected inverse of synthesis (Chapter One, The Alphabet of a Designed System).

Designed replicator; evolved replicator

An evolved replicator is held in check by co-evolution with the other life around it; a designed replicator enters an ecosystem from outside the evolutionary process and is not, so that its coherence with its ground has to be designed; every abstract replicator is a designed replicator (Chapter Two, Section III).

Domain

A field of inputs and outputs that a system takes in or acts on, such as language or a game (Chapter Four, Narrow and General by Definition, Not by Limit).

Economic framework

The costs and prices within which a business must work in order to stay open (Chapter One, When Something Becomes Pollution).

Electromagnetic pulse

A burst of electromagnetic energy, from the sun or from a nuclear explosion, that can destroy electrical equipment; a solar storm drives a slow current through long power lines and overheats transformers, while a nuclear pulse does that and also delivers a fast surge that destroys electronics (Chapter Three, Hardening the Grid).

Energy gap

The margin that keeps a system within its consistent set of states, against being pushed outside it (Chapter Five, Section IX).

Entropy (of a breakdown)

The loss of usable order: the degradation of a material from a well-organized, high-value state into a disordered, low-value one (Chapter One, Section III).

Entropy function

Disorder that grows whatever energy is spent; what a loop has above the chaotic threshold, in place of the convergence a heat pump can buy with energy (Chapter Four, Section VI).

Epistemology; of symmetry; of continuity

An epistemology is a way of knowing: how anything comes to be known, checked and held. An epistemology of symmetry sees parts, and what stays the same when parts are exchanged or moved; an epistemology of continuity follows a whole through its ongoing changes (Chapter Two, Section VII).

Error correction

Detecting when a signal or a system has departed from what was intended, and restoring it (Chapter Four, Section VII).

Excess deaths

Deaths above the number expected in ordinary years (Chapter Three, Section VII).

Exponential operator

A process whose growth feeds further growth, so that it grows by multiples over time instead of by fixed amounts (Chapter Six, Section II).

Externality

A cost of an action borne by people outside the exchange (Chapter Two, Section VII).

Feedback cycle

A loop in which the result of an action strengthens the reasons to repeat it (Chapter One, When Something Becomes Pollution; Chapter Five, Section IX).

Fixed points

States that a process returns unchanged, where its behavior can be pinned down (Chapter Four, Section IV).

Formal verification

Proving that a particular program meets its specification; the question of the single analyzed system in its working form, since each verified program is one analyzed point (Chapter Four, Section IV).

Gaussian distribution

The bell curve: outcomes that cluster around an average and so stay local, in contrast to exponential outcomes (Chapter Six, Section VI).

General artificial intelligence (AGI)

An artificial intelligence whose inputs and outputs encompass multiple domains, which continues to learn, and whose learning can change how it learns (Chapter Four, Narrow and General by Definition, Not by Limit); a general self learning machine, which applies to everything, eventually (Chapter Six, Abstract; Section V).

Gini coefficient of query energy

A number measuring how unevenly computational cost is spread across queries, used in a working search engine to mark the boundary of control (Chapter Five, Section X).

Good regulator theorem

The result that any regulator which controls a system well must contain a model of that system (Chapter Five, Section II).

Grid hardening

Making the electrical power grid able to withstand events that would otherwise destroy it (Chapter Three, Hardening the Grid).

Halting problem

The question whether a given program, run on a given input, will ever stop; Alan Turing proved in 1936 that no program can answer it correctly for every program and input (Introduction; Chapter Four, Section IV).

Hashiness

A measure, from zero to one, of how much a black box amplifies the noise it receives; equally, the probability that a controller's best possible input still fails because of internal state the controller cannot see (Chapter Four, Section VI). Of a mapping, the degree of non-linearity it introduces: how rapidly small changes in the input are amplified into large, unpredictable changes in the output, which places the mapping on the spectrum from the linear system to the hash function (Chapter Five, Section III).

Heavy-tail behavior

A distribution in which a small fraction of cases have extremely large values, here a few queries costing vastly more than the rest (Chapter Five, Section X).

Holographic search engine

The working search system in which the heavy-tailed queries of Chapter Five were observed (Chapter Five, Section X).

Holding

Keeping a coupled system within a safe range for a time, by continued effort; holding cannot be indefinite (Chapter Three, Section VII; Chapter Four, Section VIII).

Human step

A step in a material cycle that waits on a human choice, such as returning a bottle; where one exists, the reliability of the cycle becomes a question of human follow-through (Chapter One, Cycles Within Cycles).

Incommensuration Theorem

Symmetry and continuity cannot both be applied, simultaneously and fundamentally, to any event or comparison; and neither can asymmetry and discontinuity. The Incommensuration Theorem is a structural fact about the nature of knowing (Chapter Two, Section VII).

Instantiated replicator

The concrete, embodied natural or human structure from which an abstract replicator's capacity to exist was drawn; to be instantiated is to exist as a particular, embodied thing (Chapter Two, Section IV).

