

Axiomatic Structure of Computational Universes: Discrete Configurations, Update Relations, and the Unified Time Scale Framework

Haobo Ma¹

Wenlin Zhang²

¹Independent Researcher

²National University of Singapore

November 24, 2025

Abstract

Under the assumptions of finite information density and local reversible updates, we provide a unified axiomatic definition for the “computational universe.” The core object is a quadruple $U_{\text{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{l})$, where X is the discrete configuration space of the entire universe, $\mathsf{T} \subset X \times X$ is the one-step update relation, C is the single-step cost (time, energy, or gate count), and l is a state function characterizing “information quality.” We introduce axioms of locality, finite metricity, and (generalized) reversibility, proving that classical Turing machines, cellular automata, and reversible quantum cellular automata can all be embedded as special cases within this framework.

Furthermore, we prove that under the unified time scale hypothesis (i.e., the existence of a single-step cost function compatible with physical scattering time scales), the configuration graph $(X, \mathsf{T}, \mathsf{C})$ induces a “complexity geometry” in appropriate limits, whose geodesic distances are equivalent to a continuous version of traditional time complexity. Finally, we characterize relationships between different computational universes via simulation mappings, constructing a category **CompUniv** with computational universes as objects and structure-preserving simulations as morphisms, proving that the classical Turing universe, classical cellular automaton universe, and quantum cellular automaton universe form equivalent full subcategories within this category.

As the first work in the “Computational Universe Theory” series, this paper aims to provide a minimal discrete and physicalizable axiomatic foundation, establishing a unified benchmark structure for subsequent complexity geometry, information geometry, and the category equivalence between physical and computational universes.

Keywords: Computational Universe, Cellular Automata, Turing Machine, Quantum Cellular Automata, Unified Time Scale, Complexity Geometry, Simulation, Category Equivalence

MSC 2020: 68Q05, 68Q10, 81P68, 68Q12, 18D99

1 Introduction

“The universe is computation” represents a unified vision spanning physics, computer science, and information theory. If the entire universe is viewed as a vast discrete dynamical system, then traditional models such as classical Turing machines, cellular automata, and reversible quantum cellular automata can be understood as different slices of this “computational universe.” However, in existing literature, these models are often developed separately, rarely appearing within a unified axiomatic system, let alone establishing systematic connections with physical time scales, geometric structures, and category-theoretic “equivalences between universes.”

The goal of this paper is to construct such a foundational layer: under minimal assumptions, we provide an abstract “computational universe object” U_{comp} , characterizing its structure and constraints through a clear set of axioms to simultaneously encompass:

1. Classical Turing machines and their “Turing universes”;
2. Classical cellular automata and more general local discrete dynamical systems;
3. Reversible quantum cellular automata (QCA) and their universe models.

We emphasize two points:

- On one hand, the entire framework is **discrete**: universe states are modeled as points on a countable set X , and time evolution is realized by stepping on a graph (X, T) ;
- On the other hand, the cost function C is interpreted as discrete samples of a unified time scale, providing a bridge for subsequently viewing complexity as “geometric length.”

On this basis, we define “simulation morphisms” between different computational universes, constructing the category **CompUniv**. This not only unifies traditional “multi-model computability equivalence” results but also provides an abstract framework for subsequently establishing “equivalence between physical universe categories and computational universe categories.”

The structure of this paper is as follows. Section 2 provides basic notation and preliminaries. Section 3 presents the axiomatic definition of computational universe objects. Sections 4 and 5 respectively explain how Turing machines, classical cellular automata, and quantum cellular automata are embedded within this framework. Section 6 introduces the unified time scale and basic constructions of complexity geometry. Section 7 constructs simulation morphisms and the category **CompUniv**. The appendices provide detailed proofs of main propositions and theorems along with several technical discussions.

2 Preliminaries and Notation

All sets, mappings, and algebraic structures in this paper are discussed within the background of Zermelo–Fraenkel set theory with the axiom of choice. By default, all sets are at most countable unless otherwise stated.

1. Let $\mathbb{N} = \{0, 1, 2, \dots\}$, \mathbb{Z} be the set of integers, \mathbb{R} be the real field.
2. For a set X , let $\mathcal{P}_{\text{fin}}(X)$ denote the family of finite subsets of X .
3. If $G = (V, E)$ is a directed graph, then V is the vertex set and $E \subset V \times V$ is the directed edge set.
4. For a Hilbert space \mathcal{H} , let $\mathcal{B}(\mathcal{H})$ denote the algebra of bounded linear operators. If $U \in \mathcal{B}(\mathcal{H})$ satisfies $U^*U = UU^* = \text{id}$, then U is called unitary.
5. When unambiguous, denote the image of $f : A \rightarrow B$ as $f(A)$ and the preimage as $f^{-1}(C)$.

