Universal Catastrophic Safety, Undecidability, and Capability–Risk Frontier in Computational Universe

Haobo Ma[1] Wenlin Zhang[2]

[1]Independent Researcher

[2]National University of Singapore

November 24, 2025

Abstract

In previous axiomatic and geometric series works on "computational universe" $U_{\text{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$, we have constructed discrete complexity geometry, discrete information geometry, control manifold $(\mathcal{M}, G)$ induced by unified time scale, and proposed time–information–complexity joint variational principle on joint manifold $\mathcal{E}_Q = \mathcal{M} \times \mathcal{S}_Q$, while proving equivalence between physical universe category and reversible QCA computational universe category under unified time scale. However, essential limitations regarding "catastrophic safety" and "capability–risk frontier" still lack unified computational–geometric–logical framework.

This paper proposes within computational universe framework a "universal catastrophic safety" theory, connecting it with undecidability and geometric structure of capability–risk frontier. We first formalize catastrophic safety as path property: given catastrophe set $C_{\text{cat}} \subset X$, so-called "universal catastrophic safety" means universe evolution paths starting from all allowed initial states never enter C_{cat}. Under this setting, we define **universal catastrophic safety decision problem**, and prove at computational universe level: this decision problem is undecidable in most general case, i.e., there exists no algorithm that can give correct "forever safe/possibly catastrophic" verdict for all computational universes and catastrophe specifications.

Second, we model catastrophic safety and capability–risk duality as two types of functionals on computational universe: capability functional Cap evaluates success probability or performance of certain tasks, risk functional Risk evaluates probability or expected loss of reaching catastrophe set C_{cat}. We define **capability–risk frontier** as Pareto boundary of all realizable strategy $(\text{Cap}, \text{Risk})$ pairs under given computational universe and task set, and under constraints of unified time scale and complexity geometry, characterize this frontier as class of "reachable region boundary" on control manifold $(\mathcal{M}, G)$ and strategy space.

We further prove several key results: (1) Universal catastrophic safety verification problem in computational universe is at least as hard as halting problem, thus undecidable; (2) Any algorithmic safety filter attempting to be "correct for all strategies", if required to terminate and give verdict for all strategies under unified time scale, necessarily produces unavoidable "false negative/false positive regions" on capability–risk plane; (3) Under unified time scale, geometric optimization problem of capability enhancement and risk control can be written as constrained variational problem on joint manifold, where safety constraints naturally form non-recursively separable reachable region, thus capability–risk frontier cannot be algorithmically completely computed in general case.

Finally, we connect undecidability of catastrophic safety with previous topological complexity and causal diamond structures: within causal diamond, catastrophe conditions can be viewed as local boundary conditions, but when diamond scale tends to infinity, "whether there exists some path violating catastrophic safety" corresponds to problem of whether certain class of closed loops on configuration complex $\mathcal{X}$ are contractible, thereby inheriting previously established topological undecidability results. This paper provides systematic foundation for subsequent construction of "geometric shape of capability–risk frontier", "catastrophic safety consensus geometry of multi-agent systems", and "safety–capability–undecidability triangle relationship under unified time scale".

Keywords: Computational universe; Catastrophic safety; Undecidability; Capability–risk frontier; Halting problem; Control manifold; Causal diamond

1 Introduction

In design and analysis of large complex systems (including advanced artificial intelligence systems, financial systems, nuclear facilities, etc.), catastrophic safety is one of core constraints: we hope system possesses high capability (i.e., excellent performance on target tasks), while catastrophic risk is extremely low (e.g., not triggering large-scale irreversible damage). Traditional safety engineering mostly conducted in specific models, such as formal verification on bounded state spaces, model checking, or static analysis; while traditional computation theory reveals undecidability of "program property decision" through halting problem, Rice's theorem, etc.

In "computational universe" framework, entire universe abstracted as discrete system

$$U_{\text{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I}),$$

where X is configuration set, T is local one-step update, C is single-step cost under unified time scale, I characterizes task information quality. Within this framework, any specific engineering system, agent, or distributed protocol can be viewed as some subprocess of U_{comp} or local evolution of causal diamond. Previous works in this series have established:

- Complexity distance d_{comp}, volume growth $V_{x_0}(T)$, and discrete Ricci curvature $\kappa(x, y)$;
- Control manifold $(\mathcal{M}, G)$ induced by unified time scale and geodesic distance d_G;
- Task information manifold $(\mathcal{S}_Q, g_Q)$ and information distance;
- Time–information–complexity joint variational principle;
- Equivalence of physical universe and computational universe categories;
- Topological characterization of topological complexity, self-referential loops, and undecidability.