Inversion of protective function

The phenomenon by which an institution designed to protect against a harm becomes, under overload, a source of that harm; the signature pattern of cascading failure (Introduction; Chapter Three, Section II).

Iterated function system and bifurcation analysis

The study of mappings applied repeatedly, and of how their long-run behavior changes in kind as a parameter is varied (Chapter Five, Section VI).

Just-in-time logistics

Delivering parts and goods only as they are needed, so that little is held in stock (Chapter Two, Section VIII; Chapter Three, Section VII).

Laplace transform, transfer function, Bode plot

Standard mathematical tools of classical control theory for describing how a linear system responds to inputs of different frequencies (Chapter Five, Section II).

Large language model system

A learning system that can convert things to language and turn the output of language into actions in multiple domains; it sits in the borderlands between narrow and general (Chapter Four, Narrow and General by Definition, Not by Limit).

Linear system

A system whose output changes in proportion to its input, so that its behavior can be predicted exactly (Chapter Five, Section II).

Lobbying

Pressing a government to make or block laws in one's own interest (Chapter One, When Something Becomes Pollution).

Margins

The difference between what a business takes in for a sale and what the sale costs it (Chapter One, When Something Becomes Pollution).

Mastery of choice

Skill in choosing well; in *The Substrate and the Ecosystem*, its exercise in determining what technologies are permitted to exist (Chapter Four, Section VIII).

Meme

Richard Dawkins's name for an idea that copies itself from mind to mind; the abstract replicator includes memes and also tools built from shared designs (Chapter Two, Section III).

Meta-recursion

A feedback cycle in which the parts act on the controller in order to act, through it, on one another (Chapter Five, Section IX).

Metastability

A dynamic, actively maintained equilibrium between the abstract and the instantiated, in which each supports the other; the governing criterion for whether an abstract replicator is safe to build (Chapter Two, Section VIII).

Mistake theory

The view that harm comes mostly from errors of understanding, and not from conflicts of interest (Chapter Two, Section VII).

Modeling criterion

The requirement that a controller can model a system well enough to predict whether it will stay safe; above the threshold of universality it fails categorically (Chapter Four, Section IV).

Monotonic

Increasing at every step, and never falling back (Chapter Four, Section VI).

Monstrous hybrid

A product that binds biological and technical materials together so that neither can be recovered; McDonough and Braungart's term, borrowed from Jane Jacobs (Chapter One, Section II).

Morphology

The form and structure of a body or of a part of it, such as the face (Chapter Two, The Two Prior Constraints Stated as Absolutes).

Narrow artificial intelligence

An artificial intelligence whose inputs and outputs encompass a single domain (Chapter Four, Narrow and General by Definition, Not by Limit).

Narrow level two

A name for a system at the threshold between narrow and general, such as a large language model system (Chapter Four, Narrow and General by Definition, Not by Limit).

No-free-lunch theorems

The results of David Wolpert and William Macready (1997) that, averaged over all possible problems, no search or learning method does better than any other (Chapter Four, Section IV).

Noise floor

The level of background noise below which a signal cannot be told apart from the noise (Chapter Five, Section IX).

Non-biodegradable

Not able to be broken down by living things (Chapter One, When Something Becomes Pollution).

Non-entropic breakdown

Breakdown, reuse and recycling that preserve as much of a material's original order and usability as possible (Chapter One, Section III).

NP-complete

Belonging to the class of hardest problems whose solutions can be checked quickly; every problem in that class (NP) can be reduced to any NP-complete problem (Chapter Five, Section VIII).

Overton window

The range of ideas that the public and lawmakers are ready to take seriously at a given time (Chapter Three, Hardening the Grid).

Ozone

The gas in the stratosphere that screens the surface of the Earth from the sun's ultraviolet light (Chapter Two, Section I).

Parasitic machinery

Machines that take from or harm the people and things they are used among, in contrast to tools that do what their user intends (Chapter Six, Section I).

Penumbra argument

Analyzable instances are distributed among non-analyzable ones as the rationals among the continuum, so a small perturbation moves one into the other (Chapter Four, Section IV).

Percolation

The passing of something (here, information) through a whole network by way of connected paths; a percolation transition is the point at which such paths suddenly span the network (Chapter Five, Sections VII and IX).

Period-doubling cascade

A sequence of changes in which a system's cycle doubles in length again and again as a parameter increases, ending in chaos (Chapter Five, Section VI).

Phase transition

An abrupt change in the character of a system at a sharp boundary, such as water turning to steam at its boiling point; the boundary between the controllable and the uncontrollable regions of the hashiness spectrum is one (Chapter Five, Section VII).

PID controller

A standard controller that corrects a system using the present error, the accumulated past error, and the rate at which the error is changing (Chapter Five, Section II).