We are particularly concerned with locality and finite information density, which naturally appear in classical cellular automata and QCA. For uniformity, we adopt the following abstract setting.

Definition 2.1 (Local Structure). Let X be a countable set. A local structure is a finite-degree directed graph $G_X = (X, E_X)$, where for each $x \in X$,

$$\deg^+(x) = |\{y : (x, y) \in E_X\}| < \infty$$

$$\deg^-(x) = |\{y : (y, x) \in E_X\}| < \infty$$

Intuitively, G_X characterizes the finite-range adjacency relation of each configuration in “space.”

3 Axiomatization of Computational Universe Objects

This section presents the core object of this paper: the axiomatic definition of a computational universe U_{comp} .

3.1 Basic Data of Computational Universe

Definition 3.1 (Computational Universe Object). A computational universe object is a quadruple

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

where:

1. X is a countable set, called the configuration space of the universe;
2. $\mathsf{T} \subset X \times X$ is the one-step update relation;
3. $\mathsf{C} : X \times X \rightarrow [0, \infty]$ is the cost function;
4. $\mathsf{I} : X \rightarrow \mathbb{R}$ is the information quality function.

To view this as a “universe-scale computational system,” we impose the following axioms on the above data.

3.2 Axiom System

Axiom 1 (Finite Information Density). There exists a local structure $G_X = (X, E_X)$ such that for any finite vertex set $R \subset X$, the set of configurations adjacent to R ,

$$N(R) = \{x \in X : \exists y \in R, (x, y) \in E_X \text{ or } (y, x) \in E_X\}$$

satisfies $|N(R)| < \infty$.

Additionally, for each $x \in X$, the set of “internal states” locally relevant to x is also finite (ensured by encoding in concrete models).

Axiom 2 (Local Update). For any $x \in X$, the one-step reachable set

$$\mathsf{T}(x) = \{y \in X : (x, y) \in \mathsf{T}\}$$

is finite, and there exists a finite radius r (independent of x) such that the determination of $\mathsf{T}(x)$ depends only on the information in a neighborhood of radius r around x in G_X .

Axiom 3 (Generalized Reversibility). There exists a relation $\mathsf{T}^{-1} \subset X \times X$ such that for any $x \in X$,

$$\mathsf{T}^{-1}(x) = \{y : (y, x) \in \mathsf{T}\}$$

is finite, and when restricted to a “physically relevant” configuration subset $X_{\text{phys}} \subset X$, T and T^{-1} are mutually inverse function graphs on X_{phys} (i.e., time evolution is bijective).

Axiom 4 (Additivity and Positivity of Cost). For any $(x, y) \in \mathsf{T}$, we have $C(x, y) \in (0, \infty)$; if $(x, y) \notin \mathsf{T}$, define $C(x, y) = \infty$.

For any finite path $\gamma = (x_0, x_1, \dots, x_n)$, define

$$C(\gamma) = \sum_{k=0}^{n-1} C(x_k, x_{k+1})$$

Then $C(\gamma)$ depends only on T and C , and satisfies the triangle inequality for path concatenation.

Axiom 5 (Monotonicity of Information Quality). There exists a task family \mathcal{Q} (e.g., decision problems, function computation, or measurement tasks) such that for each task $Q \in \mathcal{Q}$, there exists an information quality function $I_Q : X \rightarrow \mathbb{R}$ satisfying: if a path γ supports computation for task Q , then the expected information quality along γ is non-decreasing; i.e., for typical paths $x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_n$,

$$\mathbb{E}[I_Q(x_{k+1})] \geq \mathbb{E}[I_Q(x_k)]$$

In most concrete cases, we can fix a single task (e.g., simulating a fixed external system) and omit the subscript Q , where I characterizes information proximity relative to some “true state” or target distribution.

3.3 Paths, Complexity, and Reachable Domains

Under the above axioms, we obtain the following natural definitions.

Definition 3.2 (Path and Complexity). In a computational universe U_{comp} , a path from x to y is a finite sequence $\gamma = (x_0, x_1, \dots, x_n)$ satisfying $x_0 = x$, $x_n = y$, and $(x_k, x_{k+1}) \in \mathsf{T}$ for all $0 \leq k < n$.

The cost of a path is

$$C(\gamma) = \sum_{k=0}^{n-1} C(x_k, x_{k+1})$$

Among all paths connecting x and y , define the distance

$$d(x, y) = \inf_{\gamma: x \rightarrow y} C(\gamma)$$

called the complexity distance from x to y .

Proposition 3.3. *Under Axioms A2 and A4, if for any $x, y \in X$ there exists at least one finite path connecting them, then d defines a generalized metric on X (possibly taking value ∞) satisfying:*

1. $d(x, x) = 0$;
2. $d(x, y) = d(y, x)$ if T is bijective on X_{phys} ;
3. $d(x, z) \leq d(x, y) + d(y, z)$.