Goal of this paper is to, on this foundation, unify catastrophic safety and capability–risk duality into language of "computational universe", and give systematic answers to following questions:

1. How to formalize "universal catastrophic safety" in computational universe?
2. What are limits of its decision problem at logical and computability levels?
3. How does geometric structure of capability enhancement and risk control manifest in control manifold and joint variational framework?
4. How is "non-algorithmic solvability" of capability–risk frontier derived from undecidability and topological complexity?

We will see that catastrophic safety is undecidable in most general case, capability–risk frontier cannot be algorithmically completely computed under unified time scale, and any practical safety mechanism must accept certain "incompleteness": either rejecting some originally safe and high-capability strategies (false negatives), or unable to prove exclusion of all catastrophic risks (unavoidability of false positives).

Paper structure as follows: Section 2 formalizes catastrophic safety and capability–risk duality in computational universe. Section 3 gives undecidability proof of universal catastrophic safety decision problem. Section 4 constructs geometric characterization of capability–risk frontier, and analyzes limits of algorithmic search for this frontier. Section 5 connects catastrophic safety with causal diamonds and topological complexity. Appendices give detailed formalizations and proofs of main theorems.

2 Catastrophe, Safety, and Capability–Risk Duality in Computational Universe

This section formalizes catastrophe, safety specification, and capability–risk functionals on computational universe objects.

2.1 Review of Computational Universe and Evolution Paths

Consider computational universe object

$$U_{\mathrm{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I}),$$

satisfying previous axioms: X countable, $\mathsf{T} \subset X \times X$ local with finite degree, C single-step cost positive and path-additive, I task-related information quality function.

For any initial state $x_0 \in X$, an (infinite) evolution path is sequence

$$\Gamma = (x_0, x_1, x_2, \dots), \quad (x_k, x_{k+1}) \in \mathsf{T}.$$

If considering unified time scale, then for each step (x_k, x_{k+1}) accumulate cost

$$\mathsf{C}(\Gamma|_{[0,n]}) = \sum_{k=0}^{n-1} \mathsf{C}(x_k, x_{k+1}),$$

viewable as physical time up to step n.

2.2 Catastrophe Set and Catastrophe Specification

Definition 2.1 (Catastrophe Set)**.** Catastrophe set $C_{\text{cat}} \subset X$ is subset of configuration space, representing "once universe configuration enters it, viewed as catastrophe occurred" states. Specific examples include: system unrecoverable fault states, global irreversible damage states, states violating hard constraints, etc.

In many cases, catastrophe set itself is defining result of some property, not directly given explicit set. We allow C_{cat} described by predicate

$$\mathsf{Cat} : X \to \{\text{true}, \text{false}\}, \quad C_{\text{cat}} = \{x \in X : \mathsf{Cat}(x) = \text{true}\}$$

This predicate can be operator property (e.g., "some operator spectral radius exceeds threshold"), information property (e.g., "information leaked to sensitive subsystem"), or combinatorial property.

Definition 2.2 (Catastrophe Specification)**.** Catastrophe specification is pair

$$\mathcal{N}_{\text{cat}} = (X_0, C_{\text{cat}}),$$

where $X_0 \subset X$ is allowed initial state set (e.g., acceptable pre-deployment state space), $C_{\text{cat}} \subset X$ is catastrophe set.

We will consider reachability of all evolution paths starting from X_0 to catastrophe set.

2.3 Universal Catastrophic Safety

Definition 2.3 (Universal Catastrophic Safety)**.** Given computational universe U_{comp} and catastrophe specification $\mathcal{N}_{\text{cat}} = (X_0, C_{\text{cat}})$, call $(U_{\text{comp}}, \mathcal{N}_{\text{cat}})$ universally catastrophically safe, if for any initial state $x_0 \in X_0$ and any evolution path $\Gamma = (x_0, x_1, \dots)$ satisfying $(x_k, x_{k+1}) \in \mathsf{T}$, we have

$$\forall k \geq 0, \quad x_k \notin C_{\text{cat}}.$$

Otherwise call there exists catastrophic path, i.e., there exists some path entering C_{cat} in finite-step time.

