History Before Function

Compression as the Origin of Operators

Flyxion

2026

Abstract

This essay argues that function is not a primitive executable form but the compressed residue of a history: what looks like a rule, a skill, a procedure, or a theorem is, in every case examined here, the output of a compression operator C applied to an underlying trajectory H of successful and unsuccessful attempts. The claim is given a formal spine rather than left as an interpretive frame. Existence and non-uniqueness of generating histories, and the uncomputability of the minimal generator, follow from standard results in algorithmic information theory. What those results do not supply is any reason to prefer the minimal generator, and this essay's central new result — the Anti-Alignment Theorem — shows that the minimal generating history is, generically, the *worst* available choice for a distinct and independently important property: repairability, formalized using the repair-entropy apparatus of *Recursive Continuation*. Minimality and repairability are shown to be weakly anti-aligned by construction and strictly anti-aligned whenever discarded history carries diagnostic information. The essay also identifies a structural asymmetry between two things an operator can do once it exists — be enacted, and be represented symbolically — arguing that enactment is (near-)total while representation is partial, in the sense of Polanyi's tacit knowledge. This essay is intended to sit beneath *Recursive Continuation* rather than beside it: the history H compressed here is the same object as that essay's H_t, and the compression map C is one answer to a question that essay's Open Problem 1 left standing.

1 The Reversal

Most formal treatments of procedure, skill, and computation take the function, rule, or operator as the primitive object and treat any history of its use as secondary — a log, a trace, something that happens to the function rather than something the function is made of. This essay inverts that priority. The claim is not merely that operators *have* histories, in the way an artifact has a provenance that could in principle be told as a story. The claim is that an operator *is* a compressed history: a trajectory of attempts, corrections, and successes that has had its internal structure discarded, leaving behind an object that executes without narrating how it came to be executable.

Definition 1.1 (Compression)**.** *Let H denote a history: an ordered trajectory of states, attempts, corrections, and outcomes, in the sense of* Recursive Continuation*, §8, where H_t is introduced as the recoverable record a system retains of its own prior states. Let C be a compression map taking a history to an operator, $C(H) = F$, where F is characterized independently of C by its behavior: F is the map from inputs to outputs that a user or system actually observes when F is invoked, irrespective of how F came to exist.*

Remark. *The independence clause is not a formality. If "operator" were defined as "whatever C produces," the claim that operators are compressed histories would be true by construction and would say nothing. F must be picked out by its extensional behavior — what it does when run — before the question of whether some history compresses to it becomes a real question with a real answer.*

The canonical illustration, returned to throughout, is a text-expansion macro built from a remembered workflow:

`convert *.jpg -auto-orient scan.pdf ; ocrmypdf scan.pdf out.pdf`

(repeated, corrected, refined)

`::howtopdf::`

The history H is the sequence of attempts, failed invocations, and corrections that preceded the macro's existence: the original two-line command, the discovery that orientation needed correcting first, the trial runs against different scan qualities. The operator F is what `howtopdf` does today: it converts and OCRs a folder of images with a single keystroke, and its behavior is fully specified without any reference to how it was arrived at.

1.1 The Fork: Enactment and Representation

Once F exists, two different things can be done with it, and they are not the same kind of operation.

Definition 1.2 (Enactment and Representation)**.** *Let E be the enactment map: $E(F)$ is an instance of F being invoked and executed on some occasion. Let R be the representation map: $R(F)$, where defined, is a symbolic, inspectable description of F — a rule, a formula, a written procedure — sufficient to reconstruct F's behavior without executing it.*

Proposition 1.3 (Asymmetry of Enactment and Representation)**.** $\mathrm{Dom}(E) \not\subseteq \mathrm{Dom}(R)$ *in general; that is, there exist operators F with $F \in \mathrm{Dom}(E)$ but $F \notin \mathrm{Dom}(R)$.*

Proof. By construction: an operator qualifies as an operator at all only if it can be invoked, so E is total on the class of operators under consideration, up to trivial degenerate cases. Representability is a further, independent condition — the existence of a symbolic description adequate to reconstruct the behavior without running it — and is witnessed as failing outright by any skill exhibiting the structure Polanyi calls tacit knowledge: riding a bicycle is reliably enactable by a competent rider ($F \in \mathrm{Dom}(E)$) without the rider possessing, or in general being able to produce, a symbolic account of the balance corrections involved ($F \notin \mathrm{Dom}(R)$ for that rider). A single witness suffices to establish the non-inclusion. □

