

Research Article

Exact Future Representation, Reachable Compression, and Accessibility Deformation on a Finite AKN-Type Rhombohedral Patch

Masaru Numagaki¹

1. Medcure Co., Ltd, Japan

This study asks how much of the legal future of a finite local growth system its current geometry determines. We fix a frozen 691-cell patch of Ammann–Kramer–Neri (AKN)-type icosahedral rhombohedral geometry, generated by a cut-and-project rule and reproduced exactly in $\mathbb{Q}(\sqrt{5})$ arithmetic, and equip it with a single-face attachment grammar: a catalogue cell may be added when it touches the current patch in exactly one frontier face. Unlike matching-rule and empire problems, the patch is fixed in advance; the question is which attachment sequences remain legal from the current state.

First, local three-component signatures over all move slots form a current residue $R_{\text{cat}}(s)$. Legality factors through it, legal insertions update it autonomously, and equal residues imply equal extension languages; a finite residue automaton decides every move word exactly, giving a finite exact presentation of the canonical future-signature range.

Second, this code compresses strictly without losing any future. Erasing permanently dead slots yields a coarser code that is still exact and autonomous. A formally verified witness, conditional on a finite local contact certificate audited against the catalogue, gives two distinct reachable states with legal moves that this code identifies, so their extension languages coincide; an exhaustive depth-eight search finds 562 such classes, none containing a dead end.

Third, every legal insertion strictly reduces catalogue-cell availability, and an audited branch shows that a move's coarse relative-side label does not determine its collateral loss. Because the catalogue-present rank grows by one per step, every legal word from s has length at most $691 - r_{\text{pres}}(s)$ and the reachable subgraph is finite and acyclic; after the first move of the audited branch, every run stops before the catalogue is exhausted. Release, reuse, catalogue growth, and infinite continuation are outside the present claims.

Corresponding author: Masaru Numagaki, m.numagaki@medcure.co.jp

1. Introduction

1.1. Problem

In local growth dynamics, the current structure constrains which transitions can occur next, even without an explicit record of how that structure was produced. This paper asks a forward-looking question about such constraints:

To what extent does the current finite geometry determine which finite sequences of legal moves remain executable?

For a state s , write $\text{LegalMoveSet}(s)$ for the set of moves that are legal at s . This one-step legal-move set is not the whole future structure. A legal move m produces a new state $U(s, m)$ from which further legal moves branch. The object of interest is therefore iterated reachability,

$$s \longrightarrow \text{LegalMoveSet}(s) \longrightarrow \text{LegalMoveSet}(U(s, m)) \longrightarrow \cdots,$$

equivalently, the finite-word extension language

$$\mathcal{L}(s) = \{w : \text{WordLegalFrom}(s, w)\},$$

written $\mathcal{L}_K(s)$ when the catalogue K has to be made explicit.

We study this question in one concrete setting: a frozen 691-cell patch of Ammann–Kramer–Neri (AKN)-type icosahedral rhombohedral geometry (Section 2), equipped with a labelled single-face attachment grammar in which a catalogue cell may be added through exactly one current frontier face (Section 3). The main theorems are proved for this realisation; they are not asserted for every finite local growth dynamics.

1.2. Three neighbouring lines of research

Geometry and static completion.

Rhombohedral tilings obtained by projecting $\mathbb{Z}^6$ along the icosahedral directions go back to Kramer and Neri [11], see [2][3] for the general geometric framework of projection tilings. The resulting Ammann–Kramer–Neri tiling is used today as a framework for atomic-structure models of icosahedral quasicrystals [4]. Two questions about finite patches of such tilings are classical.

The first is the question of matching and local rules: which local constraints force the admissible tilings. Katz defined facet decorations for the two Ammann rhombohedra and proved that every tiling by these rhombohedra with matching decorations is a quasiperiodic three-dimensional Penrose tiling [5]. More generally, quasiperiodic tilings obtained by the strip projection method admit local rules when the embedding has quadratic coefficients [6], and a necessary and sufficient condition is known for the slope of a canonical projection tiling to be characterised by a finite set of forbidden patterns [7].

The second is the empire problem: which tiles a patch forces. For a finite patch P , write $\Gamma(P)$ for the set of tilings of a given projection class that contain P . The empire $\mathcal{E}(P)$ is the set of tiles common to all tilings in $\Gamma(P)$ [8]. Hammock, Fang and Irwin motivate this problem by local growth: a quasicrystal growing from a local configuration is expected to have fewer defects when more of the surrounding structure is forced.

Both questions concern the global completions of a patch. Ours concerns the sequential extensions of a patch inside a catalogue that is fixed in advance. Figure 1 places the two side by side.

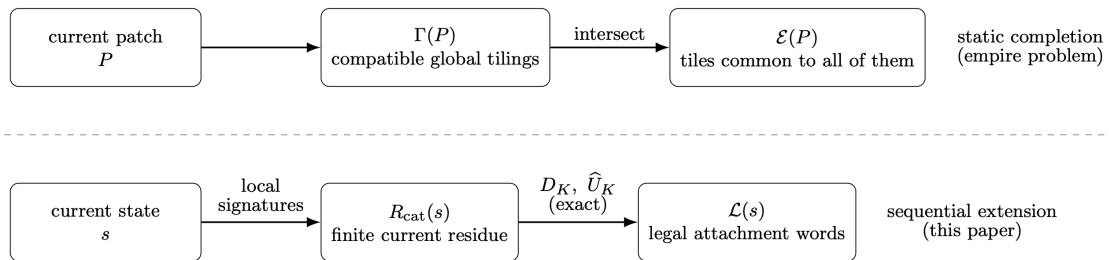


Figure 1. Two questions posed on a finite patch. Upper row: static completion, in which the global tilings compatible with a patch P determine its empire $\mathcal{E}(P)$, the tiles common to all of them [8]. Lower row (this paper): a current state s determines the finite residue $R_{\text{cat}}(s)$, which exactly decodes the language $\mathcal{L}(s)$ of legal attachment words under the single-face grammar. The lower row is not asserted to be a dynamic version of the empire construction: the upper row asks what is forced across global completions; the lower row asks which finite extension sequences remain executable under a specified sequential grammar. The two constructions are compared at the level of questions, not identified by a correspondence between their intermediate mathematical objects.

Sequential growth and attachment.

Local growth rules for Penrose tilings add tiles only where the local configuration forces them, and growth halts at a dead surface when no forced addition remains [9]; local growth of three-dimensional icosahedral tilings has been studied in [10]. In random sequential adsorption, irreversibly adsorbed objects exclude further sites until a jammed state is reached [11]. In the abstract tile assembly model, tiles attach one at a time to a growing assembly when their binding strength meets a threshold temperature, and an assembly to which no tile can attach is terminal [12][13]. In all of these models, growth constructs the tiling or the assembly. Here, the catalogue is fixed in advance, and growth only selects an order of insertion inside it. The structural analogies (available sites, attachment-induced exclusion, terminal configurations) guide the definitions of Section 7, but no identification with any of these models is asserted.

Formal future representation.

The extension language $\mathcal{L}(s)$ is a residual language in the sense of formal-language theory, and equality of extension languages is a Myhill–Nerode-type equivalence [14][15]; for the deterministic partial transition system studied here it is also a strong bisimulation [16][17]. Abstract interpretation [18] describes concrete computations by abstract objects whose execution yields information about the concrete ones, usually in an approximate way. The residue R_{cat} constructed below is an abstraction of this kind that is exact for the legality language: it neither loses nor adds a legal word.

1.3. What is new

The AKN-type geometry, finite-state language theory, and monotone-attachment concepts used here are established ingredients. The contribution of this work is their concrete conjunction: from an explicit finite rhombohedral catalogue and a specified single-face attachment grammar, we construct a geometry-derived finite-state code with an autonomous update and an exact decoder for the complete finite extension language. We prove strict loss of catalogue-cell availability under each legal insertion. Separately, the strictly increasing presence rank guarantees termination of every legal run within the fixed finite catalogue. We further give a source-audited witness that a coarse relative-side label does not determine collateral accessibility loss. Finally, permanently dead slots can be erased from the code without losing any future, and this erasure genuinely identifies distinct reachable configurations that still have legal moves.

Accordingly, no novelty is claimed for the substrate, for Myhill–Nerode theory, or for termination of monotone growth, and it is not claimed that R_{cat} or its compression $R_K^{(2)}$ is Nerode-minimal.

1.4. Main results

Main Result I: geometry-derived finite exact future representation (Section 5).

A. Complete future sufficiency:

$$R_{\text{cat}}(s) = R_{\text{cat}}(t) \implies \mathcal{L}(s) = \mathcal{L}(t).$$

B. Exact residue-automaton synchronisation: code-level and geometry-level word legality agree for every finite word.

C. Finite exact presentation of the canonical future-signature range.

Main Result II: reachable future compression (Section 6).

G. The code $R_K^{(2)}$ obtained by erasing permanently dead slots is still future-sufficient and autonomously updated. Under a finite local contact certificate audited against the frozen catalogue, two distinct states reachable from $\{165\}$, both with legal moves, satisfy $R_K^{(2)}(s) = R_K^{(2)}(t)$ and hence $\mathcal{L}(s) = \mathcal{L}(t)$; thus $R_K^{(2)}$ strictly compresses R_{cat} on reachable states.

Main Result III: accessibility deformation and failure of coarse sufficiency (Section 7).