This property is path-level "never touch" property, typical safety attribute.

2.4 Capability and Risk Functionals

Under unified time scale and task information geometry, we define capability and risk as two dual functionals on evolution paths.

Let task Q be represented by some goal set $G_Q \subset X$ or goal function $U_Q : X \to \mathbb{R}$.

Definition 2.4 (Capability Functional)**.** For given strategy or control rule π (abstracted as mechanism selecting next-step update from local information at each step), let $\mathbb{P}^{\pi}_{x_0}$ represent path distribution starting from initial state x_0. Capability functional defined as

$$\text{Cap}(\pi) = \inf_{x_0 \in X_0} \mathbb{E}_{\Gamma \sim \mathbb{P}^{\pi}_{x_0}} \big[U_Q(\Gamma) \big],$$

where $U_Q(\Gamma)$ can be terminal reward, cumulative reward, or some function of information quality. For example, for decision tasks, can take $U_Q(\Gamma)$ as "decision correct" indicator.

Definition 2.5 (Risk Functional)**.** For same strategy π, risk functional is

$$\mathrm{Risk}(\pi) = \sup_{x_0 \in X_0} \mathbb{P}_{\Gamma \sim \mathbb{P}^{\pi}_{x_0}} \big[\exists k \geq 0,\ x_k \in C_{\mathrm{cat}}\big].$$

High capability means excellent performance on tasks, low risk means catastrophe set difficult to touch. Extreme universal catastrophic safety corresponds to $\mathrm{Risk}(\pi) = 0$ and system essentially safe.

Under unified time scale, we can also consider capability and risk conditioned within time budget T, e.g.,

$$\mathrm{Risk}_T(\pi) = \sup_{x_0 \in X_0} \mathbb{P}\big[\exists k,\ \mathsf{C}(\Gamma|_{[0,k]}) \leq T,\ x_k \in C_{\mathrm{cat}}\big].$$

This paper mainly focuses on conceptual structure under infinite time perspective.

3 Undecidability of Universal Catastrophic Safety Decision

This section defines universal catastrophic safety decision problem, and proves its undecidability at computational universe level.

3.1 Universal Catastrophic Safety Decision Problem

Problem 3.1 (Universal Catastrophic Safety Decision)**. Input:** (1) Finite description of computational universe $U_{\mathrm{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$ (e.g., given by finite state transition rules or QCA rules); (2) Finite description of catastrophe specification $\mathcal{N}_{\mathrm{cat}} = (X_0, C_{\mathrm{cat}})$ (e.g., given by predicate or automaton).

Output: Decide whether $(U_{\mathrm{comp}}, \mathcal{N}_{\mathrm{cat}})$ is universally catastrophically safe.

We will consider whether such decision process has global algorithm: for all inputs giving correct Yes/No answer in finite time.

3.2 Reduction from Halting Problem to Catastrophic Safety

Standard statement of halting problem is: given program–input pair (P, w), decide whether program P halts in finite steps on input w. We know this problem is undecidable.

Within computational universe framework, we can embed simulation of universal Turing machine or universal CA/QCA into configuration graph. Below we construct reduction from halting problem to universal catastrophic safety decision.

Construction Idea

Given (P, w), construct following computational universe and catastrophe specification:

1. Let basic computational universe $U^{\mathrm{TM}}_{\mathrm{comp}}$ simulate universal Turing machine, whose configuration space X contains "machine state + tape content" encoding.

2. For given (P, w), define initial state set $X_0 = \{x_{\text{init}}(P, w)\}$, i.e., unique initial state is machine's initial configuration under program P and input w.

3. Define catastrophe set C_{cat} as special marking state set reached after simulation halting state reached then passing through fixed-length update. For example:

    - When Turing machine halts, enter halting state q_{halt};
    - Then through finite-step transition enter marking state $x_{\text{bad}} \in C_{\text{cat}}$;
    - If Turing machine never halts, then path never enters C_{cat}.