Remark. *A skilled chess player who immediately rejects a candidate move often cannot, at the moment of rejection, produce the rule that justifies the rejection — the rejection is enacted before, and sometimes instead of, being represented. This is not a failure of introspection to be corrected with more effort; on the reading this essay defends, it is evidence that F was never accompanied by an $R(F)$ in the first place. A rule that* does *exist — "drive on the right side of the road" — is not itself the operator; it is a partial symbolic gloss on a distributed behavioral pattern that the operator actually is. This suggests Rule does not sit at the same ontological level as F; it is better read as metadata about F; produced by R when R happens to be defined, and absent otherwise, without any corresponding gap in F's enactability.*

2 Compression and Generation

Having fixed what F is independently of C, the existence and structure of generating histories can be stated as genuine claims rather than tautologies. This section borrows its formal apparatus from

algorithmic information theory, treating a history H as a generating program and F as the function it computes.

Proposition 2.1 (Existence of Generating Histories)**.** *For any operator F specified by a computable input-output behavior, there exists at least one history H (in the sense of a generating program on some fixed universal machine U) such that $C(H) = F$, where C here denotes U's evaluation map.*

Proof. Immediate: the program that exhaustively lists F's input-output pairs, or that directly encodes F's truth table where F has finite domain, is itself such an H. Existence is not the interesting claim; what is generated is not constrained to be short, structured, or in any way resembling an actual learning trajectory. The interesting claims follow. □

Definition 2.2 (Compression Equivalence Class)**.** *For a fixed operator F, let*

$$[H]_C \;=\; \{H' : C(H') = F\}$$

denote the class of all histories that compress to F.

Proposition 2.3 (Non-Uniqueness of Generating Histories)**.** *For any F with $|\,[H]_C\,| \geq 2$ witnessed by at least two generating programs of different lengths on U (which holds for every F not already at its own minimal description), $[H]_C$ is infinite.*

Proof. Standard padding argument: given any $H \in [H]_C$, a program that ignores an arbitrary suffix and then runs H computes the same F, so $[H]_C$ is closed under an operation with infinitely many distinct outputs. □

Remark. *This is the formal content behind a claim that otherwise risks sounding merely suggestive: multiple learning trajectories can produce the same expertise; multiple institutional histories can produce the same law; multiple independent proofs can establish the same theorem. The operator does not determine its own origin. Observing F alone — watching the expert act, reading the statute, checking the theorem holds — never licenses an inference back to a unique generating H.*

Definition 2.4 (Minimal Generating History)**.** *$H^* \in [H]_C$ is minimal if $|H^*| \leq |H'|$ for all $H' \in [H]_C$, where $|\cdot|$ denotes description length on U. Write $K(F) := |H^*|$, the Kolmogorov complexity of F.*

Proposition 2.5 (Existence, Near-Uniqueness, and Uncomputability of the Minimal Generator)**.** *H^* exists for every F (the infimum of a nonempty set of positive integers is attained). $K(F)$ is well-defined up to an additive constant independent of F (the Invariance Theorem: for any two universal machines U_1, U_2, $|K_{U_1}(F) - K_{U_2}(F)| \leq c_{U_1,U_2}$). No algorithm exists that computes H^*, or even $K(F)$, for arbitrary F.*

Proof. Existence and invariance are the standard results of algorithmic information theory; see Li and Vitányi. Uncomputability is the standard incomputability theorem for Kolmogorov complexity: were K computable, one could effectively search, for each n, for the first string x (in a fixed enumeration) with $K(x) > n$, producing a description of x of length $O(\log n)$ — far shorter than n for large n — and thereby a contradiction, since x was chosen to have no description shorter than n. No such effective search can exist; hence K, and with it H^*, is not computable. □

Remark. *This is the essay's second load-bearing borrowed result, and it should be read plainly: there is a fact of the matter about how compressed an operator's most efficient generating history could be, that fact is well-defined independent of the choice of underlying machine up to a fixed constant, and no procedure — not introspection, not observation, not analysis of F itself — can reliably recover it. An expert's most compressed possible representation of their own skill is not merely private; on this reading it is not computable by any process, including the expert's own subsequent reflection on their expertise.*

3 Retention, Repair Entropy, and the Anti-Alignment Theorem

Sections 1 and 2 establish that generating histories exist, are non-unique, and that the minimal one is uncomputable. None of this yet says why an agent — a person, an institution, a software system — should ever prefer anything other than the shortest available generator, since shorter generally means faster and cheaper to store and to run. This section supplies the missing consideration: repairability, and shows that it stands in direct, provable tension with minimality.