D. Every legal insertion strictly decreases catalogue-cell availability.

E. The coarse relative-side class of a move is not a sufficient statistic for its collateral accessibility loss: a source-audited two-step branch with the same class at both moves has collateral costs 2 and 1.

Fixed-capacity boundary (Section 8).

F. The catalogue-present rank increases by exactly one at every legal step, so every legal word from s has length at most $691 - r_{\text{pres}}(s)$, and non-empty legal cycles are impossible.

Main Theorem F guarantees that every legal run stops; a separate proposition (Proposition 4) shows that, after the first move of the audited branch, every legal run stops before the catalogue is exhausted. Dead ends share the terminal language $\{\epsilon\}$ (Section 7). The letters A–G are fixed labels shared with the formal sources; Main Theorem G is stated after Main Theorems A–C because it compresses their code. Finiteness and acyclicity of this kind are expected for monotone attachment dynamics. Main Theorem F is recorded because the explicit bound marks the boundary of the present model (Section 8.2).

1.5. Scope and claim firewall

- *Geometry.* The catalogue is treated as an AKN-type substrate. Its exact reproduction under the stated projection convention is a reproducibility result, not an identification with the standard infinite AKN model set (Section 2.3).
- *Grammar.* The single-face attachment rule is not a matching rule or a local rule for the tiling (Section 2.4).
- *Locality.* Each component of $R_{\text{cat}}(s)$ is extracted locally, but $R_{\text{cat}}(s)$ itself is a catalogue-wide table. No bounded-radius code for the complete legal future is claimed (Section 4).
- *Compression.* Part (ii) of Main Theorem G is a Lean theorem whose hypothesis is a finite local contact certificate; the certificate is established for the frozen catalogue by an external audit, not inside Lean. Neither R_{cat} nor $R_K^{(2)}$ is claimed to be Nerode-minimal (Section 6.6).
- *Fixed catalogue.* All theorems concern the frozen catalogue K and its fixed move alphabet M_K . Dynamics in which the catalogue changes with time, $K_0 \subset K_1 \subset \dots$, and the existence of infinite legal runs in such systems are not addressed.
- *No physical claim.* Extension-language structure is not physical release or reuse of capacity, and neither is Capacity Recycling. The paper does not address Persistent Reconfiguration, a physical realisation of No-Dead-End, energy, physical history carriers, a bridge from an algebraic history residue to a physical carrier, or a primordial or cosmological one-bit mechanism.
- *Companion papers.* The relation to two companion papers is described in Section 10.1. No object of those papers is an input of any theorem below.

Appendix 13 lists every claim with its evidential status: Lean theorem, Lean theorem with an audited hypothesis or payload, external audit, paper proof, finite computation, citation, or open problem.

2. AKN-Type Geometric Substrate

2.1. Cells and acceptance rule

A cell is a rhombohedron specified by a base address $n \in \mathbb{Z}^6$ and a set $T \subset \{0, \dots, 5\}$ of three directions; it is the physical-space image of the unit cube $n + [0, 1]^T \subset \mathbb{R}^6$. Its six two-dimensional faces are

$$n + \sigma e_y + [0, 1]^{T \setminus \{y\}}, \quad y \in T, \sigma \in \{0, 1\}.$$

We keep the implementation name D1b for these cells and for the transition realization built on them in Section 3.

The cells come from an icosahedral cut-and-project construction of the type introduced by Kramer and Neri [1], we refer to the resulting geometry as AKN-type (Section 2.3). The six basis vectors of $\mathbb{Z}^6$ are projected onto the six five-fold directions of the icosahedron in physical space and onto the complementary three-dimensional perpendicular space. A cell (n, T) is accepted when the perpendicular image of n , shifted by a fixed offset γ , has all three coordinates in $(0, 1)$ with respect to the perpendicular images of the three basis vectors not in T .

2.2. The frozen 691-cell catalogue

The catalogue was generated once, before any of the analysis reported here, by enumerating the accepted cells in a fixed lattice box within a fixed physical-space radius. The offset γ is the perpendicular projection of a generic vector in $\mathbb{R}^6$ drawn from a pseudorandom generator with a declared seed; an earlier offset choice was found to be degenerate, and the present catalogue was generated only after it had been replaced by this generic one. The resulting patch of 691 cells passed a fixed suite of structural checks (including covering of the inner region without overlap, at most two cells per face, vertex-set identification, and connectivity) and was then frozen as the file `t7_ref2_tiles.json`, which also records the offset. “Frozen” means that the cell set, and hence the move alphabet of Section 3, is never changed afterwards. In this sense the number 691 is a fixed input of the present paper, not a fitted output: no parameter was adjusted after any quantity studied in this paper was inspected, and 691 enters the theorems only through the remaining-capacity bound of Section 8.1.

A later re-implementation of the same acceptance rule in exact $\mathbb{Q}(\sqrt{5})$ arithmetic, using the offset recorded in the file, reproduces the catalogue exactly. Ordering the accepted cells by the physical-space norm of their base points, the 691 cells are precisely those whose norm is at most R , for every R from the largest norm attained in the catalogue (≈ 4.6845 , with unit-length physical projections) up to, but excluding, the next norm attained by an accepted cell (≈ 4.7679). The audit records the midpoint $R_0 \approx 4.7262$ of this gap; the radius is therefore read off from the catalogue rather than chosen to produce 691 cells. Because the offset is likewise taken from the file, this is a reproducibility check of the same construction, not an independent derivation. Its content is that, within the audited enumeration domain, the frozen catalogue is a complete radial patch of the cut-and-project set, with no missing and no extra cells; completeness of the finite lattice-box enumeration itself is audited numerically rather than proved.

2.3. Status of the geometric statements

Table 1 separates what is audited for the catalogue from what is not claimed.

Statement	Status
Generation by the acceptance rule above, with a generic seeded offset γ recorded in the file	Documented generation procedure
Frozen file <code>t7_ref2_tiles.json</code> (691 entries)	Identified by SHA256 (Section 7.4)
Covering without overlap, at most two cells per face, vertex-set identification, connectivity	Passed at generation
Exact reproduction in $\mathbb{Q}(\sqrt{5})$ arithmetic; complete radial patch within the audited enumeration domain	Audited reproducibility check of the same construction
Completeness of the finite lattice-box enumeration	Audited numerically, not proved
Identification with the standard infinite AKN model set	Not claimed
Any matching-rule or local-rule property of the catalogue	Not claimed (Section 2.4)

Table 1. Status of the geometric statements about the 691-cell catalogue.

The geometric catalogue is treated here as an AKN-type finite icosahedral rhombohedral substrate. Exact reproduction of the finite catalogue under the stated projection convention is a reproducibility result; it is not used here as a proof of full identification with the standard infinite AKN model set. The companion paper ^[19] uses the same catalogue under the same designation.

2.4. Single-face attachment is not a matching rule

Matching rules for three-dimensional Penrose tilings were constructed by Katz, by means of facet decorations of the two Ammann rhombohedra ^[2], and local rules exist in general for projection tilings with quadratic slopes ^[6]; canonical projection tilings characterized by finite sets of forbidden patterns are studied in ^[7]. Such rules select which tilings are admissible. The single-face attachment rule of Section 3 does not select tilings: every cell it can insert already belongs to the frozen catalogue, and the rule constrains only the order of insertion. It carries no decorations, and no statement in this paper depends on a matching or local rule for the catalogue.

3. A Finite Sequential Attachment Realization

This section fixes the transition system studied in the rest of the paper: states are finite patches, moves are cell–face slots of the frozen catalogue, and a move is legal when the new cell attaches through exactly one current frontier face. The implementation name D1b/GROW-D1 used in the formal sources refers to this realization. Table 2 places it next to established questions about projection tilings and attachment growth; the comparison is at the level of questions and rules, and no correspondence between the mathematical objects of the four columns is asserted.

	Matching rules / empire ^{[5][8]}	Quasicrystal growth rules ^{[9][10]}	Sequential attachment ^{[11][12][13]}	This paper
Tiling / lattice	global tilings compatible with a patch	constructed by growth	fixed lattice or tile set	fixed finite AKN-type patch
Legality	admissible tilings; forced tiles	matching rules; forced sites	exclusion (RSA); bond strength, e.g. non-cooperative at $\tau = 1$ (aTAM)	exactly one face in contact, no other face in contact
Dynamics	none (static)	attachment	attachment	attachment only; no flips
Failure modes	—	defects	model dependent	catalogue violations excluded by construction; non-saturated dead ends occur (Proposition 4)
Stopping	—	dead surface (no forced site)	jamming / terminal assembly	dead end (no legal move)
Question	admissible completions; forced tiles	growth to infinity; defects	coverage; assembly outcome	exact extension language; accessibility loss

Table 2. Single-face attachment on a fixed patch compared with established questions about projection tilings and attachment growth.

3.1. Patch states and frontier faces

Let K be the frozen catalogue with retained cell set K . cells. In the formal branch studied here,

$$|K.\text{cells}| = 691.$$

A current patch state s is a finite set of D1b cells; it need not be assumed to be a subset of $K.\text{cells}$.

For a face f , define the current incidence count

$$n_s(f) = \#\{c \in s : f \text{ is a face of } c\}.$$

A frontier face is a face with $n_s(f) = 1$.