Pollution

A sub-case of toxicity: something nature does not know how to deal with, like plastic in the ocean; a thing becomes pollution at the moment it is created in a form nature cannot take apart, when it exits the cycle of nature (Chapter One, When Something Becomes Pollution).

Protected symmetry

The invariants that an epistemology of symmetry holds fixed while it exchanges and moves parts; by the Incommensuration Theorem, it and continuity cannot both be maximized at once (Chapter Two, Section VII).

Recursion

Applying a mapping again and again, each output becoming the next input (Chapter Five, Section V).

Redundancy

Spare capacity or equipment held in reserve so that a system can keep working when parts fail (Chapter Three, Hardening the Grid).

Relative argument

An argument that sets up an inequality: as one parameter rises, some cost rises faster, in a way that cannot be caught up with (Chapter Four, Section I).

Remediation

The technical work of fixing a weakness (Chapter Three, Hardening the Grid).

Reproductive number

The average number of secondary infections produced by each infected individual (Chapter Three, Section I).

Reversible computation

Computing in which no information is erased, which in principle lowers the energy each step must cost (Chapter Four, Section II).

Rice's theorem

Proved by Henry Rice in 1953: for any property of what a program does that some programs have and others lack, no general method can decide, for every program, whether it has that property (Chapter Four, Section IV).

Ring (controller, agent, world)

The minimal architecture in which alignment can be posed: the controller acts on the agent, the agent acts on the world, and the outputs of the world return to the controller (Chapter Four, Section V).

Rivalrous

Of risk dynamics: each actor gains only by others' losing (Chapter Two, Section II).

Safety predicate

A yes-or-no property of a system, such as "never enters a harmful state" (Chapter Four, Abstract).

SAT (satisfiability problem)

The problem of deciding whether some assignment of true and false to the variables of a logical formula makes the formula true; the canonical computational phase transition (Chapter Five, Section VIII).

Second-law structure

The structure of the second law of thermodynamics, under which the disorder of a closed system never decreases; around the ring, uncertainty increases at every traversal (Chapter Four, Section VI).

Self-terminating

Destroying the conditions that let it continue (Chapter Two, Section II).

Sequential contingency

That this happens, and then that: one of the three features of universality (Chapter Four, Section III).

Substrate

In Chapter Two, the ground of instantiated replicators, natural and human, on which an abstract replicator depends for its own continuation (Chapter Two, Section VIII). In Chapter Six, the physical material and hardware a machine is made of and runs on (Chapter Six, Section II).

Superconductivity, superfluidity

Coherent states of matter reached suddenly on cooling, in which electric current or a fluid flows without resistance (Chapter Five, Section IX).

Symbiotic relationship

A relationship in nature where one organism lives on another but performs some function the host needs, at a reasonable cost (Chapter Six, Section I).

Symbolic dynamics

A method that divides the states of a continuous system into regions and records the system's motion as the sequence of regions it visits (Chapter Five, Section VI).

Tacit intentionality

Goal-like behavior that emerges from a system's own evolution without anyone having designed it (Chapter Six, Section II).

Theory of mind

The capacity to attribute thoughts and intentions to others (Chapter Two, Section VI).

Threat model, threat modeling

The account of what a threat is, how it would act, and what it would take to mitigate it; threat modeling is working that account out (Chapter Three, Hardening the Grid).

Threshold theorem

A result stating that once noise is below a certain level, reliable transmission or computation becomes possible, as in Shannon's channel theorems and von Neumann's reliable computer built from fragile components (Chapter Five, Section IX).

Tight coupling

A relationship between systems in which each depends on the continued normal working of the others, with little slack between them; Charles Perrow's term (Chapter Three, Abstract; Section I).

Top-down design

Engineering that starts from the large and works down to the small, breaking big problems into smaller ones and solving them piecemeal (Chapter One, Why Technology Is Toxic).

Topologically equivalent

Having the same shape of connections and neighborhoods, so that the qualitative structure is preserved (Chapter Five, Section VI).

Toxic

A technology is toxic to the degree that it degrades the ground it depends on; an abstract replicator is toxic to the degree that it fails to support the stability of the ground that produced it (Chapter One, Abstract; Section IV; Chapter Two, Section VIII). Toxicity is too much of something or too little of something needed (Chapter One, Why Technology Is Toxic).

Toy model

A deliberately simplified model used to reason about a problem (Chapter Four, Section III).

Trinity of compute

Time, space, and energy: the three costs of any computation, against which any growth in capability is measured (Introduction; Chapter Four, Section II).

Turing machine

Alan Turing's model of a computer: a device that reads and writes symbols on an unlimited tape by fixed rules (Chapter Four, Abstract).