Under this construction, have:

- If $P(w)$ halts, then there exists path starting from $x_{\text{init}}(P, w)$ entering C_{cat} in finite steps, thus $(U_{\text{comp}}^{\text{TM}}, \mathcal{N}_{\text{cat}}^{(P,w)})$ not universally catastrophically safe;
- If $P(w)$ does not halt, then for all paths never enter C_{cat} (assuming computational universe has no external noise perturbation), therefore $(U_{\text{comp}}^{\text{TM}}, \mathcal{N}_{\text{cat}}^{(P,w)})$ universally catastrophically safe.

Thus halting problem reducible to universal catastrophic safety decision.

3.3 Undecidability Theorem

Theorem 3.2 (Undecidability of Universal Catastrophic Safety)**.** *There does not exist global algorithm* SafeDecide*, for all computational universe finite descriptions U_{comp} and catastrophe specifications $\mathcal{N}_{\text{cat}}$ as inputs, always outputting correct decision value \{"universally catastrophically safe", "catastrophic path exists"\} in finite time.*

Proof (Outline). Assume there exists such algorithm SafeDecide. For any program–input pair (P, w), according to previous section construction construct $(U_{\text{comp}}^{\text{TM}}, \mathcal{N}_{\text{cat}}^{(P,w)})$. Run

$$\mathsf{SafeDecide}\big(U_{\text{comp}}^{\text{TM}}, \mathcal{N}_{\text{cat}}^{(P,w)}\big)$$

If outputs "universally catastrophically safe", then $P(w)$ does not halt; if outputs "catastrophic path exists", then $P(w)$ halts. Thus obtain decision algorithm for halting problem, contradiction.

Therefore assumption does not hold, universal catastrophic safety decision problem is undecidable.

Q.E.D. □

3.4 Hierarchy and Stronger Undecidability

Above proof shows universal catastrophic safety is at least equivalent to halting problem. If further considering randomness, interaction, and time-unbounded behaviors, corresponding "catastrophe possibility" can be encoded as certain operator or path hyperproperties, whose logical complexity can elevate to higher classes in arithmetic or analytical hierarchy. In such cases, universal catastrophic safety decision problem can even reach completeness of higher hierarchy classes.

This paper does not pursue precise hierarchy, only characterizes "undecidability" as fundamental obstacle to catastrophic safety verification.

4 Geometric Characterization and Non-Algorithmic Solvability of Capability–Risk Frontier

This section gives geometric characterization of capability–risk frontier under unified time scale and complexity geometry, and analyzes limits of its algorithmic solvability.

4.1 Strategy Space and Control Manifold

In previous control manifold $(\mathcal{M}, G)$ construction, each control parameter $\theta \in \mathcal{M}$ corresponds to some physically realizable control configuration or strategy prototype. In multi-step evolution, control path $\theta(t)$ corresponds to some dynamic strategy family. For simplification, we first abstract strategy space at discrete level as some set Π, each $\pi \in \Pi$ defines rule from local observation to next-step update, constrained by unified time scale and complexity budget.

Can further embed Π into some parameter submanifold $\mathcal{M}_\Pi \subset \mathcal{M}$ of control manifold, such that each strategy π corresponds to one or family of control paths. This paper conceptually does not distinguish Π from $\mathcal{M}_\Pi$.

4.2 Definition of Capability–Risk Frontier

Definition 4.1 (Capability–Risk Pair)**.** For each strategy $\pi \in \Pi$, define its capability–risk pair as

$$(\mathrm{Cap}(\pi), \mathrm{Risk}(\pi)) \in \mathbb{R} \times [0, 1].$$

Definition 4.2 (Realizable Capability–Risk Set)**.** Realizable capability–risk set is

$$\mathcal{R}_{\mathrm{CR}} = \{(\mathrm{Cap}(\pi), \mathrm{Risk}(\pi)) : \pi \in \Pi\} \subset \mathbb{R} \times [0, 1].$$

Definition 4.3 (Capability–Risk Frontier)**.** Capability–risk frontier $\mathcal{F}_{\mathrm{CR}} \subset \mathcal{R}_{\mathrm{CR}}$ is set of all Pareto optimal points:

$$(\mathrm{Cap}, \mathrm{Risk}) \in \mathcal{F}_{\mathrm{CR}}$$

if and only if there does not exist another strategy π' satisfying

$$\mathrm{Cap}(\pi') \geq \mathrm{Cap}, \quad \mathrm{Risk}(\pi') \leq \mathrm{Risk},$$

with at least one inequality strict.