3.2. Fixed complete move alphabet

For the unit variant, the complete-catalogue move alphabet is

$$M_K = \{(c, f) : c \in K.\text{cells}, f \text{ is a face of } c\}.$$

A slot $i = (c, f)$ carries

$$\text{newCell}(i) = c, \quad \text{frontierFace}(i) = f.$$

This is a finite, state-independent, labelled alphabet. Row identifiers are provenance labels, not part of the move ontology. The quotient constructed below is therefore a *fixed-catalogue labelled future quotient*; no additional catalogue-relabelling invariance is asserted.

3.3. Actual-geometry legality

A move $i = (c, f)$ is actual-geometry admissible at patch s when:

1. $c \in K.\text{cells}$;
2. $c \notin s$;
3. f is a current frontier face, i.e. $n_s(f) = 1$;
4. f is a face of c ;
5. no other face $g \neq f$ of c is already incident to the current patch.

Thus the new cell attaches through exactly one selected frontier face. The update is

$$U(s, i) = s \cup \{c\}.$$

Actual-geometry legality is distinct from any SAT/UNSAT/UNKNOWN epistemic certification status; this paper uses the former.

4. Extension Languages and Exact State Abstraction

4.1. Extension languages and future equivalence

Let S be a state space and M a common labelled move alphabet. Legality is

$$\text{Legal} : S \times M \rightarrow \{\text{true}, \text{false}\},$$

and for a legal pair (s, m) an update $U(s, m)$ is defined.

For a finite word $w = (m_1, \dots, m_k)$, $\text{WordLegalFrom}(s, w)$ means that m_1 is legal at s , m_2 is legal after the first update, and so on. The empty word is always legal. The future language is

$$\mathcal{L}(s) = \{w \in M^* : \text{WordLegalFrom}(s, w)\}.$$

Future equivalence is

$$s \sim_F t \Leftrightarrow \mathcal{L}(s) = \mathcal{L}(t).$$

These definitions are generic. From now on, S is the set of patch states, $M = M_K$, Legal is actual-geometry legality, and $U(s, i) = s \cup \{c\}$ is the insertion of Section 3. We write $\mathcal{L}(s)$, or $\mathcal{L}_K(s)$ when the catalogue has to be made explicit.

Future equivalence is a Myhill–Nerode-type equivalence on states, and for this deterministic partial transition system it coincides with strong bisimulation (Section 1.2). We use “Waterway” as a shorthand for the current-state-dependent legal extension structure $(\text{LegalMoveSet}(s), \mathcal{L}(s), [s]_F)$, where $[s]_F$ is the class of s under $\sim_F$; the term does not denote a physical flow (Section 10.2).

4.2. Explicit local state signature

For each move slot $i = (c, f)$, extract from current geometry

$$\rho_s(i) = (P_s(i), I_s(i), O_s(i)),$$

where

- $P_s(i) \in \{0, 1\}$ records whether the new cell c is already present;
- $I_s(i) \in \{0, 1, \geq 2\}$ records the incidence class of the selected frontier face f ;
- $O_s(i) \in \{0, 1\}$ records whether another face of c is already incident to the current patch.

The static move signature is

$$\mu_K(i) = (C_K(i), F_K(i)),$$

where C_K records catalogue membership and F_K records that the selected face belongs to the new cell. The local decoder is true exactly when

$$C_K = 1, P_s = 0, F_K = 1, I_s = 1, O_s = 0.$$

Because $C_K = F_K = 1$ on the fixed complete alphabet M_K , the effective current-state conditions are simply

$$P_s = 0, I_s = 1, O_s = 0.$$

The static fields are retained to expose the generic factorization interface.

The coarse relative-side signature used later is

$$\sigma_{\text{side}}(m) := \text{decide}(m.\text{frontierFace.base} = m.\text{newCell.base}) \in \{\text{true}, \text{false}\}.$$

Here true denotes base-aligned and false denotes non-base-aligned. These are the LOW and HIGH relative-side classes of the companion paper ^[19], where the same label is written as a $\mathbb{Z}_2$ -valued transition parity with LOW = 0 and HIGH = 1; thus $\sigma_{\text{side}}(m) = \text{true}$ exactly when that parity is 0. The present manuscript uses base-aligned/non-base-aligned as the primary terminology.

4.3. Complete current residue

The complete current residue is the table of local signatures over all move slots:

$$\boxed{R_{\text{cat}}(s) : M_K \rightarrow \Sigma_{\text{loc}}, R_{\text{cat}}(s)(i) = \rho_s(i),}$$

with

$$\Sigma_{\text{loc}} = \{0, 1\} \times \{0, 1, \geq 2\} \times \{0, 1\}.$$

Because M_K and Σ_{loc} are finite, the residue-code type is finite.

Locality firewall.

Each component $\rho_s(i)$ is extracted locally: it depends only on the new cell c of slot i , its selected face f , and the incidence of the faces of c in the current patch. The complete residue $R_{\text{cat}}(s)$, by contrast, is a catalogue-wide finite table indexed by every slot of M_K , and the residue update below uses a static relation table of the same catalogue-wide extent. Hence

$$\boxed{\text{componentwise local} \neq \text{bounded-radius global code.}}$$

No bounded-radius descriptor and no catalogue-size-independent code for the complete legal future is claimed in this paper.

Finiteness alone is not the main point. A static executed-slot/target-slot relation table induces an autonomous residue update

$$\hat{U}_K(R, i),$$

and for a legal move i ,

$$\boxed{R_{\text{cat}}(U(s, i)) = \hat{U}_K(R_{\text{cat}}(s), i).}$$

Out-of-catalogue cells may contribute to the pre-update incidence and other-face components, but a legal update adds exactly one catalogue cell. Hence every residue-component change is determined by its pre-update value together with the static relation of the inserted cell to the target slot.

Minimality.

The residue R_{cat} is introduced as an exact code, not as a minimal one. Main Theorems A–C below show that it decodes the complete extension language; Section 6 shows that a strictly coarser code is still exact.

5. Main Result I: Geometry-Derived Finite Exact Future Representation

5.1. Exact one-step legality factorization

Proposition 1 (Exact one-step legality factorization). *For every state s and slot $i \in M_K$,*

$$\boxed{\text{Legal}(s, i) \Leftrightarrow D_K(R_{\text{cat}}(s), i).}$$

Thus equal complete residues imply identical one-step legality on all labelled slots.

Proof sketch. Each residue component is the three-part local state signature above. The actual-geometry admissibility conditions factor exactly into the local decoder conditions, slot by slot. $\square$

5.2. Exact autonomous residue update

Proposition 2 (Exact autonomous residue update). *For every legal slot i ,*

$$\boxed{R_{\text{cat}}(U(s, i)) = \hat{U}_K(R_{\text{cat}}(s), i).}$$

Proof sketch. Insertion changes new-cell presence, selected-face incidence class, and other-face occupancy of a target slot according to the static relation between the executed and target slots. Applying this local update pointwise gives the actual next-state residue. $\square$

5.3. Complete future sufficiency

Main Theorem A (Complete future sufficiency). *For all states s, t ,*

$$\boxed{R_{\text{cat}}(s) = R_{\text{cat}}(t) \implies \forall w \in M_K^*, \text{WordLegalFrom}(s, w) \Leftrightarrow \text{WordLegalFrom}(t, w).}$$

Therefore,

$$R_{\text{cat}}(s) = R_{\text{cat}}(t) \implies \mathcal{L}(s) = \mathcal{L}(t).$$

Proof sketch. Induct on the word length. One-step legality agrees by the factorization theorem. If the first move is legal, the next residues remain equal by the exact residue update, so the induction hypothesis applies to the suffix. $\square$

The primary formal certificate is `differenceBarrierD1bCompleteCatalogueResidueIsFutureSufficientV1`; see Supplementary Provenance S1.

5.4. Canonical future-language signature and quotient

Define

$$\chi_s : M_K^* \rightarrow \{\text{false}, \text{true}\}, \chi_s(w) = \text{WordLegalFrom}(s, w).$$

This is the characteristic function of the complete future language, not a finite current code. Extensionality gives

$$\chi_s = \chi_t \Leftrightarrow s \sim_F t.$$

The canonical future/Nerode state may therefore be represented either by the realised range of χ_s or by the quotient $S / \sim_F$.

5.5. Exact residue-automaton synchronisation

Define code-level legality, step, and word legality recursively from the decoder and residue update:

$$\text{LegalCode}(R, i), \quad \text{StepCode}(R, i), \quad \text{WordLegalFromCode}(R, w).$$

Main Theorem B (Exact residue-automaton synchronisation).

For every concrete patch state s and finite word w ,

$$\boxed{\text{WordLegalFromCode}(R_{\text{cat}}(s), w) \Leftrightarrow \text{WordLegalFrom}(s, w).}$$

Proof sketch. The empty word is immediate. For $i :: w$, code-level legality agrees with geometry-level legality by the one-step theorem, and after a legal step the code agrees with the actual next-state residue. Apply the induction hypothesis to the suffix. $\square$

The primary formal certificate is `differenceBarrierD1bCompleteCatalogueResidueCodeWordLegalOfStateIfFV1`.