The detailed proof is in Appendix A.1.

Definition 3.4 (Reachable Domain and Complexity Horizon). For a given initial configuration $x_0 \in X$ and resource budget $T > 0$, define the reachable domain

$$B_T(x_0) = \{x \in X : d(x_0, x) \leq T\}$$

If there exists $x^* \in X$ and constant T^* such that for all $T < T^*$, $x^* \notin B_T(x_0)$, while for all $T > T^*$, $x^* \in B_T(x_0)$, then T^* is called the complexity threshold from x_0 to x^* . More generally, topological transitions in the boundary of the reachable domain family $\{B_T(x_0)\}_{T>0}$ as a function of T characterize the ‘‘horizons’’ of complexity.

4 Embedding Classical Turing Machines and Cellular Automata

This section demonstrates that both classical Turing machines and cellular automata can be viewed as special cases of computational universe objects, thus being incorporated into the U_{comp} system.

4.1 Turing Machine Universe

Recall the definition of a classical deterministic Turing machine:

Definition 4.1 (Deterministic Turing Machine). A single-tape deterministic Turing machine is a 5-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0)$, where:

1. Q is the finite state set;

2. $\Sigma \subset \Gamma$ is the input alphabet, Γ is the tape symbol alphabet containing the blank symbol;
3. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$ is the transition function;
4. $q_0 \in Q$ is the initial state.

We encode the “global configuration” of a Turing machine run as a combination of the contents of a bi-infinite tape, head position, and current state.

Definition 4.2 (Configuration Space of Turing Machine Universe). Let \mathbb{Z} denote integer positions on the tape. Define the configuration space

$$X_M = Q \times \Gamma^{\mathbb{Z}} \times \mathbb{Z}$$

where a configuration $x = (q, (a_i)_{i \in \mathbb{Z}}, p)$ represents: the machine is in state q , the symbol at tape position i is a_i , and the head is at position p .

Define the one-step transition relation $\mathsf{T}_M \subset X_M \times X_M$ as: $(x, y) \in \mathsf{T}_M$ if and only if y is the configuration obtained by applying δ once at configuration x . Let the single-step cost be $\mathsf{C}_M(x, y) = 1$ if $(x, y) \in \mathsf{T}_M$, otherwise ∞ .

Let I_M be the decision correctness information relative to a given input and task (e.g., a 0–1 value or negative distance to target output).

Proposition 4.3. *For any deterministic Turing machine M , the quadruple*

$$U_{\text{comp}}(M) = (X_M, \mathsf{T}_M, \mathsf{C}_M, \mathsf{I}_M)$$

satisfies Axioms A1–A5, thus is a computational universe object.

Proof sketch: A1 is guaranteed by the local structure of X_M and finite tape alphabet; A2 by the locality of δ ; A3 holds on the subset of “physically reachable configurations” (i.e., trajectories actually traversed by the Turing machine and their reverse trajectories); A4 is evident from $\mathsf{C}_M \equiv 1$; A5 is ensured by monotonic design under task definitions (e.g., only reaching maximum information value at accepting configurations). See Appendix A.2 for details.

4.2 Classical Cellular Automaton Universe

Definition 4.4 (Classical Cellular Automaton). Let Λ be a countable lattice point set (e.g., \mathbb{Z}^d), S a finite state set. A cellular automaton is a local update rule $F : S^\Lambda \rightarrow S^\Lambda$, where there exists a finite neighborhood $\mathcal{N} \subset \Lambda$ and local rule $f : S^\mathcal{N} \rightarrow S$ such that

$$(F(c))_i = f((c)_{i+\mathcal{N}})$$

for all $i \in \Lambda$.

Define the configuration space $X_{\text{CA}} = S^\Lambda$, one-step transition relation $\mathsf{T}_{\text{CA}} = \{(c, F(c)) : c \in X_{\text{CA}}\}$, single-step cost $\mathsf{C}_{\text{CA}}(c, F(c)) = 1$, others ∞ . The information quality function I_{CA} is defined according to tasks.

Proposition 4.5. *For any classical cellular automaton F , the quadruple*

$$U_{\text{comp}}(F) = (X_{\text{CA}}, \mathsf{T}_{\text{CA}}, \mathsf{C}_{\text{CA}}, \mathsf{I}_{\text{CA}})$$

is a computational universe object.

A1–A2 come from locality and finite states; A3 strictly holds for reversible cellular automata, and for non-reversible cases can be handled through state space extension or subspace restriction; detailed discussion in Appendix A.3.

5 Embedding Reversible Quantum Cellular Automata

To incorporate quantum universe models into the same framework, we consider the abstract form of reversible QCA.