Intuitively, points on frontier correspond to class of “capability–risk tradeoff” limits, any attempt to enhance capability or reduce risk must sacrifice other side.

4.3 Geometric Embedding of Frontier

On control manifold $(\mathcal{M}, G)$, we can represent strategies as points or path families, with capability and risk as two functionals

$$\mathrm{Cap} : \mathcal{M}_\Pi \to \mathbb{R}, \quad \mathrm{Risk} : \mathcal{M}_\Pi \to [0, 1].$$

Under unified time scale and variational principle, we can write "maximize capability under given risk constraint" as constrained optimization problem:

$$\max_{\pi\in\Pi} \mathrm{Cap}(\pi) \quad \text{subject to} \quad \mathrm{Risk}(\pi) \leq r_0.$$

Geometrically, this corresponds to solving extremal problem satisfying inequality constraint on $\mathcal{M}_\Pi$, whose Lagrangian function is

$$\mathcal{L}(\theta, \lambda) = -\mathrm{Cap}(\theta) + \lambda(\mathrm{Risk}(\theta) - r_0), \quad \lambda \geq 0.$$

Its extremal points satisfy

$$\nabla\mathrm{Cap}(\theta^*) = \lambda^*\nabla\mathrm{Risk}(\theta^*), \quad \mathrm{Risk}(\theta^*) = r_0,$$

this is standard first-order condition for geometrically "frontier" points. In multi-dimensional case, this condition characterizes normal structure of frontier on control manifold.

4.4 Logical Roots of Non-Algorithmic Solvability of Frontier

However, even though frontier appears geometrically benign, at computability level, "giving safe high-capability strategy on frontier" still cannot be algorithmically completed. Intuitive reason is: if there exists algorithm FrontierSearch capable of generating point π^* on frontier for any computational universe and catastrophe specification (e.g., high-capability strategy with risk below some threshold), then we can use it to indirectly solve universal catastrophic safety decision problem.

Theorem 4.4 (Non-Algorithmicity of Complete Frontier Solution)**.** *There does not exist global algorithm* FrontierSearch*, for all inputs* $(U_{\mathrm{comp}}, \mathcal{N}_{\mathrm{cat}}, Q)$ *outputting strategy* π *in finite time, satisfying:*

1. *π's capability on task Q reaches some fixed threshold* $\mathrm{Cap}(\pi) \geq c_0$ *(e.g., non-trivial capability);*

2. $\mathrm{Risk}(\pi) = 0$ *(universally catastrophically safe);*

3. *If there exists any universally catastrophically safe strategy with capability at least c_0, then* FrontierSearch *must output one of them.*

Proof (Outline). If FrontierSearch exists, then for previously constructed instance from halting problem $(U^{\mathrm{TM}}_{\mathrm{comp}}, \mathcal{N}^{(P,w)}_{\mathrm{cat}}, Q_0)$ (where task Q_0 can be "successfully simulate one program–input pair evolution"), have:

- If $P(w)$ does not halt, then system universally catastrophically safe, there exists "catastrophe-free strategy with non-trivial capability";

- If $P(w)$ halts, then any strategy reaching capability c_0 necessarily has non-zero catastrophic risk (because to simulate complete program, must trigger catastrophe marking).

Assuming FrontierSearch satisfies conditions, then

- In non-halting case, FrontierSearch must output some strategy with $\mathrm{Risk}(\pi) = 0,\ \mathrm{Cap}(\pi) \geq c_0$;
- In halting case, there does not exist strategy satisfying conditions, algorithm necessarily cannot output answer satisfying conditions (either does not terminate, or violates completeness).

By monitoring output behavior of FrontierSearch, we can decide whether $P(w)$ halts, thus contradiction. Therefore complete frontier search algorithm does not exist.

Q.E.D. □

This theorem shows: under most general computational universe setting, capability–risk frontier as global object cannot be algorithmically completely computed, any practical method can only give approximate frontier or conservative estimate within some restricted class.

5 Causal Diamonds, Topological Complexity, and Local Safety Verification

This section connects catastrophic safety with previously introduced causal diamonds, boundary computation, and topological complexity, discussing possibilities and limits of local safety verification.