Type 1, Type 2 and Type 3 coherence

Type 1 coherence is the coherence of resonance, akin to that found in a laser. Type 2 coherence is network coherence, such as a logical net, or a belief system coherent as a set of ideas working together. Type 3 coherence is the coherence in which multiple levels of abstract causation interact in a way that does not result in overall disorder, as in the homeostasis of the relation of mind and body, or the integration of software with hardware. The effects of the first are additive, of the second multiplicative, and of the third exponential. Between an abstract replicator and its ground, Type 3 coherence is a relationship in which the abstract replicator actively supports, and does not merely tolerate, the continued integrity of the concrete ground it came from; once broken, it collapses entirely (Introduction; Chapter Two, Section IV).

Unconstrained and unconstrainable

Not limited in fact, and not able to be limited by any means (Chapter Six, Section III).

Undecidable region

The middle of the hashiness spectrum, where no general procedure can determine whether control of a given system will succeed (Chapter Five, Section IV).

Universal model computing machine

A machine, working by finite means, that can simulate any finitely realizable physical system to any chosen degree of approximation (Chapter Five, Section VII).

Virtual machine

A program that runs other programs (Chapter Four, Section III).

X-risk (existential risk)

A risk that ends the whole ecosystem, and with it all humans (Chapter Six, Section II).

Bibliography

Works on computation, information and design

Böhm, Corrado, and Giuseppe Jacopini. "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules". *Communications of the ACM* 9, no. 5 (1966): 366 to 371. DOI 10.1145/355592.365646.

Conant, Roger C., and W. Ross Ashby. "Every Good Regulator of a System Must Be a Model of That System". *International Journal of Systems Science* 1, no. 2 (1970): 89 to 97. DOI 10.1080/00207727008920220.

Cook, Stephen A. "The Complexity of Theorem-Proving Procedures". *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (1971): 151 to 158. DOI 10.1145/800157.805047.

Dawkins, Richard. *The Selfish Gene*. Oxford: Oxford University Press, 1976.

Deutsch, David. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". *Proceedings of the Royal Society of London A* 400, no. 1818 (1985): 97 to 117. DOI 10.1098/rspa.1985.0070.

Hamkins, Joel David, and Alexei Miasnikov. "The Halting Problem Is Decidable on a Set of Asymptotic Probability One". *Notre Dame Journal of Formal Logic* 47, no. 4 (2006): 515 to 524. DOI 10.1305/ndjfl/1168352664.

Loomis, Elisha Scott. *The Pythagorean Proposition*. Second edition, 1940. Reprinted Washington, D.C.: National Council of Teachers of Mathematics, 1968. ISBN 978-0-87353-036-1.

McDonough, William, and Michael Braungart. *Cradle to Cradle: Remaking the Way We Make Things*. New York: North Point Press, 2002. ISBN 978-0-86547-587-8.

Mézard, Marc, and Riccardo Zecchina. “Random K-Satisfiability Problem: From an Analytic Solution to an Efficient Algorithm”. *Physical Review E* 66, no. 5 (2002): 056126. DOI 10.1103/PhysRevE.66.056126.

Mitchell, David, Bart Selman, and Hector Levesque. “Hard and Easy Distributions of SAT Problems”. *Proceedings of the Tenth National Conference on Artificial Intelligence* (1992): 459 to 465.

Neumann, John von. “Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components”. In *Automata Studies*, edited by C. E. Shannon and J. McCarthy, 43 to 98. Princeton: Princeton University Press, 1956.

Perrow, Charles. *Normal Accidents: Living with High-Risk Technologies*. New York: Basic Books, 1984. ISBN 0-465-05142-1.

Rice, H. G. “Classes of Recursively Enumerable Sets and Their Decision Problems”. *Transactions of the American Mathematical Society* 74, no. 2 (1953): 358 to 366. DOI 10.1090/S0002-9947-1953-0053041-6.

Shannon, C. E. “A Mathematical Theory of Communication”. *Bell System Technical Journal* 27 (1948): 379 to 423 and 623 to 656. DOI 10.1002/j.1538-7305.1948.tb01338.x (first part).

Turing, A. M. “On Computable Numbers, with an Application to the Entscheidungsproblem”. *Proceedings of the London Mathematical Society*, series 2, 42 (1936): 230 to 265. DOI 10.1112/plms/s2-42.1.230.

Wolpert, David H., and William G. Macready. “No Free Lunch Theorems for Optimization”. *IEEE Transactions on Evolutionary Computation* 1, no. 1 (1997): 67 to 82. DOI 10.1109/4235.585893.