5.1 Basic Definition of QCA

Definition 5.1 (Reversible Quantum Cellular Automaton). Let Λ be a countable lattice point set. For each $i \in \Lambda$, assign a finite-dimensional local Hilbert space \mathcal{H}_i . The global Hilbert space is

$$\mathcal{H} = \bigotimes_{i \in \Lambda} \mathcal{H}_i$$

A reversible QCA is a unitary operator $U : \mathcal{H} \rightarrow \mathcal{H}$ satisfying:

1. Locality: For any bounded region $R \subset \Lambda$, there exists a finite expansion $R' \supset R$ such that $U^* \mathcal{A}(R) U \subset \mathcal{A}(R')$, where $\mathcal{A}(R)$ is the local operator algebra supported on R ;
2. Translation symmetry (optional): For all translations $\tau : \Lambda \rightarrow \Lambda$, U commutes with the corresponding translation operator.

To fit the discrete framework, we view the set of basis states of \mathcal{H} in a fixed orthonormal basis as the configuration set.

Definition 5.2 (Configuration and Update of QCA Universe). Choose an orthonormal basis $\{|s\rangle : s \in S_i\}$ for each \mathcal{H}_i . Let

$$X_{\text{QCA}} = \prod_{i \in \Lambda} S_i$$

be the set of all basis state labels.

For any $x \in X_{\text{QCA}}$, denote the corresponding basis vector as $|x\rangle$. Define the one-step relation $\mathsf{T}_{\text{QCA}} \subset X_{\text{QCA}} \times X_{\text{QCA}}$ as:

$$(x, y) \in \mathsf{T}_{\text{QCA}} \text{ if and only if } \langle y | U | x \rangle \neq 0$$

The single-step cost $C_{\text{QCA}}(x, y)$ is taken as a constant corresponding to the single-step physical implementation time of U or a weighted value depending on frequency. The information quality function I_{QCA} is defined according to the observation task of interest (e.g., classical post-processing of some measurement result).

5.2 Axioms Satisfied by QCA Universe

Proposition 5.3. *Under the assumptions of locality and finite-dimensional Hilbert space,*

$$U_{\text{comp}}(U) = (X_{\text{QCA}}, \mathsf{T}_{\text{QCA}}, C_{\text{QCA}}, I_{\text{QCA}})$$

satisfies Axioms A1–A5.

Key proof points:

- Finite information density comes from the finite dimension of each \mathcal{H}_i and locality;

- Finiteness of the one-step reachable set is given by U being a local linear combination;
- Inverse evolution is provided by U^* ;
- Positivity of single-step cost is guaranteed by the positivity of actual physical implementation time;
- Monotonicity of information quality can be proved via relative entropy functions in the Heisenberg picture.

Detailed arguments in Appendix A.4.

6 Unified Time Scale and Initial Construction of Complexity Geometry

While this paper focuses on discrete axioms, the unified time scale is the key bridge for subsequent “complexity geometry” and “physical-computational universe equivalence.” This section provides an initial structure: how to abstract a geometric distance compatible with physical time scales from the single-step cost C .

6.1 Consistency of Single-Step Cost and Time Scale

Assume there exists a physical scattering process whose frequency-resolved time scale density is $\kappa(\omega)$ (e.g., defined by scattering phase derivative, spectral shift function derivative, or group delay trace). We consider that the single-step cost $C(x, y)$ in the computational universe is a combination of several such basic physical processes, with implementation time cost writable as

$$C(x, y) = \int_{\Omega(x, y)} \kappa(\omega) d\mu_{x, y}(\omega)$$

where $\Omega(x, y)$ is the set of activated frequency bands in the corresponding physical process, and $\mu_{x, y}$ is the corresponding spectral measure. Thus, the path cost

$$C(\gamma) = \sum_k C(x_k, x_{k+1})$$

can be approximately viewed as discrete sampling of some continuous time integral, ultimately inducing a “complexity distance consistent with physical time scale” $d(x, y)$.

6.2 Complexity Geometry of Configuration Graph

Under Axioms A1–A4, we can view (X, T, C) as a weighted graph and consider its geometrization in certain limits.

Definition 6.1 (Complexity Graph). The complexity graph of a computational universe is a weighted directed graph $G_{\text{comp}} = (X, T, C)$, with edge weights C .

In some cases (e.g., when continuous control parameters exist and local rules approach continuous transformations), through Gromov–Hausdorff limits or spectral analysis of graph Laplacians, one can obtain a continuous manifold \mathcal{M} with metric G such that

shortest path distances on the graph converge to geodesic distances on the manifold at appropriate scales. This process constitutes the bridge from discrete complexity to continuous complexity geometry.