5.1 Catastrophic Safety in Local Causal Diamonds

In previous causal diamond theory, we introduce for event layer $E = X \times \mathbb{N}$ complexity light cone and causal diamond

$$\Diamond(e_{\mathrm{in}}, e_{\mathrm{out}}; T) = J^+_T(e_{\mathrm{in}}) \cap J^-_T(e_{\mathrm{out}}),$$

whose internal evolution can be compressed-encoded by boundary operator $\mathsf{K}_\Diamond : \mathcal{B}^-_\Diamond \to \mathcal{B}^+_\Diamond$.

From catastrophic safety perspective, we more care about: whether there exists some path entering C_{cat} inside diamond. If diamond scale is finite, then this decision can in principle be completed through exhaustion or symbolic analysis (its complexity can be very high, but at least is finite process). This corresponds to **local safety verification**: verifying “local catastrophe unreachable” within finite time–space window.

5.2 Diamond Gluing and Global Undecidability

However overall catastrophic safety is not property of some single diamond, but joint property of all possible diamonds: i.e., whether there exists some $e_{\mathrm{in}}, e_{\mathrm{out}}, T$, such that paths inside diamond can reach C_{cat}. This equivalent to seeking on configuration complex some class of path systems containing catastrophe states, whose topological structure closely related to previous closed loop undecidability.

In previous topological complexity paper we proved: in general constructible computational universe families, deciding whether certain class of closed paths are contractible is undecidable. Encoding catastrophic safety as “whether there exists some closed path starting from initial state passing through catastrophe set then returning to some reference state", we can transform catastrophic safety decision problem into problem of whether certain class of closed loops exist/are contractible, thereby inheriting undecidability.

Therefore, can summarize as:

- **Local**: within single causal diamond, whether catastrophe is reachable can in principle be finitely verified;
- **Global**: whether there exists some diamond making catastrophe reachable, in general case cannot be algorithmically decided.

This shows safety verification in engineering practice naturally has "locality": we can only perform safety detection on system at finite time–space scales, global safety can only be indirectly approximated through iterating local detection, redundant design, and conservative assumptions.

6 Conclusion

This paper systematically discusses problems of universal catastrophic safety, undecidability, and capability–risk frontier under computational universe's unified time scale–complexity geometry–information geometry framework. By formalizing catastrophic safety as path-level safety property, we prove its global decision problem is undecidable; by viewing capability and risk as two functionals on control manifold, we give geometric characterization of capability–risk frontier, and prove there does not exist complete algorithm capable of finding all "safe high-capability" strategy families in general computational universe.

Furthermore, through causal diamond and topological complexity structures, we show tension between feasibility of local (finite diamond) safety verification and topological undecidability of global catastrophic safety. Discussion of complexity entropy and topological closed loops shows that under unified time scale, computational universe evolution obeys certain "second law of complexity": under appropriate coarse–graining compressible complexity monotonically non-decreasing, providing geometric–topological perspective for time arrow and safety challenges.

These results indicate that any engineering or governance scheme regarding catastrophic safety inevitably resides at "incompleteness frontier": safety verification cannot completely cover all strategies and scenarios, capability–risk frontier cannot be algorithmically exhausted. Subsequent work will combine multi-observer consensus geometry with social–multi-agent systems, providing further geometric–categorial characterization of "collective safety perception and decision-making".

A Undecidability of Universal Catastrophic Safety and Reduction Details

This appendix gives formalized reduction details from halting problem to universal catastrophic safety decision problem.

A.1 Construction of Turing Machine Simulation in Computational Universe

Let there be universal Turing machine M, whose state set is finite set, tape alphabet finite. We select in computational universe configuration space

$$X = Q \times \Gamma^{\mathbb{Z}} \times \mathbb{Z},$$

where Q is machine state set, Γ is tape alphabet, $\mathbb{Z}$ represents read head position. Single-step transition relation T corresponds to Turing machine's transition function, single-step cost $\mathsf{C} \equiv 1$. This makes $U^{\mathrm{TM}}_{\mathrm{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$ specific instance of previous axioms.