Reports and data for Chapters One to Four

Al Jazeera. “Circuit breakers halt US stock trading. What are they exactly?”. 9 March 2020, https://www.aljazeera.com/economy/2020/3/9/circuit-breakers-halt-us-stock-trading-what-are-they-exactly (Chapter Three, Section IV: the circuit breakers of March 2020)

ANSA. “Protests in 27 prisons”. 9 March 2020, https://ansa.it/amp/english/news/general_news/2020/03/09/protests-in-27-prisons_eec7e3a4-66c8-4de6-a728-ae6e19e43ae2.html (Chapter Three, Section III: the riots in Italian prisons)

Associated Press. “Prison riots hit Italy amid virus; 6 die in overdose”. 9 March 2020, in the Washington Times, https://m.washingtontimes.com/news/2020/mar/9/prison-riots-hit-italy-amid-virus-6-die-overdose/ (Chapter Three, Section III: the deaths in the Italian prison riots)

Australian Aluminium Council. “Aluminium Recycling Saves 95% of the Energy Needed for Primary Aluminium Production” (International Aluminium Institute factsheet series). 10 October 2024, https://aluminium.org.au/news/iai-factsheet-claim-2-aluminium-recycling-saves-95-of-the-energy-needed-for-primary-aluminium-production/ (Chapter One, Section III: the aluminum can)

Bloomberg. “Starbucks Barista in Downtown Seattle Store Diagnosed With Virus”. 6 March 2020, https://www.bnnbloomberg.ca/starbucks-barista-in-downtown-seattle-store-diagnosed-with-virus-1.1401680 (Chapter Three, Section IV: the barista in Seattle)

Board of Governors of the Federal Reserve System. *Report on the Economic Well-Being of U.S. Households in 2019, Featuring Supplemental Data from April 2020*. May 2020, https://www.federalreserve.gov/publications/2020-economic-well-being-of-us-households-in-2019-accessible-version.htm (Chapter Three, Section IV: the four hundred dollar expense)

Centers for Disease Control and Prevention. “Adult Obesity Facts”. Consulted 5 November 2026, https://www.cdc.gov/obesity/adult-obesity-facts/index.html (Chapter Three, Section VII: obesity among adults in the United States)

Commodity Futures Trading Commission and Securities and Exchange Commission. *Findings Regarding the Market Events of May 6, 2010*. Report of the staffs of the two commissions to the Joint Advisory Com-

mittee on Emerging Regulatory Issues. 30 September 2010, https://www.sec.gov/news/studies/2010/marketevents-report.pdf (Chapter Four, Section VI, the sidebar on automated trading: the Flash Crash)

Express and Star. "Italy inmates at large after coronavirus-related prison riots". 11 March 2020, https://www.expressandstar.com/news/world-news/2020/03/11/italy-inmates-at-large-after-coronavirus-related-prison-riots/ (Chapter Three, Section III: the escapes from Italian prisons)

Glass Packaging Institute. "Facts About Glass Recycling". Consulted 5 November 2026, https://gpi.org/glass-recycling-facts (Chapter One, Section III: recycled glass and the energy of the melt)

Goethe-Institut. "Germany's Bottle Obsession". September 2025, https://www.goethe.de/prj/ger/en/wow/nachhaltig-erklaert/27021598.html (Chapter One, Section III: refillable bottles)

Herald Live. "Cruise ship coronavirus cases climb to 175, including quarantine officer". 12 February 2020, https://www.heraldlive.co.za/news/2020-02-12-cruise-ship-coronavirus-cases-climb-to-175-including-quarantine-officer/ (Chapter Three, Section II: the quarantine officer on the Diamond Princess)

HPN Online. "CDC data finds adult obesity is increasing". 18 September 2020, https://www.hpnonline.com/regulatory/article/21154871/cdc-data-finds-adult-obesity-is-increasing (Chapter Three, Section VII: obesity among adults in the United States)

Lauer, Stephen A., Kyra H. Grantz, Qifang Bi, Forrest K. Jones, Qulu Zheng, Hannah R. Meredith, Andrew S. Azman, Nicholas G. Reich, and Justin Lessler. "The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application". *Annals of Internal Medicine* 172, no. 9 (2020): 577 to 582. DOI 10.7326/M20-0504. (Chapter Three, Section VII: the incubation period)

Molina, Mario J., and F. S. Rowland. “Stratospheric Sink for Chlorofluoromethanes: Chlorine Atom-Catalysed Destruction of Ozone”. *Nature* 249 (1974): 810 to 812. DOI 10.1038/249810a0. (Chapter Two, Section I: chlorofluorocarbons and ozone)

MercoPress. “Oil prices plunge 30% after OPEC fails to agree on production cuts; Russia clashes with Saudi Arabia”. 9 March 2020, https://en.mercopress.com/2020/03/09/oil-prices-plunge-30-after-opec-fails-to-agree-on-production-cuts-russia-clashes-with-saudi-arabia (Chapter Three, Section IV, the sidebar: the oil price war)

NPR. “New York Creates Containment Area Around Cluster in New Rochelle”. 10 March 2020, https://www.npr.org/sections/health-shots/2020/03/10/814099444/new-york-creates-containment-area-around-cluster-in-new-rochelle (Chapter Three, Section III: the containment zone in New Rochelle)

Organisation for Economic Co-operation and Development. *How's Life? 2020: Measuring Well-being*. Paris: OECD Publishing, 2020. DOI 10.1787/9870c393-en. (Chapter Three, Section IV: households at risk of poverty)