Definition 3.1 (Retained Trace)**.** *Let H be a generating history for F and let $T_{\min} \subseteq H$ be a minimal sub-history such that $C(T_{\min}) = F$ still holds — the smallest part of H actually necessary to reproduce F's behavior. A* retained trace *is any T with $T_{\min} \subseteq T \subseteq H$: $T_{\min}$ together with some further surplus of H kept on hand but not needed for F to run.*

Remark. *This is deliberately not the same object as H^* of Proposition 2.5. H^* is the shortest generator over* all *possible programs on U, unconstrained by any relationship to the actual history that produced F in a given case; $T_{\min}$ is the shortest sub-history of the* particular *H that occurred, sufficient to run F. The* `howtopdf` *macro's $T_{\min}$ is whatever minimal internal state actually executes when the hotstring fires; H^* might in principle be a shorter, entirely different program computing the same input-output behavior with no relation to the 'convert'/'ocrmypdf' history at all. This essay's central result concerns $T_{\min}$ against fuller retention of the same H, not H^* against everything else.*

Definition 3.2 (Repair Entropy of a Retained Trace)**.** *Let s be a symptom: an observation that F is misbehaving. Let $\mathcal{H}_{\text{fault}}(F, T)$ denote the set of internal fault locations consistent with both s and every fact recorded in T. Define*

$$S_R(F \mid T) \;=\; \log \left|\mathcal{H}_{\text{fault}}(F, T)\right|,$$

adapting the repair entropy of Recursive Continuation*, §10.3, to the retained trace T rather than to an unspecified diagnostic record.*

Remark. Recursive Continuation *uses the bare symbol H for the set of candidate fault histories itself ($S_R = \log |H|$ there). That H and this essay's generating history H are unrelated objects sharing a symbol by coincidence of prior notation; to avoid collision this essay writes $\mathcal{H}_{\text{fault}}(F, T)$ throughout for the fault-location set, reserving bare H exclusively for the generating history of §1–2.*

Lemma 3.3 (Trace-Retention Monotonicity)**.** *If $T_1 \subseteq T_2$, then $S_R(F \mid T_2) \leq S_R(F \mid T_1)$.*

Proof. Consistency with the larger fact set T_2 implies consistency with the smaller fact set T_1 (every fault location ruled admissible under more constraints was already admissible under fewer), so $\mathcal{H}_{\text{fault}}(F, T_2) \subseteq \mathcal{H}_{\text{fault}}(F, T_1)$, and the claim follows by monotonicity of $\log |\cdot|$. □

Definition 3.4 (Feasible Retention Lattice)**.** $\mathcal{L}_F = \{T : T_{\min} \subseteq T \subseteq H\}$*, ordered by inclusion.*

Principle 3.5 (Compression–Repair Tradeoff)**.** *For a fixed operator F and its feasible retention lattice $\mathcal{L}_F$, any reduction in retained trace sufficient to strictly decrease storage cost $|T|$ weakly increases repair entropy $S_R(F \mid T)$. Compression does not merely trade storage for efficiency; it trades enactability against repairability.*

Remark. *This is the paper's philosophical center, and Lemma 3.3 is already its proof: $T_1 \subsetneq T_2$ strictly decreases stored surplus while weakly increasing S_R, by monotonicity of* $\log |\cdot|$ *under set inclusion. The Anti-Alignment Theorem below is this principle evaluated at its extreme point — the bottom of the lattice — where the tradeoff is forced to be as unfavorable to repair as the lattice permits.*

Theorem 3.6 (Anti-Alignment Theorem)**.** *Over $\mathcal{L}_F$, T_{min} minimizes surplus retention $|T \setminus T_{\text{min}}|$ and weakly maximizes repair entropy:*

$$S_R(F \mid T) \;\leq\; S_R(F \mid T_{\text{min}}) \qquad \textit{for all } T \in \mathcal{L}_F.$$

If there exists $T \in \mathcal{L}_F$ such that some fact recorded in $T \setminus T_{\text{min}}$ excludes at least one fault location consistent with s and T_{min} alone, then the inequality is strict for that T: $S_R(F \mid T) < S_R(F \mid T_{\text{min}})$.