5.6. Finite exact presentation of the canonical future-signature range

Let

$$\mathcal{R}_K^{\text{real}} = \{R_{\text{cat}}(s) : s \in S\}.$$

For a realised residue R , define the code-level future signature

$$\chi_R^{\text{code}}(w) = \text{WordLegalFromCode}(R, w).$$

Main Theorem C (Finite exact presentation of the canonical future-signature range). *The realised residue set $\mathcal{R}_K^{\text{real}}$ is finite, and*

$$\boxed{\text{range}(\chi_R^{\text{code}} \mid_{\mathcal{R}_K^{\text{real}}}) = \text{range}(\chi_s).}$$

Hence the canonical future-signature range has a finite exact presentation.

Proof sketch. Because M_K and Σ_{loc} are finite, the residue-code type and $\mathcal{R}_K^{\text{real}}$ are finite. Main Theorem B identifies the code-level signature of $R_{\text{cat}}(s)$ with χ_s for every state s . $\square$

This theorem does *not* claim that R_{cat} itself is Nerode-minimal. What is established is the implication

$$R_{\text{cat}}(s) = R_{\text{cat}}(t) \implies s \sim_F t,$$

not the converse.

The primary formal certificate is `differenceBarrierD1bCompleteCatalogueFiniteNerodeRangeEqCanonicalV1`.

Section 6 returns to the converse direction: it removes from R_{cat} the information that the future never reads, and shows that the resulting code is still exact and strictly coarser on reachable states.

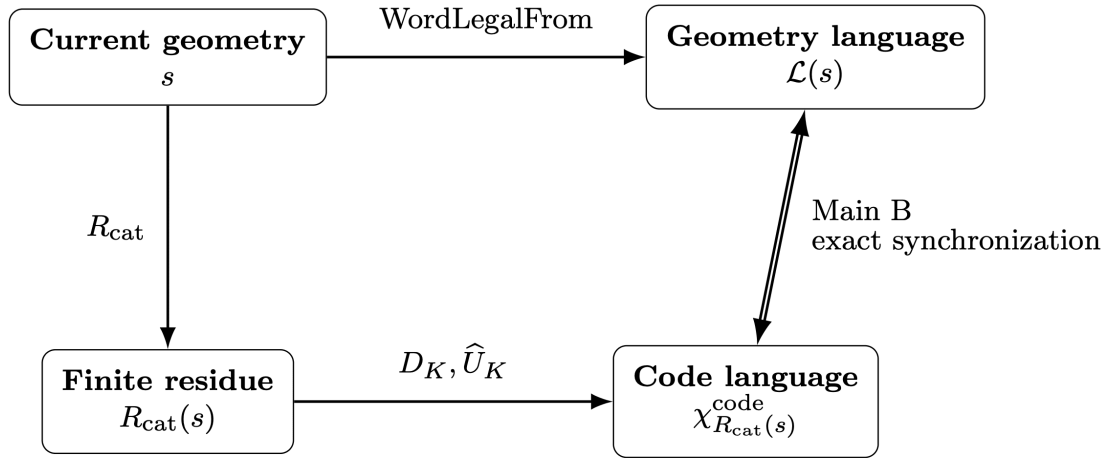


Figure 2. Commutative view of geometry-derived exact future representation. The upper route defines future language directly from geometry; the lower route passes through the finite current residue and residue automaton. Main Theorem B proves exact finite-word synchronisation, $\text{WordLegalFromCode}(R_{\text{cat}}(s), w) \Leftrightarrow \text{WordLegalFrom}(s, w)$, for every finite move word.

6. Main Result II: Reachable Future Compression

Main Theorem C presents the canonical future-signature range exactly, but it does not claim that R_{cat} is Nerode-minimal: equal residues imply equal futures, not conversely. This section identifies one part of R_{cat} that the future never reads, removes it, and shows that the removal genuinely identifies distinct reachable states.

6.1. Permanently dead slots and the normalised code

Call a local signature $\rho = (P, I, O) \in \Sigma_{\text{loc}}$ *dead* when

$$P = 1 \quad \text{or} \quad I \geq 2 \quad \text{or} \quad O = 1.$$

A dead signature is rejected by the local decoder, since legality requires $P = 0$, $I = 1$, and $O = 0$. Moreover, deadness is permanent. Inserting a cell can only set the presence bit of a target slot, raise the incidence class of its selected face (the class ≥ 2 being absorbing), or set its other-face bit; it never resets any of the three. Hence a dead signature stays dead under the local update induced by any insertion, legal or not.

Let $\rho_{\text{f}} = (1, 0, 0)$ be a fixed canonical dead value and define

$$\nu(\rho) = \begin{cases} \rho_i, & \rho^{\text{dead}}, \\ \rho, & \text{otherwise,} \end{cases} \quad R_K^{(2)}(s)(i) = \nu(R_{\text{cat}}(s)(i)) \quad (i \in M_K).$$

Thus $R_K^{(2)}(s)$ keeps the local signature of every slot that can still become legal, and replaces every permanently dead slot by ρ_i , forgetting which of the three reasons made it dead and every later change of its components.

Formal statements.

The following are proved in Lean for the local signature and for the complete residue of an arbitrary frozen catalogue object K :

1. deadness persists under the local update, for every insertion relation
(differenceBarrierD1bExtendedDeadPersistsUnderLocalUpdateV1);
2. normalisation does not change the decoder,
 $D_K(\nu(\rho), i) \Leftrightarrow D_K(\rho, i)$ (differenceBarrierD1bLocalLegalityDecodeNormalizeExtendedDeadIfV1);
3. normalisation commutes with the local update, $\nu(\hat{u}(\rho, r)) = \nu(\hat{u}(\nu(\rho), r))$ for every relation r , where $\hat{u}(\rho, r)$ is the local update of a target signature ρ under an insertion whose static relation to the target slot is r (differenceBarrierD1bNormalizeExtendedDeadCommutesWithLocalUpdateV1);
4. consequently $R_K^{(2)}$ has an exact autonomous update $\hat{U}_K^{(2)}$, with $R_K^{(2)}(U(s, i)) = \hat{U}_K^{(2)}(R_K^{(2)}(s), i)$ for every legal slot i (differenceBarrierD1bCompleteCatalogueExtendedNormalizedStateResidueUpdateCorrectV1).

Trivially, $R_{\text{cat}}(s) = R_{\text{cat}}(t)$ implies $R_K^{(2)}(s) = R_K^{(2)}(t)$. An intermediate code that normalises only the slots with $P = 1$ was formalised earlier, in a module that is not part of the archive for Main Theorems A–F and is included in Supplementary Archive S2; we write it $R_K^{(1)}$, and the same implication holds for it (differenceBarrierD1bCompleteCatalogueNormalizedResidueEqImpliesExtendedEqV1).

6.2. Main Theorem G

The reachable witness uses eleven catalogue rows. Their cells (n, T) are listed in Table 3.

Row	Base n	Directions T
165	$(-2, -1, -1, -1, 0, 1)$	$\{0, 2, 4\}$
92	$(-2, -2, -1, -1, 0, 1)$	$\{0, 1, 4\}$
507	$(-1, -2, -1, -1, 0, 0)$	$\{1, 4, 5\}$
394	$(-1, -2, -1, -1, 1, 0)$	$\{1, 2, 5\}$
345	$(-1, -2, -1, -1, 1, 0)$	$\{1, 2, 3\}$
551	$(-1, -2, -1, -1, 1, 0)$	$\{2, 3, 4\}$
163	$(-2, -2, -1, -1, 1, 0)$	$\{0, 2, 4\}$
206	$(-2, -2, -1, -1, 1, 0)$	$\{0, 2, 5\}$
0	$(-2, -2, -1, -1, 1, 1)$	$\{0, 1, 2\}$
299	$(-2, -2, -1, -1, 0, 0)$	$\{0, 4, 5\}$
6	$(-1, -2, -1, 0, 1, 0)$	$\{0, 1, 2\}$

Table 3. Catalogue rows used in the reachable witness of Main Theorem G. Rows 0 and 299 enter only through the local contact certificate; row 6 supplies a legal move that remains after the witness states.

Definition 1 (Local contact certificate).

The frozen catalogue object K satisfies the local contact certificate if

1. (a) the cells of rows 165, 92, 507, 394, 345, 551, 163, 206, 6 belong to K . cells; and
2. (b) every cell of K . cells that shares a two-dimensional face with row 163 or with row 206 is one of rows 0, 163, 206, 299, 394, 551.

Main Theorem G (Reachable strict compression of the exact future code).

- i. (Future sufficiency.) For all patch states s, t ,

$$R_K^{(2)}(s) = R_K^{(2)}(t) \implies \mathcal{L}(s) = \mathcal{L}(t).$$

- ii. (Reachable witness.) Assume the local contact certificate. Let b be the patch of rows 92, 165, 345, 394, 507, 551, and put

$$s = b \cup \{163\}, t = b \cup \{206\}.$$

Then s and t are reachable from $\{165\}$ by legal words of length six; both admit a legal move; $R_{\text{cat}}(s) \neq R_{\text{cat}}(t)$ and $R_K^{(1)}(s) \neq R_K^{(1)}(t)$; and

$$R_K^{(2)}(s) = R_K^{(2)}(t).$$

In particular $\mathcal{L}(s) = \mathcal{L}(t)$ although $s \neq t$.

Consequently, on states reachable from $\{165\}$, the code $R_K^{(2)}$ is a strict coarsening of R_{cat} and of $R_K^{(1)}$ that still decodes the complete extension language exactly.

Status of the hypotheses.