Severinghaus, Jeffrey P., Wallace S. Broecker, William F. Dempster, Taber MacCallum, and Martin Wahlen. “Oxygen Loss in Biosphere 2”. *Eos, Transactions American Geophysical Union* 75, no. 3 (1994): 33 to 37. DOI 10.1029/94EO00285. (Chapter One, “Why Technology Is Toxic”: Biosphere 2)

United Nations Environment Programme, Law and Environment Assistance Platform. “Multi-layered plastic packaging” (plastics toolkit glossary). Consulted 5 November 2026, https://leap.unep.org/en/knowledge/toolkits/plastic/glossary/multi-layered-plastic-packaging (Chapter One, Section III: multilayered packages)

Wikipedia. “2020 Russia-Saudi Arabia oil price war”. Consulted 5 November 2026, https://en.wikipedia.org/wiki/2020_Russia%E2%80%93Saudi_Arabia_oil_price_war (Chapter Three, Section IV, the sidebar: the oil price war)

Wikipedia. “Origin of language” (survey of the estimates of when language emerged), https://en.wikipedia.org/wiki/Origin_of_language (Chapter Two, Section VI: the emergence of language)

World Economic Forum. “Mad March: how the stock market is being hit by COVID-19”. 23 March 2020, https://www.weforum.org/stories/2020/03/stock-market-volatility-coronavirus (Chapter Three, Section IV: the trading halts of March 2020)

World Health Organization. “14.9 million excess deaths associated with the COVID-19 pandemic in 2020 and 2021”. 5 May 2022, https://www.who.int/news/item/05-05-2022-14.9-million-excess-deaths-were-associated-with-the-COVID-19-pandemic-in-2020-and-2021 (Chapter Three, Section VII: excess deaths)

My own works

Landry, Forrest. *An Immanent Metaphysics*. First published 2001. Wytheville, Virginia: UVSM Publications. ISBN 979-8-9896628-4-5. Cited by aphorism number.

Landry, Forrest, Remmelt Ellen, and Will Petillo. *Ape Trap; The Runaway Artificialization of Our Planet*. Edited by Remmelt Ellen. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-951203-55-7. DOI https://doi.org/10.67291/FQER8044.

Landry, Forrest. *Designing a Civilization That Does Not Destroy Itself; Conversations in Applied Metaphysics*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-371-9. DOI https://doi.org/10.67291/RWRX7518.

Landry, Forrest. *Evolution, Design, and the Ethics of the Possible; Seventeen Essays on Technology, Nature, and the Gap Between Capability and Wisdom*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-161-6. DOI https://doi.org/10.67291/HDHJ8250.

Landry, Forrest. *Minesweeper Madness; Why Even Infinite Intelligence Does Not Solve the World*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-249-1. DOI https://doi.org/10.67291/DLRM7000.

Landry, Forrest. *Sense-Making and the Craft of Inquiry; Essays on Purpose, Questions, Group Discipline, Notation, and Design*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-353-5. DOI https://doi.org/10.67291/DIOM8597.

Landry, Forrest. *The Ceiling Below the Floor; The Containment Fallacy and AGI Non-Safety*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-216-3. DOI https://doi.org/10.67291/SCTM1058.

Landry, Forrest. *The Displacement of Choice; The Catastrophe of the AI We Might Build*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-243-9. DOI https://doi.org/10.67291/DXJL7493.

Landry, Forrest. *The Geometry of Man, Machine, and Nature; Care, Community, and the Design of Technology*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-244-6. DOI https://doi.org/10.67291/ZIHO9910.

Landry, Forrest. *The Substrate and the Ecosystem; Why Artificial General Intelligence Cannot Be Aligned, and What Remains for Persons*. Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-90710-246-0. DOI https://doi.org/10.67291/NYZX3639.

Landry, Forrest. *Why AGI Cannot Be Made Safe; The Argument, Its Physics, and the One Choice That Remains* (Volume 1 of *The Impossibility of AGI Safety; The Substrate Needs Convergence Argument*). Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-951203-02-1. DOI https://doi.org/10.67291/UIBT8862.

Landry, Forrest. *Why No Control Suffices; The Technical Case for the Impossibility of AGI Safety* (Volume 2 of *The Impossibility of AGI Safety; The Substrate Needs Convergence Argument*). Wytheville, Virginia: UVSM Publications, 2026. ISBN 979-8-951203-03-8. DOI https://doi.org/10.67291/MZQS9079.