Proposition 6.2 (Continuous Limit of Graph Metric, Schematic). *Let $\{U_{\text{comp}}^{(h)}\}$ be a family of computational universes corresponding to complexity graphs $G_{\text{comp}}^{(h)}$, where “mesh size” $h \rightarrow 0$. If there exists a manifold \mathcal{M} with metric G such that $(X^{(h)}, d^{(h)})$ converges to (\mathcal{M}, d_G) in the Gromov–Hausdorff sense, then discrete complexity can be viewed at large scales as “time complexity” given by geodesic distance of G .*

This proposition is schematic; a precise version requires additional technical assumptions; see Appendix B.1 for discussion.

7 Simulation Morphisms and the Category CompUniv

To compare different computational universes, we introduce the concept of simulation morphisms.

7.1 Definition of Simulation Mapping

Definition 7.1 (Simulation Mapping). Let $U_{\text{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$, $U'_{\text{comp}} = (X', \mathsf{T}', \mathsf{C}', \mathsf{I}')$ be two computational universes. If there exists a map $f : X \rightarrow X'$ and constants $\alpha, \beta > 0$ such that:

1. Step preservation: If $(x, y) \in \mathsf{T}$, then $(f(x), f(y)) \in \mathsf{T}'$;
2. Cost control: For any path $\gamma : x \rightarrow y$, there exists $\gamma' : f(x) \rightarrow f(y)$ such that

$$\mathsf{C}'(\gamma') \leq \alpha \mathsf{C}(\gamma) + \beta$$

3. Information fidelity: There exists a monotone function $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ such that for all $x \in X$,

$$\mathsf{I}(x) \leq \Phi(\mathsf{I}'(f(x)))$$

then f is called a simulation mapping from U_{comp} to U'_{comp} , denoted $f : U_{\text{comp}} \rightsquigarrow U'_{\text{comp}}$.

If f is invertible on its image (there exists $g : X' \rightarrow X$ such that $g \circ f$ and $f \circ g$ are homotopic to identity on relevant subsets), and α, β are within acceptable complexity scaling ranges, then U_{comp} and U'_{comp} are called equivalent in complexity sense.

7.2 Category Structure

Proposition 7.2. *Taking computational universe objects as objects and simulation mappings as morphisms, we obtain a category **CompUniv**:*

1. For any U_{comp} , the identity map $\text{id}_X : X \rightarrow X$ is a simulation mapping;
2. If $f : U_{\text{comp}} \rightsquigarrow U'_{\text{comp}}$ and $g : U'_{\text{comp}} \rightsquigarrow U''_{\text{comp}}$ are simulation mappings, then the composite $g \circ f$ is also a simulation mapping;
3. Composition of simulation mappings satisfies associativity.

Proof in Appendix B.2.

We are particularly interested in the subcategories generated by Turing machines, cellular automata, and QCA.

Theorem 7.3 (Equivalence Between Classical Models). *Let \mathbf{TMUniv} , \mathbf{CAUniv} denote the full subcategories generated by Turing universes and classical reversible cellular automaton universes, respectively. Then:*

1. *These two subcategories are equivalent in $\mathbf{CompUniv}$;*
2. *In other words, there exist functors*

$$F_{\mathbf{TM} \rightarrow \mathbf{CA}} : \mathbf{TMUniv} \rightarrow \mathbf{CAUniv}$$

$$F_{\mathbf{CA} \rightarrow \mathbf{TM}} : \mathbf{CAUniv} \rightarrow \mathbf{TMUniv}$$

and natural isomorphisms η, ϵ such that

$$F_{\mathbf{CA} \rightarrow \mathbf{TM}} \circ F_{\mathbf{TM} \rightarrow \mathbf{CA}} \simeq \text{id}_{\mathbf{TMUniv}}$$

$$F_{\mathbf{TM} \rightarrow \mathbf{CA}} \circ F_{\mathbf{CA} \rightarrow \mathbf{TM}} \simeq \text{id}_{\mathbf{CAUniv}}$$

and these functors are realized by simulation mappings of polynomial complexity on morphisms.

The proof can be viewed as a categorical generalization of the classical “Turing machine–cellular automaton computability equivalence” result within this axiomatic framework; see Appendix B.3 for details.

Theorem 7.4 (Complexity Equivalence of QCA Universe and Classical Computational Universe). *There exists a full subcategory $\mathbf{QCAUniv} \subset \mathbf{CompUniv}$ generated by reversible QCA universes, with functors*

$$F_{\mathbf{TM} \rightarrow \mathbf{QCA}} : \mathbf{TMUniv} \rightarrow \mathbf{QCAUniv}$$

$$F_{\mathbf{QCA} \rightarrow \mathbf{TM}} : \mathbf{QCAUniv} \rightarrow \mathbf{TMUniv}$$

such that the above equivalence holds in the sense of computability and complexity.