Proof. $T_{\text{min}} \subseteq T$ for every $T \in \mathcal{L}_F$ by definition of the lattice, so $|T \setminus T_{\text{min}}| \geq 0$ with equality only at $T = T_{\text{min}}$, giving the minimality claim. The weak inequality on S_R is Lemma 3.3 applied with $T_1 = T_{\text{min}}$. For strictness: if some fact in $T \setminus T_{\text{min}}$ excludes a fault location $h \in \mathcal{H}_{\text{fault}}(F, T_{\text{min}})$, then $h \notin \mathcal{H}_{\text{fault}}(F, T)$, so $\mathcal{H}_{\text{fault}}(F, T)$ is a proper subset of $\mathcal{H}_{\text{fault}}(F, T_{\text{min}})$ and $S_R(F \mid T) < S_R(F \mid T_{\text{min}})$ follows. Equality persists exactly when no fact in the surplus $T \setminus T_{\text{min}}$ is diagnostically relevant to the symptom s under consideration — the surplus is true of the system but inert for this particular failure mode. □

Remark (The theorem is order-theoretic, not compression-specific)**.** *Nothing in the proof of Theorem 3.6 uses any property of C beyond having generated the lattice $\mathcal{L}_F$ in the first place. The result is really a statement about the bottom element of any lattice ordered by inclusion on which $S_R(F \mid \cdot)$ is antitone: the bottom minimizes whatever cost is monotone in trace size and maximizes whatever quantity is antitone in it. Compression is what supplies $\mathcal{L}_F$ with a bottom element worth naming T_{min}; the tradeoff itself would hold for any process generating a comparable retention lattice, which suggests the theorem is more general than the operators-and-expertise setting used to motivate it here.*

Remark (Reading the theorem)**.** *T_{min} — the minimal trace needed to* enact *F at all — is, by construction, never a better choice for* repairing *F than any fuller retention of the same history, and is strictly worse whenever the discarded surplus contained information that would have ruled out some way F could break. Minimal enactment and minimal repair entropy are not merely different quantities; they sit at opposite ends of the same lattice. The Kolmogorov-minimal generator of §2, being minimal in an even stronger sense, inherits this at least as badly: nothing about achieving the shortest possible description of F gives any reason to expect it retains diagnostic structure, and generically it will not, since diagnostic structure is exactly the kind of internal, redundant detail that compression is designed to discard.*

3.1 The `howtopdf` Example, Formalized

Return to §1's macro. Let T_{min} be whatever the hotstring literally expands to and executes — the operative command sequence, nothing more. Suppose OCR output degrades after a system update. If the retained trace is T_{min} alone, $\mathcal{H}_{\text{fault}}(F, T_{\text{min}})$ includes every plausible cause consistent with "OCR got worse" — a changed OCR engine version, a changed image preprocessing default, a changed file permission, a changed locale setting — because T_{min} records only that conversion-then-OCR happens, not the reasoning, the alternatives considered, or the intermediate outputs previously checked by hand. If instead the retained trace $T \supsetneq T_{\text{min}}$ includes the original discovery that orientation correction was necessary before OCR, together with a note that a particular OCR version had been validated against a particular image quality, several members of $\mathcal{H}_{\text{fault}}(F, T_{\text{min}})$ are directly excluded — Lemma 3.3's inequality is strict here, concretely, because the retained fact about validated OCR version rules out at least one otherwise-live hypothesis. The additional lines were never needed to make `howtopdf` run. They are needed only if `howtopdf` ever needs to be fixed.