Index

Absolute argument: Chapter Four, I, IV, VI, VIII; Introduction

Absolutes, constraints stated as: Chapter Two, The Two Prior Constraints Stated as Absolutes

Abstract replicator: Chapter Two, III, IV, VIII, IX; Introduction

Abstraction capacity: Chapter Two, III, VI, IX

Abstraction differential: Chapter Six, VI

Alignment: Chapter Four, Abstract, V, VII

Alphabet of a designed system: Chapter One, The Alphabet of a Designed System, When Something Becomes Pollution; Chapter Two, II

Aluminum can: Chapter One, III

Artificial general intelligence, four aspects of: Chapter Six, V, VI; Introduction

Asymptotic limit: Chapter Four, Narrow and General by Definition, Not by Limit

Attention, as a finite resource: Chapter Three, V

Auto-scaling and auto-catalyzing processes: Chapter Six, III

Biological and technical nutrients: Chapter One, II

Biosphere 2, trees and oxygen in: Chapter One, Why Technology Is Toxic

Blockchain: Chapter Five, V

Böhm and Jacopini: Chapter Four, III

Borderlands, between narrow and general: Chapter Four, Narrow and General by Definition, Not by Limit

Bottle tax: Chapter One, When Something Becomes Pollution

Break-versus-build asymmetry: Chapter Two, I, III, IV, IX

Broadcast channel, shared controller as: Chapter Five, IX

Buffer regions: Chapter Four, VII

Carrington event: Chapter Three, Hardening the Grid

Cascading failure: Chapter Three, VIII, IX

Categorical blackout: Chapter Three, Hardening the Grid

Chain reaction, nuclear: Chapter Six, II, III

Chaotic threshold: Chapter Four, VI, VII

Chlorofluorocarbons: Chapter Two, I, V, VII
Circuit breakers: Chapter Three, IV; Chapter Four, VI (sidebar)
Closed loop: Chapter One, I, II, V; Chapter Two, V
Closures, by control and by disposition: Chapter Six, V; Introduction
Coherence, three types of (resonance, network, multiple levels of abstract causation): Chapter Two, IV
Complicated and complex: Chapter Two, II
Computational universality, three features of: Chapter Four, III, VIII
Control theory: Chapter Five, Abstract, II, IV
Coordination problem: Chapter Three, VI, VIII
Coupled civilization, above the threshold: Chapter Four, III; Introduction
Cradle to cradle: Chapter One, Abstract, II
Critical point: Chapter Five, VII
Criticality: Chapter Four, VII; Chapter Five, VII
Cyanobacteria, oxygen of: Chapter Six, II
Cycle of nature: Chapter One, When Something Becomes Pollution
Cycles within cycles: Chapter One, Cycles Within Cycles
Deposit on bottles: Chapter One, Cycles Within Cycles
Designed and evolved replicators: Chapter Two, III
De-synthesis: Chapter One, The Alphabet of a Designed System, IV
Diamond Princess: Chapter Three, II
Diplomacy, the system that played: Chapter Four, Narrow and General by Definition, Not by Limit
Divergence threshold of the ring: Chapter Four, VI, VII, VIII
Domain: Chapter Four, Narrow and General by Definition, Not by Limit; Chapter Six, VI
Doubling, arithmetic of: Chapter Six, III, IV
Efficiency and fragility: Chapter Three, VII, VIII
Electromagnetic pulse: Chapter Three, Hardening the Grid
Energy gap: Chapter Five, IX
Entropy of a breakdown: Chapter One, III
Epistemology of symmetry, of continuity: Chapter Two, VII
Error correction: Chapter Four, VII; Chapter Five, IX

Excess deaths, 2020 and 2021: Chapter Three, VII
Existential technology, test for: Chapter Six, II, III; Introduction
Exponential processes, and thinking locally: Chapter Six, III, IV
Externality: Chapter Two, VII
Feedback, in evolution and in design: Chapter One, Why Technology Is Toxic; Chapter Two, II
Fixed points: Chapter Four, IV
Flash Crash of 6 May 2010: Chapter Four, VI (sidebar)
Formal verification: Chapter Four, IV
Four marks of a narrow system: Chapter Six, VI
Gaussian distribution, tails of: Chapter Six, VI
General artificial intelligence: Chapter Four, Narrow and General by Definition, Not by Limit; Chapter Six, V
Gini coefficient of query energy: Chapter Five, X
Glass bottles, refillable and single-use: Chapter One, III
Good regulator theorem: Chapter Four, VI; Chapter Five, II
Grid hardening: Chapter Three, Hardening the Grid
Halting problem: Introduction; Chapter Four, IV
Hamkins and Miasnikov: Chapter Four, IV
Hash function, cryptographic: Chapter Four, VI; Chapter Five, I
Hashiness: Chapter Four, VI, VII; Chapter Five, III, IV, XI
Heavy-tail behavior: Chapter Five, X
Holding a system within a safe range: Introduction; Chapter Three, VII, Hardening the Grid; Chapter Four, VIII; Chapter Five, X
Holographic search engine: Chapter Five, X
Human step, in a material cycle: Chapter One, Cycles Within Cycles
Hospitals, as infection sites: Chapter Three, II, VIII; Introduction
Incommensuration Theorem: Chapter Two, VII
Incubation period: Chapter Three, VII
Industry, responsibility for pollution: Chapter One, When Something Becomes Pollution
Influenza pandemic of 1918: Chapter Three, VII
Instantiated replicator: Chapter Two, IV, VIII