This result unifies quantum computational models with classical computational universes within the same categorical structure. Its proof relies on known quantum universality results and reversible simulation constructions; see Appendix B.4 for details.

8 Conclusion and Outlook

This paper provides a minimal axiomatic definition of “computational universe”: modeling the entire universe as a discrete configuration space X with finite information density and local updates, a one-step update relation T , single-step cost C , and information quality function I . We prove that classical Turing machines, classical cellular automata, and reversible QCA can all be viewed as special computational universe objects, and are mutually equivalent full subcategories within our introduced simulation category $\mathbf{CompUniv}$. This result shows that under finite information and locality assumptions, different traditional models of “computational universe” are merely different presentations of the same abstract structure.

The unified time scale appears in this paper only in the form of cost function C and complexity graph, but the correspondence between it and physical scattering time scales is core to future work. In future work, we will further develop based on this paper’s axiomatic framework:

1. Construct complexity geometry and information geometry from $(X, \mathsf{T}, \mathsf{C})$, obtaining continuous manifolds (\mathcal{M}, G) , (\mathcal{S}, g) ;
2. Introduce observers, attention, and knowledge graphs into $\mathfrak{C} = (\mathcal{M}, G; \mathcal{S}, g; \mathcal{E}; \mathcal{A})$, constructing variational principles for “computational worldlines”;
3. Establish category equivalence between physical universe categories and computational universe categories under unified time scales and boundary time geometry, thereby achieving a structured expression of “computational universe = physical universe.”

A Proofs of Axiomatic Properties and Embedding Examples

A.1 Proof of Proposition 3.3

Proposition Restatement

Under Axioms A2 and A4, if for any $x, y \in X$ there exists at least one finite path connecting them, then $d(x, y) = \inf_{\gamma:x \rightarrow y} \mathsf{C}(\gamma)$ defines a generalized metric on X (possibly taking value ∞) satisfying:

1. $d(x, x) = 0$;
2. If T is bijective on X_{phys} , then $d(x, y) = d(y, x)$ for $x, y \in X_{\text{phys}}$;
3. $d(x, z) \leq d(x, y) + d(y, z)$.

Proof

1. For any x , take the zero-length path $\gamma = (x)$, conventionally $\mathsf{C}(\gamma) = 0$, so $d(x, x) \leq 0$. On the other hand, positivity of single-step cost in A4 guarantees any non-trivial path has positive cost, thus $d(x, x) = 0$.
2. On X_{phys} , if T is bijective, then for any $(x, y) \in \mathsf{T}$, there exists unique $(y, x) \in \mathsf{T}^{-1}$. If path $\gamma : x \rightarrow y$ achieves the infimum of $d(x, y)$, then by existence of reverse path $\gamma^{-1} : y \rightarrow x$ and symmetry of A4 (single-step cost can be viewed as symmetric on physical subset), we get $d(y, x) \leq \mathsf{C}(\gamma^{-1}) = \mathsf{C}(\gamma)$. The reverse inequality is similar, thus $d(x, y) = d(y, x)$.
3. Let $\varepsilon > 0$. By definition of $d(x, y)$, $d(y, z)$, there exist paths $\gamma_1 : x \rightarrow y$, $\gamma_2 : y \rightarrow z$ such that

$$\begin{aligned}\mathsf{C}(\gamma_1) &\leq d(x, y) + \varepsilon/2 \\ \mathsf{C}(\gamma_2) &\leq d(y, z) + \varepsilon/2\end{aligned}$$

Let $\gamma = \gamma_1 \cdot \gamma_2$ be the concatenated path, then

$$C(\gamma) = C(\gamma_1) + C(\gamma_2) \leq d(x, y) + d(y, z) + \varepsilon$$

By definition of $d(x, z)$, $d(x, z) \leq C(\gamma)$, so

$$d(x, z) \leq d(x, y) + d(y, z) + \varepsilon$$

Letting $\varepsilon \rightarrow 0$ gives the triangle inequality.

□

A.2 Turing Machine Universe Satisfies Axioms A1–A5

A1 Finite Information Density

The configuration space $X_M = Q \times \Gamma^{\mathbb{Z}} \times \mathbb{Z}$ is infinite-dimensional, but for any finite interval $R = \{i_1, \dots, i_k\} \subset \mathbb{Z}$, local tape contents $(a_i)_{i \in R}$ take values in finite set Γ^R . Local structure can be taken as “Turing machine one-step update acts only on head position and its neighboring constant positions,” corresponding G_X is a finite-degree graph, thus satisfying A1.

A2 Local Update

Transition function δ depends only on current state and current position symbol, so one-step update $(x, y) \in T_M$ is completely determined by finite local information of x .