Part (i) is unconditional and holds for every frozen catalogue object K ; its formal certificate is `differenceBarrierD1bCompleteCatalogueExtendedNormalizedResidueIsFutureSufficientV1`. Part (ii) is proved in Lean as a theorem whose hypothesis is the local contact certificate (`DifferenceBarrierD1bB2H4LocalContactCertificateV1`); the formal certificate is `differenceBarrierD1bB2H4ReachableStrictCoarseningWitnessV1`. The 691 rows are not encoded in Lean, in line with the import interface of the existing archive, so the certificate is a hypothesis inside Lean. That the frozen catalogue `t7_ref2_tiles.json` satisfies it is established by a frozen external audit, which additionally shows that equality holds in (b): over all 691 rows, the cells sharing a face with row 163 or row 206 are exactly rows 0, 163, 206, 299, 394, 551. The audit also checks, line by line, that every cell and face literal in the Lean module coincides with the catalogue data (Section 6.6). We therefore do not state that the application to the actual catalogue is closed inside Lean.

6.3. Proof of Main Theorem G and the geometry of the witness

Part (i).

Normalisation does not change the decoder, so one-step legality factors through $R_K^{(2)}$. By the commutation property, equal normalised residues remain equal after the same legal move. Induction on the word length, exactly as for Main Theorem A, gives agreement on every finite word.

Reachability.

The words

$$\begin{aligned} 165 &\rightarrow 92 \rightarrow 507 \rightarrow 394 \rightarrow 345 \rightarrow 551 \rightarrow 163, \\ 165 &\rightarrow 92 \rightarrow 507 \rightarrow 394 \rightarrow 206 \rightarrow 345 \rightarrow 551 \end{aligned}$$

are legal, where each arrow inserts the named row through its unique occupied face; they end in s and t . In each state, a row that is insertable at all has exactly one face in contact with the patch, so the row sequence determines the move word. The common state after

the fifth insertion of the first word is b itself, so both s and t are one legal step from b . In s and in t , row 6 remains insertable.

Separation by R_{cat} and $R_K^{(1)}$.

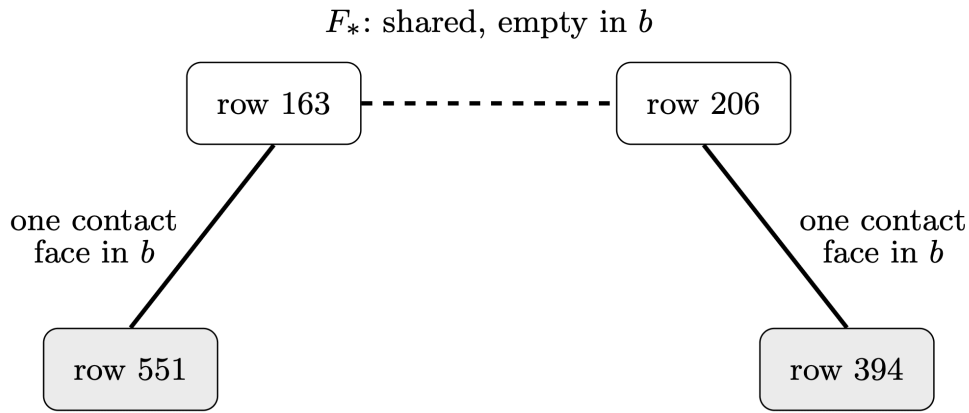
Row 163 is present in s and absent from t , so every slot of row 163 has $P = 1$ in s and $P = 0$ in t . The presence bit survives the normalisation of $R_K^{(1)}$, so both codes separate s and t .

Mutual exclusion.

In b , row 163 touches the patch in exactly one face, which it shares with row 551; row 206 touches it in exactly one face, which it shares with row 394. In addition, rows 163 and 206 share the face

$$F_* = (-2, -2, -1, -1, 1, 0) + [0, 1]^{\{0,2\}},$$

which is not occupied in b . Inserting row 163 occupies F_* , so row 206 acquires a second occupied face and becomes overconnected; inserting row 206 does the same to row 163. Hence the two rows exclude each other (Fig. 3).



patch $b = \{92, 165, 345, 394, 507, 551\}$; only the two shaded rows are drawn

Figure 3. Schematic of the mutual exclusion in the reachable witness of Main Theorem G (combinatorial, not to scale). In the patch b , row 163 touches only row 551, and row 206 touches only row 394, each through one face; both are insertable. The two rows share a face F_* that is empty in b . Inserting either row occupies F_* and makes the other row overconnected, hence permanently non-insertable. The states $s = b \cup \{163\}$ and $t = b \cup \{206\}$ differ only in which of the two rows is present and which is blocked.

Equality of $R_K^{(2)}$.

Let $i = (c, f) \in M_K$. Two cases exhaust all slots.

Case 1: c is one of rows 0, 163, 206, 299, 394, 551. Every slot of c is dead both in s and in t . Rows 394 and 551 are present in both states. Row 163 is present in s and overconnected in t , and row 206 is overconnected in s and present in t . Rows 0 and 299 already have two distinct occupied faces in s and in t . In every case each slot of c has $P = 1$ or $O = 1$, so ν sends it to ρ_i in both states.

Case 2: otherwise. By part (b) of the certificate, c shares no face with row 163 or row 206, and c is neither of them. Adding row 163 to b therefore changes neither the presence of c nor the incidence of any face of c , so $\rho_s(i) = \rho_b(i)$. The same holds for row 206, so $\rho_t(i) = \rho_b(i)$, and the two signatures coincide before normalisation.

Hence $R_K^{(2)}(s) = R_K^{(2)}(t)$, and part (i) gives $\mathcal{L}(s) = \mathcal{L}(t)$. The differences between $R_{\text{cat}}(s)$ and $R_{\text{cat}}(t)$ are confined to slots of the six rows of Case 1, all dead in both states; those between $R_K^{(1)}(s)$ and $R_K^{(1)}(t)$ lie on slots of rows 0, 163, 206, and 299. At the level of the future, the code forgets whether a row is present or permanently blocked.

All finite facts used above (incidence counts, face membership, legality of each step, and the case analysis on the six rows) are decided in Lean by kernel evaluation of computable predicates that are proved equivalent to the archive definitions; no native code evaluation is used.

6.4. Which states $R_K^{(2)}$ identifies

The witness suggests a general description. It uses the structural property, checked at generation (Table 1) and confirmed by the audit of Section 6.5, that every two-dimensional face is a face of at most two cells of K . cells.

Proposition 3. Assume that every face is a face of at most two cells of K . cells. For patch states $s, t \subseteq K$. cells:

- a. $R_{\text{cat}}(s) = R_{\text{cat}}(t)$ implies $s = t$;
- 1. $R_K^{(2)}(s) = R_K^{(2)}(t)$ if and only if $\text{Avail}(s) = \text{Avail}(t)$ and $\text{LegalMoveSet}(s) = \text{LegalMoveSet}(t)$.

Proof. (a) A slot of a cell c has $P = 1$ exactly when $c \in s$, so s is read off from the presence bits of $R_{\text{cat}}(s)$.

(b) Let $c \in K$. cells with $c \notin s$. Each face of c is a face of at most one other catalogue cell, so its incidence in s is at most one, and $I \geq 2$ never occurs on the slots of c . Exactly one of the following holds, and it fixes all six normalised signatures of c :

- $c \in s$: all slots are dead ($P = 1$);
- $c \notin s$ and c is overconnected: all slots are dead ($O = 1$);
- $c \notin s$ and exactly one face g of c is occupied: the slot (c, g) is $(0, 1, 0)$ and legal, and the other five are dead ($O = 1$);
- $c \notin s$ and no face of c is occupied: all slots are $(0, 0, 0)$.

The cells of the last two kinds form $\text{Avail}(s)$; those of the third kind are exactly the cells having a slot in $\text{LegalMoveSet}(s)$, and the slot is determined. Hence $R_K^{(2)}(s)$ determines $(\text{Avail}(s), \text{LegalMoveSet}(s))$, and conversely this pair determines, for each cell, which of the four cases holds and hence all its normalised signatures. $\square$

This proposition is a paper proof and has not been formalised; unlike Main Theorem G, it is not supported by a Lean certificate. It explains the witness: $R_K^{(2)}$ identifies exactly those configurations that differ only in which cells are present and which are permanently blocked, without any difference in the cells that remain available or in the moves that are legal. Part (a) shows that, on catalogue subsets, every identification made by $R_K^{(2)}$ is a genuine compression of distinct configurations.

6.5. Computational scope

To see how common such identifications are, all states reachable from $\{165\}$ within eight legal insertions were enumerated, with states identified as cell sets. There are 720,473 such states. Grouping them by the pair $(\text{Avail}(s), \text{LegalMoveSet}(s))$ gives 562 classes with at least two states: 527 of size two, 31 of size three, 3 of size four, and one of size five. For every one of these classes, the complete normalised residues were recomputed exactly over all 4,146 slots: $R_K^{(2)}$ is constant on each class and R_{cat} takes pairwise distinct values. No class contains a dead end, so all identifications occur between states that still have legal moves. The shallowest identifications occur at depth six, and the witness of Main Theorem G is one of them. In 165 classes the identified states differ in the number of present cells (Section 8.4).

These counts are the result of an audited computation over the explored states only. They are not a statement about all reachable states or about all subsets of K . cells. The per-class recomputation does not depend on Proposition 3. The statement that these 562 classes are all

the identifications of $R_K^{(2)}$ within depth eight does depend on it, through the grouping key. The audit also confirms that every face is a face of at most two catalogue cells.