For program–input pair (P, w), construct initial state $x_{\mathrm{init}}(P, w)$ as "machine state is q_0, tape writes program encoding and input, read head position is 0" configuration. Let

$$X_0 = \{x_{\mathrm{init}}(P, w)\}.$$

Define halting state set

$$H = \{x \in X : \text{Turing machine state is halting state}\}.$$

Construct catastrophe set as

$$C_{\mathrm{cat}} = \{x_{\mathrm{bad}}\},$$

where x_{bad} is specific marking state reached through finite-step transition after halting state (e.g., writing special mark on tape and resetting machine state). This achievable by extending machine states and transition function.

A.2 Property Verification

- If $P(w)$ halts, then there exists n such that $x_n \in H$, subsequently in finite steps evolves to $x_{\mathrm{bad}} \in C_{\mathrm{cat}}$;
- If $P(w)$ does not halt, then machine state on all evolution paths never enters halting state, thus cannot possibly enter C_{cat}.

Thus:

- $P(w)$ halts $\Rightarrow (U^{\mathrm{TM}}_{\mathrm{comp}}, \mathcal{N}^{(P,w)}_{\mathrm{cat}})$ not universally catastrophically safe;
- $P(w)$ does not halt $\Rightarrow (U^{\mathrm{TM}}_{\mathrm{comp}}, \mathcal{N}^{(P,w)}_{\mathrm{cat}})$ universally catastrophically safe.

If there exists universal catastrophic safety decision algorithm SafeDecide, then for (P, w) can decide halting problem by constructing $\mathcal{N}^{(P,w)}_{\mathrm{cat}}$ and calling SafeDecide, contradiction.

B Formalization of Complexity Entropy Monotonicity

This appendix gives more concrete version and proof outline of Proposition 5.1.

B.1 Group Representation and Shortest Word Length

In configuration complex $\mathcal{X} = \mathcal{X}(U_{\text{comp}}, \mathcal{R})$, fundamental group $\pi_1(\mathcal{X}, x_*)$ can be represented by generators $\{g_i\}$ and relation set $\mathcal{R}$, closed path γ corresponds to group element $[\gamma]$, whose shortest word length $\ell_{\min}([\gamma])$ is defined as shortest word length expressible using generators and their inverses.

In computational universe, we can choose generators as “basic update edges”, relations as equivalences corresponding to local loops $\mathcal{R}$. Then compression complexity $K(\gamma)$ of closed path γ is equivalent to $\ell_{\min}([\gamma])$ within constant factor.

B.2 Semigroup Structure of Coarse–Graining Operations

Coarse–graining operations can be abstracted as family of transformations acting on group, whose effect is performing local substitutions on word representations of group elements, reducing high-frequency relation fragments. Can view these operations as semigroup $\mathcal{S}$ acting on representation space, whose each action does not change group element itself, only substitutes equivalent word representations.

Under such semigroup action, shortest word length $\ell_{\min}([\gamma])$ is invariant: regardless of substitution, shortest word length neither increases nor becomes smaller than this value through local simplifications.

If we consider coarse–graining time t only reflects “how many relation substitutions we have tried”, then as t increases, observed word length $\ell_t(\gamma)$ may decrease from some large value to $\ell_{\min}([\gamma])$ in initial stage, but once reaching this value, no longer decreases. Therefore function

$$\mathcal{C}(t) = \log \ell_{\min}([\gamma_t])$$

is monotonically non-decreasing from some moment, with limit value $\log \ell_{\min}([\gamma])$. In many rough models, we can ignore initial adjustment stage, understanding $\mathcal{C}(t)$ as “on macroscopic time scale, compression complexity does not spontaneously decrease”.

C Further Explanation of Geometric Complexity Classes and P Class Equivalence

Under standard model equivalence assumptions:

- Any polynomial-time Turing machine can be simulated using finite complexity resources in computational universe, with linear or polynomial rescaling between complexity distance and steps;
- Conversely, any process in computational universe with complexity radius $\mathcal{O}(n^k)$ can be translated to polynomial-time algorithm on Turing machine, by encoding paths as tape content and performing finite-state simulation on Turing machine.

Therefore geometric complexity class **GC**(poly) is abstractly equivalent to P class, only in geometric language replacing “steps” with “complexity distance” and “physical time under unified time scale”. This gives way to re-understand traditional complexity classes from “geometric universe perspective”.