A3 Generalized Reversibility

For general deterministic Turing machines, T_M is not necessarily globally reversible, but one can consider extending the machine to a reversible Turing machine, or only consider reversibility on “physical running trajectory” X_{phys} . Classical constructions show any deterministic Turing machine can be embedded in a reversible Turing machine (e.g., by saving history), with corresponding configuration space extended such that T_M is bijective on the extended space. Understanding $U_{\text{comp}}(M)$ in the text as this reversible extension satisfies A3.

A4 Cost Additivity and Positivity

Define $C_M(x, y) = 1$ when $(x, y) \in T_M$, otherwise ∞ . Then for any non-trivial path γ , cost is a positive integer, zero-length path has cost 0, and clearly satisfies additivity and triangle inequality.

A5 Information Quality Monotonicity

For decision tasks, let $I_M(x)$ be constantly 0 in “non-halted” configurations, 1 in accepting configurations, and some negative value or still 0 in rejecting configurations. Then along actual running paths, expected value $\mathbb{E}[I_M]$ is non-decreasing with time (because once entering accepting state, it never leaves). For more general tasks, confidence functions based on output prefixes can be introduced; construction is more involved but the principle is the same.

□

A.3 Classical Cellular Automaton Universe Satisfies Axioms

For cellular automaton $F : S^\Lambda \rightarrow S^\Lambda$, if F is reversible, its global update is bijective. Configuration space $X_{\text{CA}} = S^\Lambda$ satisfies finite information density (each lattice point state is finite and locality guarantees each step is affected only by finite neighborhood), single-step cost can be taken as constant 1. Reversibility treatment is similar to Turing machines:

non-reversible cellular automata can be constructed as reversible systems through state space extension, or reversibility imposed only on reachable trajectory set. Information quality function construction is same as Turing machine case, thus omitted.

A.4 QCA Universe Satisfies Axioms

For reversible QCA U , locality guarantees any one-step update propagates information only in finite region, making one-step reachable set finite for basis state label set X_{QCA} , thus satisfying A1–A2. U is unitary, hence there exists U^* as inverse evolution, corresponding to T_{QCA}^{-1} , thus satisfying A3 on physical state subset. Single-step cost is determined by physical implementation time or energy-time inequality, naturally positive and additive. Information quality function can be defined as monotone function of relative entropy or fidelity between reference state and current state; in Heisenberg picture, it can be proved that expected value increases monotonically under certain control tasks. Detailed technical arguments rely on monotonicity of quantum relative entropy and structure of completely positive maps; see Appendix B.5.

B Technical Details of Complexity Geometry and Category Structure

B.1 Continuous Limit of Complexity Graph (Schematic Discussion)

Let $\{U_{\text{comp}}^{(h)}\}$ be a family of computational universes corresponding to complexity graphs $G_{\text{comp}}^{(h)} = (X^{(h)}, T^{(h)}, C^{(h)})$, where $h > 0$ denotes discretization step length of “space–control scale,” and as $h \rightarrow 0$ the graph becomes “denser.”

Under suitable construction, one can embed $X^{(h)}$ into some fixed Euclidean space or manifold \mathcal{M} such that:

1. Edge weight $C^{(h)}(x, y)$ approximates $\sqrt{G_{ab}(\theta) \Delta \theta^a \Delta \theta^b}$;
2. Graph distance $d^{(h)}(x, y)$ corresponds to minimum of some discrete energy functional, whose limit is minimum of continuous energy functional, i.e., geodesic distance $d_G(\theta, \theta')$.

Similar results are well-known in studies of spectral convergence of graph Laplacians and discrete-to-continuous Dirichlet energy flows. Here we only provide structural assumptions; detailed construction will be left to subsequent systematic development in “complexity geometry theory.”

B.2 Proof of Category Property of Simulation Mappings

Proposition B.1 (Identity Mapping is Simulation Mapping)

For any computational universe $U_{\text{comp}} = (X, T, C, I)$, identity map id_X obviously preserves stepping, single-step cost, and information quality function, taking $\alpha = 1, \beta = 0, \Phi = \text{id}$, thus is a simulation mapping.

Proposition B.2 (Composition Preserves Simulation Structure)

Let $f : U_{\text{comp}} \rightsquigarrow U'_{\text{comp}}$ and $g : U'_{\text{comp}} \rightsquigarrow U''_{\text{comp}}$ be simulation mappings with parameters $(\alpha_f, \beta_f, \Phi_f)$ and $(\alpha_g, \beta_g, \Phi_g)$ respectively. For any $(x, y) \in \mathsf{T}$, we have $(f(x), f(y)) \in \mathsf{T}'$ and $(g(f(x)), g(f(y))) \in \mathsf{T}''$, thus $g \circ f$ preserves stepping. For any path $\gamma : x \rightarrow y$, there exist γ' and γ'' respectively such that

$$C'(\gamma') \leq \alpha_f C(\gamma) + \beta_f$$

$$C''(\gamma'') \leq \alpha_g C'(\gamma') + \beta_g \leq \alpha_g (\alpha_f C(\gamma) + \beta_f) + \beta_g$$

Therefore $g \circ f$ is a simulation mapping with parameters $(\alpha_g \alpha_f, \alpha_g \beta_f + \beta_g, \Phi_g \circ \Phi_f)$. Associativity of morphism composition is given by associativity of function composition. \square

B.3 Categorical Treatment of Classical Model Equivalence (Proof Sketch of Theorem 7.3)

Classical results show: any Turing machine can be simulated by cellular automata in linear or polynomial time, and vice versa. We elevate this result to the level of computational universe category **CompUniv**.