6.6. What is not claimed

- $R_K^{(2)}$ is not claimed to be Nerode-minimal. Since future-equivalent states have the same legal moves, Proposition 3 implies that any pair identified by $\sim_F$ but not by $R_K^{(2)}$ has $\text{LegalMoveSet}(s) = \text{LegalMoveSet}(t)$ and $\text{Avail}(s) \neq \text{Avail}(t)$. Whether such pairs exist among reachable states is left open (Section 10.3).
- Only the witness of Main Theorem G is formalised; the other classes of Section 6.5 are audited by computation only.
- The local contact certificate is a Lean hypothesis audited externally, not derived inside Lean from the 691 rows.
- The locality firewall of Section 4 applies unchanged: $R_K^{(2)}$ is still a catalogue-wide table, and no bounded-radius code for the complete legal future is claimed.
- “Forgetting” refers to the code only. No statement is made about physical memory, erasure, or irreversibility.

Provenance of this section.

The Lean sources supporting Main Theorem G form Supplementary Archive S2. It contains the complete A5 import closure of the B-2H4 module (34 Lean files), an entry point that names every certificate cited in this section, and the axiom audits of the B-2H3 and B-2H4 theorems, each of which depends only on `propext`, `Classical.choice`, and `Quot.sound`. The frozen certificate record and the scripts of Section 6.5 form Supplementary Archive S3, together with a script that reruns every audit and compares its outputs with the recorded ones. Both archives are separate from the corresponding archive for Main Theorems A–F, whose content and digest are unchanged. File names and digests are listed under Data and Code Availability; build records are in Supplementary Provenance S1.

7. Main Result III: Accessibility Deformation and Failure of Coarse Sufficiency

7.1. Catalogue-cell availability

Availability is a cell-level object and is not identical to move-slot legality. It is defined only for catalogue cells:

$$\text{Avail}_K(s) = \{c \in K.\text{cells} : c \notin s \wedge \neg \text{Overconnected}(s, c)\},$$

where $\text{Overconnected}(s, c)$ means that two distinct faces of c already have positive incidence in the current patch. Cells outside $K.\text{cells}$ are never available. Since K is fixed throughout, we abbreviate $\text{Avail}(s) := \text{Avail}_K(s)$. By contrast,

$$\text{LegalMoveSet}(s) \subseteq M_K$$

is a move-level exact legality set that includes the selected frontier face. The two objects differ both in type and in determining criteria.

7.2. Strict accessibility loss

Main Theorem D (Strict accessibility loss). *For every state s and every legal slot i ,*

$$\text{Avail}(U(s, i)) \subsetneq \text{Avail}(s)$$

and, by finiteness,

$$|\text{Avail}(U(s, i))| < |\text{Avail}(s)|.$$

Proof sketch. For a legal insertion i , face-incidence counts do not decrease, so overconnected cells remain overconnected. Therefore

$$\text{Avail}(U(s, i)) \subseteq \text{Avail}(s).$$

Legality also makes the executed cell c_i available before the move, whereas it is present after the move:

$$c_i \in \text{Avail}(s), c_i \notin \text{Avail}(U(s, i)).$$

□

The primary formal certificate is `differenceBarrierD1bWaterwayStrictAccessibilityLossEveryLegalStepV1`.

73. Loss set and collateral accessibility cost

Define

$$\text{Lost}(s, i) = \text{Avail}(s) \setminus \text{Avail}(U(s, i)).$$

The executed cell always belongs to $\text{Lost}(s, i)$. Remove this universal self-loss:

$$\text{CollateralLost}(s, i) = \text{Lost}(s, i) \setminus \{c_i\},$$

and define the *collateral accessibility cost*

$$C(s, i) = |\text{CollateralLost}(s, i)|.$$

This is a one-cell-baseline subtraction, not normalisation by a scale or total capacity. Historical Lean identifiers retain the name `NormalizedAccessibilityCost`. The bookkeeping identity is

$$|\text{Lost}(s, i)| = C(s, i) + 1.$$

The quantity $C(s, i)$ is a combinatorial accessibility-loss measure in the frozen insertion dynamics; it is not an energy or a physical resource-consumption variable.

74. Coarse-label insufficiency for collateral accessibility cost

Consider the source-grounded actual-ref2 two-step branch

$$s_0 \xrightarrow{i_1} s_1 \xrightarrow{i_2} s_2,$$

corresponding to source rows $165 \rightarrow 0 \rightarrow 394$.

Main Theorem E (Coarse-label insufficiency for collateral accessibility cost). *Both executed moves are non-base-aligned in the coarse relative-side signature:*

$$\sigma_{\text{side}}(i_1) = \sigma_{\text{side}}(i_2).$$

An independent source audit of the authoritative frozen 691-cell catalogue gives

$$C(s_0, i_1) = 2, C(s_1, i_2) = 1.$$

Therefore

$$\sigma_{\text{side}}(i_1) = \sigma_{\text{side}}(i_2) \quad \text{but} \quad C(s_0, i_1) \neq C(s_1, i_2).$$

In particular, no law

$$F : \{\text{true}, \text{false}\} \rightarrow \mathbb{N}$$

depending only on the coarse side label can fit both audited moves:

$$\boxed{\text{same coarse side label} \Rightarrow \text{same collateral accessibility cost.}}$$

This claim is intentionally limited. The two audited points use different state–slot pairs, so the theorem does not attribute the difference solely to ambient-state variation. A stronger same-slot/different-state witness is not claimed here.

Relation to the companion branch.

The branch $s_0 = \{\text{row-165}\} \xrightarrow{i_1} s_1 \xrightarrow{i_2} s_2$ is the same source-level legal two-step branch that [19] uses as its HIGH–HIGH witness. There, the parity register returns to its initial value, $\Phi_0 \rightarrow \Phi_0 + 1 \rightarrow \Phi_0$; here, catalogue availability decreases strictly along the same branch, $690 \rightarrow 687 \rightarrow 685$. Equality of the parity register therefore does not determine equality of catalogue–cell availability. This comparison is a remark only: neither Φ nor its certificate is an input of Main Theorem E.

Provenance boundary.

Main Theorem E is not an end-to-end Lean calculation from raw JSON bytes. Its evidentiary chain is

authoritative source-catalogue audit $\rightarrow (C_1, C_2) = (2, 1) \rightarrow$ Lean-certified theorem consequence.

The authoritative catalogue is the file `t7_ref2_tiles.json`, with 691 entries, 36,922 bytes, and SHA256

`aca99e0c9b55398d7dd0cd35831996fdad5e9250aa4a4bcb23961329b6fd60f9`.

On this file the verifier `WATERWAY_MAINE_SOURCE_AUDIT_v1_1.py` (deposited with the data; see Data and Code Availability) reproduces

$$|\text{Avail}| : 690 \rightarrow 687 \rightarrow 685,$$

with

$$\text{CollateralLost}_1 = \{92, 366\}, \text{CollateralLost}_2 = \{206\},$$

and therefore $C_1 = 2$, $C_2 = 1$. The file is identified by its SHA256 digest. The Lean overlay accepts this exact source-audited payload and type-checks the label equality and the theorem consequences of $2 \neq 1$.

The primary formal certificates are `w20WaterwaySourceSameLabelDifferentNormalizedCostV1` and `w20WaterwayNoSideOnlyNormalizedCostLawV1`.

same coarse label: non-base-aligned

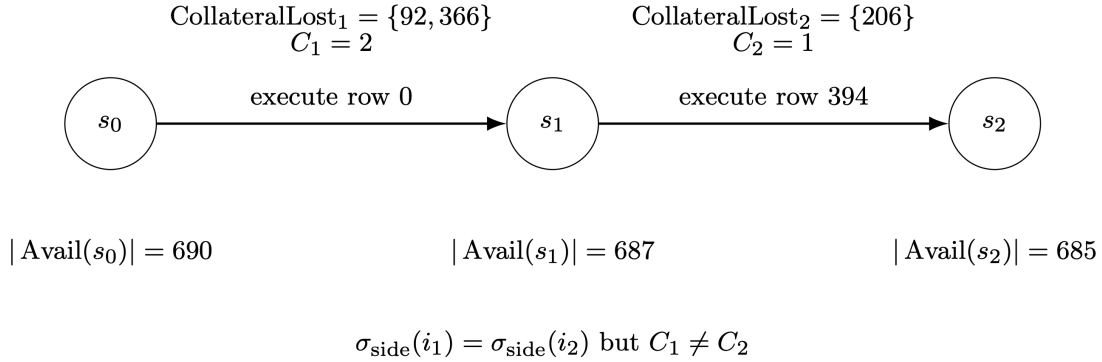


Figure 4. Audited two-step witness for coarse-label insufficiency on source rows $165 \rightarrow 0 \rightarrow 394$. Row numbers in the loss sets are zero-based catalogue indices in the authoritative source JSON. The executed row is removed before computing collateral loss, so $|\text{Lost}| = C + 1$ gives $3 = 2 + 1$ and $2 = 1 + 1$.

7.5. Dead-end future collapse and saturated sufficient condition

If a state s has no one-step legal moves, then no nonempty word is legal. Hence every dead end satisfies

$$\mathcal{L}(s) = \{\epsilon\}.$$

Therefore any dead-end states s, t are future-equivalent.