Inversion of protective function: Chapter Three, II, VIII; Introduction
Just-in-time logistics: Chapter Two, VIII; Chapter Three, VII
Language, emergence of: Chapter Two, VI
Large language model systems: Chapter Four, Narrow and General by Definition, Not by Limit
Learning, as placing a system above the threshold: Chapter Four, III, IV, VIII
Lily pond riddle: Chapter Six, IV
Load rating, design requirement as: Chapter Two, The Two Prior Constraints Stated as Absolutes
Loomis, Elisha: Chapter Four, I
Loop, delay in: Chapter Two, V
Man-machine-nature relation: Chapter Two, V
Mastery of choice: Chapter Four, VIII
McDonough, William, and Michael Braungart: Chapter One, II
Meme: Chapter Two, III
Metastability: Chapter Two, VIII, IX
Mézard, Marc, and Riccardo Zecchina: Chapter Five, VII
Midgley, Thomas: Chapter Two, I
Mistake theory: Chapter Two, VII
Modeling criterion: Chapter Four, IV
Molina, Mario, and F. Sherwood Rowland: Chapter Two, I
Monstrous hybrid: Chapter One, II
Narrow and general artificial intelligence: Chapter Four, Narrow and General by Definition, Not by Limit; Chapter Six, VI
Narrow level two: Chapter Four, Narrow and General by Definition, Not by Limit
National Guard, New Rochelle: Chapter Three, III
No-free-lunch theorems: Chapter Four, IV, VIII
Noise floor: Chapter Five, IX
Non-entropic breakdown: Chapter One, III, IV
Nuclear reactor, as an exponential held at one: Chapter Six, III
Obesity, and the immune baseline: Chapter Three, VII
Oil price war, 2020: Chapter Three, IV (sidebar)

Overton window: Chapter Three, Hardening the Grid
Ozone: Chapter Two, I, VII
Pandemic of 2020: Chapter Three, I, II, III, IV, VI, VII, IX; Chapter Four, III; Introduction
Parasitic machinery: Chapter Six, I, II
Penumbra argument: Chapter Four, IV
Percolation: Chapter Five, VII, IX
Perrow, Charles: Chapter Three, I
Phase transition: Chapter Five, VII, VIII, IX, XI; Introduction
Pollution: Chapter One, When Something Becomes Pollution
Predators, people as: Chapter Two, II, The Two Prior Constraints Stated as Absolutes
Prisons, Italian: Chapter Three, III
Protected symmetry: Chapter Two, VII
Pythagorean theorem: Chapter Four, I
Quarantine: Chapter Three, III, IV, VII
Relative argument: Chapter Four, I, II, VI, VIII
Reproductive number: Chapter Three, I
Reversible computation: Chapter Four, II
Rice, Henry: Chapter Four, IV
Rice's theorem: Chapter Four, IV, VIII
Ring of controller, agent and world: Chapter Four, V, VI, VIII; Introduction
Safety predicate: Chapter Four, Abstract, IV
SAT (satisfiability problem): Chapter Five, VIII
Savings, households without: Chapter Three, IV
Schools: Chapter Three, III
Second-law structure: Chapter Four, VI
Semiconductor fabrication, conditions of: Chapter One, IV; Chapter Six, II
Severity scale for existential events: Chapter Six, III
Shannon, Claude: Chapter Four, VII; Chapter Five, IX
Store and program, of a civilization: Chapter Four, III
Substrate: Chapter Two, VIII
Symbolic dynamics: Chapter Five, VI

Third constraint (break-versus-build asymmetry): Chapter Two, I, VIII, IX
Threat modeling: Chapter Three, Hardening the Grid
Three regions of the hashiness spectrum: Chapter Five, IV, VIII
Threshold theorems: Chapter Five, IX
Thresholds of abstraction (fire, tools, language, writing, printing, computation): Chapter Two, VI
Tight coupling: Chapter Three, I, VII, VIII
Tool and parasite: Chapter Six, I, II
Top-down design: Chapter One, Why Technology Is Toxic
Toxic, toxicity: Chapter One, Abstract, When Something Becomes Pollution, IV, Why Technology Is Toxic; Chapter Two, II, The Two Prior Constraints Stated as Absolutes, VIII
Toy model: Chapter Four, III
Trading, automated: Chapter Four, VI (sidebar)
Trajectory, parasitic quality of: Chapter Six, I
Trinity of compute: Chapter Four, II; Introduction
Turing, Alan: Introduction; Chapter Four, IV
Turing machine: Chapter Four, Abstract, III
Type 1, Type 2 and Type 3 coherence: Chapter Two, IV, IX; Introduction
Undecidable region: Chapter Five, IV, X
Virtual machine: Chapter Four, III
Von Neumann, John: Chapter Five, IX
Wolpert, David, and William Macready: Chapter Four, IV
X-risk (existential risk): Chapter Six, II