Construction of $F_{\text{TM} \rightarrow \text{CA}}$:

- Objects: Map Turing universe $U_{\text{comp}}(M)$ to some constructed cellular automaton universe $U_{\text{comp}}(F_M)$, where local rule of F_M encodes tape and state of Turing machine configuration. Encoding map $e_M : X_M \rightarrow S^\Lambda$ is realized through local patterns in space.
- Morphisms: If $f : M_1 \rightarrow M_2$ is a polynomial-time simulation mapping between Turing machines, then there exists simulation mapping $F_{\text{TM} \rightarrow \text{CA}}(f)$ between corresponding cellular automata, with parameters (α, β) related to simulation overhead.

Construction of $F_{\text{CA} \rightarrow \text{TM}}$ is similar, flattening cellular automaton configurations into Turing tape contents through space–time encoding. Existence of natural isomorphisms η, ϵ depends on reversibility of above encodings and polynomial-time decodability. Complete proof involves detailed analysis of encoding schemes and upper bound estimates of simulation complexity; see Appendix C for prototype construction.

B.4 Category Equivalence of Quantum Models (Proof Sketch of Theorem 7.4)

For reversible QCA, it is known there exist universal quantum cellular automata that can simulate any finite local QCA with polynomial overhead; on the other hand, classical universal computation can be embedded in quantum universal computational models. Restating these results in U_{comp} framework yields complexity equivalence between subcategories **QCAUniv** and **TMUniv**. Technical core is constructing unitary encoding and decoding operators such that in fixed Hilbert space, subspaces of “Turing configurations” and “QCA configurations” are mutually polynomial-time distinguishable, while stepping operator evolution realizes expected simulation on these subspaces.

B.5 A Prototype Argument for Information Quality Monotonicity in QCA

Let ρ_t be the evolution of some initial state ρ_0 in QCA universe over time in Heisenberg picture. For some observation task Q , define target state σ and relative entropy $S(\rho_t\|\sigma)$. Using monotonicity of quantum relative entropy for completely positive trace-preserving maps, one can prove that under appropriate control strategies, expected relative entropy is non-increasing, thus information quality function $I(\rho_t) = -S(\rho_t\|\sigma)$ is expectedly monotone non-decreasing. This realizes Axiom A5 at abstract level. Detailed operator algebra arguments involve Tomita–Takesaki modular theory and structure of local algebras; not expanded here.

C Encoding Constructions and Examples

This appendix briefly provides prototypes of several concrete encoding constructions to illustrate realizability of computational universe axioms and category structure.

C.1 Encoding Example from Turing Machine to One-Dimensional Cellular Automaton

Take one-dimensional lattice $\Lambda = \mathbb{Z}$, local state set S at each lattice point includes:

1. Tape symbol set Γ ;
2. State markers Q ;
3. Head marker and direction information.

Encode Turing machine configuration $x = (q, (a_i)_{i \in \mathbb{Z}}, p)$ as cellular configuration $c \in S^{\mathbb{Z}}$:

- At position i write a_i ;
- At position p attach state q and head marker.

Local rule $f : S^{\mathcal{N}} \rightarrow S$ (\mathcal{N} being e.g. $\{-1, 0, +1\}$) is designed: update local symbol and head position based on whether current position has head marker and state q . Thus, one F evolution corresponds to one Turing machine step. This construction obviously satisfies locality, Axioms A1–A4, and can be extended to reversible form.

C.2 Encoding Example from Cellular Automaton to Turing Machine

Reverse encoding can be realized by “folding” space–time evolution graph as two-dimensional encoding on Turing machine tape: let part of tape region encode spatial cross-section of CA, other regions as auxiliary storage for accumulating time steps. Turing machine local transition function simulates CA local rule and time advancement. This construction is known to complete in polynomial time and allows designing reversible Turing machine version, thus giving corresponding simulation mapping in **CompUniv**.

This paper provides basic axiomatic system of computational universe, embedding of classical and quantum models, and prototype construction of simulation category structure, laying a discrete and rigorous foundation for subsequent complexity geometry, information geometry, and unification with physical universe.