For the present grammar, catalogue saturation

$$K.\text{cells} \subseteq s$$

is a sufficient condition for a dead end: every new cell in the complete move alphabet is already present. Thus all saturated states lie in the terminal future class.

This statement alone does not establish strict state compression, since it concerns only states without a future. Distinct reachable states that still have legal moves and are future-equivalent are supplied by Main Theorem G (Section 6).

8. Finite-Capacity Boundary

This section records the boundary of the present model. Its main statement is expected for monotone attachment dynamics; it is recorded because the explicit bound shows where the fixed-catalogue grammar ends.

8.1. Catalogue-present rank and remaining-capacity bound

Define

$$r_{\text{pres}}(s) = |s \cap K.\text{cells}|.$$

A legal insertion adds exactly one previously absent catalogue cell, so

$$r_{\text{pres}}(U(s, i)) = r_{\text{pres}}(s) + 1.$$

For a legal word w , let $U_w(s)$ denote the state reached from s by executing the moves of w in order. Then

$$r_{\text{pres}}(U_w(s)) = r_{\text{pres}}(s) + |w|.$$

Because $r_{\text{pres}} \leq 691$,

$$|w| \leq 691 - r_{\text{pres}}(s).$$

8.2. Acyclic terminal structure

Main Theorem F (Acyclic terminal structure).

no nonempty legal complete-catalogue cycle exists.

Moreover,

For every fixed initial patch s , its reachable subgraph is finite and acyclic, with depth at most $691 - r_{\text{pres}}(s)$.

Proof sketch. If a legal word w returned to the original patch state, rank equality would give

$$r_{\text{pres}}(s) = r_{\text{pres}}(s) + |w|,$$

hence $|w| = 0$. The finiteness statement is local to a fixed initial patch. The patch-state space itself is not finite, because a patch is an arbitrary finite set of D1b cells and need not lie inside K . cells. However, every legal insertion adds one catalogue cell, so every state reachable from s has the form $s \cup T$ with $T \subseteq K$. cells; combine this with the remaining-capacity bound. $\square$

The finiteness of the reachable set is an immediate consequence of the update rule and is not recorded as a separate formal certificate. Waterway therefore does not imply indefinite continuation; any release/reuse mechanism requires additional dynamics outside the present theorem chain.

Primary formal certificates include `differenceBarrierD1bNoNonemptyLegalCatalogueCycleV1` and `differenceBarrierD1bCatalogueSaturatedFutureEquivalentV1`.

For monotone attachment dynamics, finiteness and acyclicity of the reachable set from a fixed initial patch are expected; we record them as Main Theorem F, with a formal certificate, because the explicit bound $|w| \leq 691 - r_{\text{pres}}(s)$ marks the boundary of the present model: a fixed finite catalogue with irreversible insertion admits no unbounded legal run. Any continuation beyond this bound requires additional dynamics, such as release, reuse, or catalogue growth, none of which is part of the present theorem chain.

8.3. Premature termination

Main Theorem F guarantees that every legal run stops; the audited branch of Main Theorem E shows that runs can be forced to stop before the catalogue is exhausted.

Proposition 4 (Premature termination). *Let $s_1 = \{165, 0\}$ be the state after the first move of the audited branch of Section 74. No saturated state is reachable from s_1 . Consequently every maximal legal run from s_1 ends in a dead end that is not saturated, and non-saturated dead ends are reachable from $\{165\}$.*

Proof. Row 92 belongs to `CollateralLost1`, so $92 \notin \text{Avail}(s_1)$ and $92 \notin s_1$. By the monotonicity in the proof of Main Theorem D, $92 \notin \text{Avail}(s)$ for every state s reachable from s_1 . A cell inserted by a legal move is available before the move, so row 92 is never inserted, and $92 \notin s$ for every such s . Hence K . cells $\not\subseteq s$. By Main Theorem F, every legal run from s_1 is finite, so every maximal legal run ends in a dead end, which by the above is not saturated. $\square$

The proposition is a paper proof. It combines the Lean-certified strict loss of Main Theorem D and the Lean-certified remaining-capacity bound of Main Theorem F with the elementary fact that a legally inserted cell is available before the move and with the source-audited fact $92 \in \text{CollateralLost}_1$; it is not recorded as a separate formal certificate. Row 92 is excluded because it becomes overconnected in s_1 , which an independent recomputation on the frozen catalogue confirms. Row 366 is excluded for the same reason.

8.4. The presence rank is not a function of the future

The presence rank is not a function of the future language. The exhaustive depth-8 search of Section 6.5 contains states with $R_K^{(2)}(s) = R_K^{(2)}(t)$, and hence $\mathcal{L}(s) = \mathcal{L}(t)$, but $r_{\text{pres}}(s) = 7 \neq 8 = r_{\text{pres}}(t)$; an example is $s = \{117, 141, 165, 255, 366, 507, 594\}$ and $t = \{92, 141, 165, 255, 391, 479, 507, 594\}$. The bound of Main Theorem F is therefore an upper bound on run length, not an invariant of the future. The equality $R_K^{(2)}(s) = R_K^{(2)}(t)$ for this pair is established by the exact computation of Section 6.5, not by a Lean certificate.

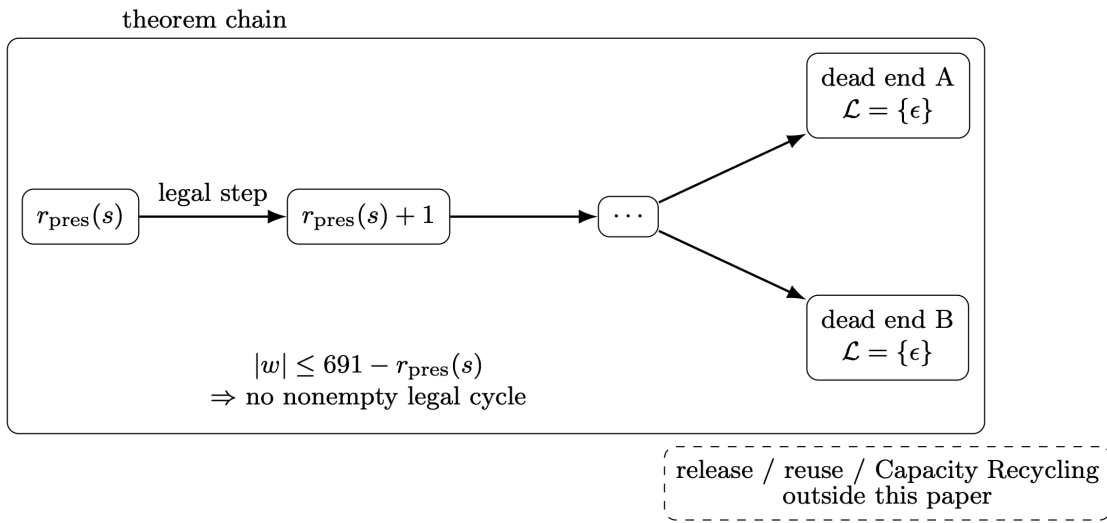


Figure 5. Monotone catalogue-present rank and the acyclic terminal structure of the reachable subgraph from a fixed initial patch. The dashed recycling box is deliberately outside the theorem chain and has no transition arrow from a dead end; release/reuse is a separate theory, not a consequence of terminality.

9. Novelty Boundary

The ingredients of this paper are established, and no novelty is claimed for them. The substrate is an AKN-type projection catalogue (Section 2). The future-language quotient is in the Myhill–Nerode tradition of formal-language and automata theory [\[14\]\[15\]](#); no novelty is claimed for future-language equality or automaton minimisation as such. Bisimulation provides another behaviour-preserving state equivalence [\[16\]\[17\]](#). Here the transition system is deterministic and partial on a fixed labelled alphabet: for each state and legal label there is a unique successor. Under these conditions, future-language equality gives equality of enabled labels and equality of the suffix languages after matching legal labels, so $\sim_F$ is naturally a strong bisimulation relation. Classical lumpability addresses preservation of probabilistic transition laws under state aggregation [\[20\]](#); the present setting has no transition probabilities. Finiteness and acyclicity of monotone attachment are expected, and Main Theorem F is recorded as a boundary of the model, not as a new phenomenon (Section 8).

Two distinctions locate the question among neighbouring problems. Static completion asks which global tilings contain a patch and which tiles all of them share (Fig. 1); the question here is which finite sequences of insertions remain legal from the current state. Growth

models construct the tiling or the assembly; here the catalogue is fixed in advance, and growth only selects an order of insertion inside it (Table 2).

The contribution is a concrete, formally verified chain from an explicit finite geometry to an exact representation of the complete legal future,

$$\begin{array}{c} \text{AKN-type catalogue and single-face grammar} \rightarrow \\ \text{explicit finite current residue} \rightarrow \text{exact decoder/update} \rightarrow \\ \text{exact finite-word recognition} \rightarrow \text{finite canonical future presentation,} \end{array}$$

followed by a strict compression of that representation on reachable states,

$$\text{exact residue} \rightarrow \text{dead-slot normalization} \rightarrow \text{reachable strict compression.}$$

General Myhill–Nerode theory supplies a coarsest exact code, the quotient map $s \mapsto [s]_F$, but not an explicit one. The second chain shows that an explicit, geometrically defined erasure is exact and identifies distinct reachable states that still have legal moves. Its witness is a Lean theorem whose hypothesis, the local contact certificate, is established for the frozen catalogue by an external audit (Section 6.2).