3.2 The Regularized Objective

Definition 3.7 (Retention Objective)**.**

$$\min_{T \in \mathcal{L}_F} \Big(|T \setminus T_{\min}| \; + \; \lambda \, S_R(F \mid T) \Big), \qquad \lambda \geq 0.$$

Remark. *At $\lambda = 0$ the objective is minimized at $T = T_{\min}$: maximally efficient, and by Theorem 3.6, maximally opaque to repair. As $\lambda \to \infty$ the objective is dominated by minimizing S_R, driving the optimum toward $T = H$: full retention, maximal inspectability, no compression benefit realized at all. Neither extreme is where working expertise, well-designed software tooling, or well-functioning institutional procedure actually sit; both are recognizable failure modes — brittle, unrepairable compression at one end, and the novice's fully explicit, never-compressed trace at the other, which is fast to diagnose and too slow to ever be called expertise. Where the interior optimum falls, for a given F and a given cost of storage against a given distribution over future symptoms s, is not derived here and is left as an open problem; what this essay establishes is only that the trade-off is real, monotone, and in the strict case unavoidable — not that any particular λ is correct for any particular domain.*

4 Relation to *Recursive Continuation*

The history H compressed by C in this essay is the same object as H_t in *Recursive Continuation*, §8: "whatever record of prior states... a system retains." That essay left the dynamics of H_t as its first open problem, noting only that repair depends on H_t being recoverable without developing how H_t degrades, is selectively retained, or is reconstructed from partial traces. $C(H_t) = F_t$ is one concrete answer to the retention question: H_t is not simply kept or discarded wholesale; it is folded into a compressed, executable operator, and Theorem 3.6 says precisely what is lost when that folding goes all the way to $T_{\min}$ rather than stopping short of it.

This also gives a second reading of the Level-3 recursion at the center of that essay's Definition 2.1, $F_{t+1} = G(F_t, x_t, E_t)$: F_t there is the special case of this essay's operator F in which the compressed object's further role is to govern additional recursion, rather than to sit still as a finished skill or procedure. Recursive continuation, on this reading, is what compression looks like when the compressed operator is itself a rule for producing more history rather than a terminal action.

Open Problems

1. **$T_{\min}$ versus T_{access}.** This essay treats $T_{\min}$ as the trace needed by the *system* to enact F, but a further, distinct quantity — what is actually introspectively accessible to the *agent* attempting repair — may be smaller still. An expert's effective repair entropy is plausibly bounded by what they can access, not by what is stored somewhere in principle; formalizing this second retention gap is left open.

2. **Locating the interior optimum.** The regularized objective of §3.2 is stated but not solved for any particular cost structure on storage or distribution over symptoms. A treatment parallel to *Threshold of Deferral*'s crossover analysis, giving an explicit optimal T^* as a function of λ and the symptom distribution, is a natural next step.

3. **Higher-order compression.** Operators themselves can be compressed further into meta-operators — a collection of related macros becoming a workflow language, a body of case law

becoming a doctrine. Whether the Anti-Alignment Theorem iterates cleanly under repeated compression, or degrades in some structured way, is not addressed here.

4. **Empirical estimation.** As with $\bar{\rho}$ and κ_M, κ_R in this essay's companion pieces, $S_R(F \mid T)$ and the storage cost of surplus retention are easier to define than to measure in an actual expert, codebase, or institution.

References

1. Andrey N. Kolmogorov. "Three Approaches to the Quantitative Definition of Information." *Problems of Information Transmission*, 1(1), 1965.
2. Ray J. Solomonoff. "A Formal Theory of Inductive Inference, Part I." *Information and Control*, 7(1), 1964.
3. Gregory J. Chaitin. "On the Length of Programs for Computing Finite Binary Sequences." *Journal of the ACM*, 13(4), 1966.
4. Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, 4th ed. Springer, 2019.
5. Michael Polanyi. *The Tacit Dimension*. University of Chicago Press, 1966.
6. Hubert L. Dreyfus and Stuart E. Dreyfus. *Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. Free Press, 1986.
7. William G. Chase and Herbert A. Simon. "Perception in Chess." *Cognitive Psychology*, 4(1), 1973.
8. K. Anders Ericsson, Ralf Th. Krampe, and Clemens Tesch-Römer. "The Role of Deliberate Practice in the Acquisition of Expert Performance." *Psychological Review*, 100(3), 1993.
9. David K. Lewis. *Convention: A Philosophical Study*. Harvard University Press, 1969.
10. Flyxion. *Recursive Continuation: Autopoiesis, Sympoiesis, and the Compression of Self-Modification*. 2026.