Two further results concern accessibility rather than representation: every legal insertion strictly reduces catalogue-cell availability (Main Theorem D), and, after compulsory self-loss is removed, the coarse relative-side label is insufficient for the collateral accessibility cost (Main Theorem E).

10. Scope, Companion Papers, and Open Problems

10.1. Companion papers

Two companion papers study different objects. No object of either paper is an input of any theorem here.

The companion paper ^[19] rereads one history-derived parity from current geometry through a current-state potential

$$\Phi : S \rightarrow \mathbb{Z}_2,$$

without requiring a dedicated event log. That result is neither full history reconstruction nor future-behaviour preservation. That paper distinguishes a history register from a history-effective residue, whose current-state differences change the legal future, and reserves future-effective reachability for a companion study in preparation; the present paper takes up that future side on the same frozen catalogue. The two papers address different directions:

past	→	current observable
Current-State Compilation		
current state	→	reachable futures
Waterway		

Their central objects are different (Table 4):

$$\text{history register} \neq \text{future-sufficient residue} \neq \text{future-language quotient.}$$

In particular, neither $R_{\text{cat}}, R_K^{(2)}$, nor the finite residue automaton uses Φ as a theorem input. The essential firewall is

$$\Phi(s) = \Phi(t) \Rightarrow \mathcal{L}(s) = \mathcal{L}(t).$$

This non-implication is stated in ^[19]; the audited branch of Section 7.4 gives a concrete instance. Its states s_0, s_1, s_2 and moves i_1, i_2 are those of the certificate-conditional HIGH–HIGH branch of ^[19], read with the same catalogue, the same exact-one-face legality, and the same insertion update; there, under a registered source certificate, $\Phi(s_0) = \Phi(s_2)$. This equality is a result of that paper and is not proved here. The move i_1 is legal at s_0 but not at s_2 , because its target cell, row 0, is present in s_2 . Hence $\mathcal{L}(s_0) \neq \mathcal{L}(s_2)$.

Layer	Question	Central object	Meaning of equality
Current-State Compilation	Can a past-derived observable be reread from the present?	$\Phi(s)$	The current value of one history-derived observable is equal.
Waterway residue	Is the current finite code sufficient for the complete legal future?	$R_{cat}(s)$	The finite code used to reproduce complete future behaviour is equal.
Compressed Waterway code	Which part of that code can be erased without losing any future?	$R_K^{(2)}(s)$	The code is equal after permanently dead slots are erased; the complete legal future language is then equal (Main Theorem G).
Waterway quotient	Which current-state differences do not affect the legal future?	$[s]_F / \chi_s$	The complete legal future language is equal.

Table 4. Boundary between the current-state compilation problem and the Waterway problem.

A second companion paper [\[21\]](#) studies yet another object: a central 2-torsion residue of finite histories, together with the question of which admissible history deformations preserve it, with the central extension $1 \rightarrow C_2 \rightarrow 2I \rightarrow A_5 \rightarrow 1$ as the concrete case. That residue concerns the past rather than the future, and it is likewise not an input of any theorem here.

10.2. What “Waterway” denotes

The term *Waterway*, introduced in Section 4.1, is used in this paper as a shorthand for the current-state-dependent legal extension structure; it does not denote a physical field, matter flow, or energy flow. The paper uses three compatible layers:

$$\begin{aligned} \text{W1 : } & \text{LegalMoveSet}(s), \\ \text{W2 : } & \mathcal{L}(s), \\ \text{W3 : } & [s]_F \in S / \sim_F. \end{aligned}$$

Thus

$$\text{Waterway} = \text{legal choices} + \text{their iterated future language} + \text{future-equivalence class.}$$

The term is deliberately limited to

$$\text{a finite reachability network allowed by the current structure.}$$

Main Theorem F makes this limitation explicit. The results stated above, and their limits, can be summarised as

$$\begin{aligned} \text{same coarse relative-side label} & \not\Rightarrow \text{same collateral accessibility cost,} \\ \text{dead-end states} & \Rightarrow \text{the same terminal future language } \{\epsilon\}, \\ \text{Waterway} & \not\Rightarrow \text{a nonempty legal cycle,} \\ \text{future-language sufficiency} & \not\Rightarrow \text{persistent evolution.} \end{aligned}$$

The following statements are intentionally not identified:

$$\begin{aligned} \text{future-equivalent} & \not\Rightarrow \text{physically interchangeable carrier,} \\ \text{future-effectively irrelevant} & \not\Rightarrow \text{physically released,} \\ \text{erased from the code} & \not\Rightarrow \text{physically erased,} \\ \text{availability loss} & \neq \text{capacity consumption in a recycling theory,} \\ \text{terminal future class} & \neq \text{physical end of the universe,} \\ \text{acyclic frozen insertion grammar} & \not\Rightarrow \text{all extended dynamics are acyclic,} \\ \text{Waterway} & \not\Rightarrow \text{Capacity Recycling.} \end{aligned}$$

10.3. Open problems

Nerode minimality on reachable states.

By Proposition 3, any pair of reachable states identified by $\sim_F$ but not by $R_K^{(2)}$ has the same legal moves and different availability. Does such a pair exist? Equivalently: for two reachable states with the same legal moves, does every difference in availability eventually become visible in the legality of some finite word? Because the subgraph reachable from a fixed initial patch is finite (Main Theorem F), the restriction of $\sim_F$ to it can in principle be computed by partition refinement [22]. We have not determined whether this computation is feasible for the subgraph reachable from $\{165\}$; the search of Section 6.5 covers depth eight only.

Formal status of Propositions 3 and 4.

Proposition 3 is a paper proof. A Lean version would separate a hypothesis-free form from the consequence that uses the property that each face lies in at most two catalogue cells. Proposition 4 combines Lean-certified statements, elementary consequences of the legality definition, and a source-audited fact, and has not been recorded as a single formal certificate.

History-derived parity versus the future quotient.

For future-equivalent states, does the parity register of [19] take equal values? The firewall above excludes only the converse direction. No result on this question is used or claimed here.

Same-slot witness.

Main Theorem E uses two different state–slot pairs. A witness in which the same slot has different collateral costs in different states, which would show a stronger dependence on the ambient state, remains open.

Release, reuse, and growing catalogues.

The next question lies outside this paper:

Under finite capacity, by what mechanism could new legal update possibilities be regenerated?

The present work stops after determining which futures the current structure permits and which accessibility options legal updates remove. The question bifurcates into two mathematically distinct extensions. One may modify the state dynamics on a fixed catalogue, for example by introducing release or reuse. Alternatively, one may enlarge the catalogue itself,

$$K_0 \subset K_1 \subset K_2 \subset \cdots,$$

thereby enlarging the labelled move alphabet. Neither extension is part of Main Theorems A–G.

Main Theorem F is pointwise in the frozen catalogue K and its fixed alphabet M_K . It does not imply acyclicity, or any bound on run length, for a non-autonomous sequence of catalogues: each extension enlarges the alphabet, $M_{K_n} \subset M_{K_{n+1}}$, and can introduce moves absent from the earlier alphabet. Thus

$$\boxed{\text{acyclic at every fixed } K \not\Rightarrow \text{acyclic under } K_0 \subset K_1 \subset \cdots.}$$

Even if every frozen stage is acyclic, continuation enabled by later catalogue growth is a separate question. The same applies to other state spaces built on the same cut-and-project data, for example catalogue families generated under inflation. An infinite run in such an extended system would not contradict the bound $|w| \leq 691 - r_{\text{pres}}(s)$, which is proved here for the frozen catalogue only.

Subsequent formal work, in preparation, takes the second route. It generalises the capacity bookkeeping of the frozen catalogue to arbitrary finite catalogues, defines from the cut-and-project data underlying the 691-cell catalogue a nested family of finite catalogues whose cardinality is unbounded, and shows, for the present legality kernel, that enlarging a finite catalogue never changes the legality of an existing move at a fixed state: within the nested family, for every state s and every $i \in M_{K_n}$,

$$\text{Legal}_{K_n}(s, i) \Leftrightarrow \text{Legal}_{K_{n+1}}(s, i).$$

This locality of existing moves does not imply $\mathcal{L}_{K_n}(s) = \mathcal{L}_{K_{n+1}}(s)$, because the larger alphabet contains new move labels. The two objects therefore cannot be identified:

The fixed-catalogue future language and the future language over a growing alphabet need not coincide.

If further conditions, such as orientation or boundary rules, are later added to legality, this locality must be re-established. Moreover, an unbounded catalogue is not the same as unbounded legal dynamics; whether an infinite legal run exists in the growing system remains open. None of these results alters a theorem of the present paper. They identify a separate mechanism by which new future possibilities may enter, namely enlargement of the substrate itself.

11. Discussion and Formal Provenance

11.1. What the results say about the current state

The current state is not merely an instantaneous configuration: together with the transition law, it contains the finite information needed to determine legal future words. This does not mean that a list of future trajectories is stored. Rather, the finite residue and its autonomous update generate the future language exactly.

Not all of this information matters. All dead-end states have the same terminal future language $\{\epsilon\}$, so the terminal future language does not encode microscopic identity. Main Theorem G shows that the same holds before termination: two distinct reachable states that still have legal moves can have the same future, and the normalised code $R_K^{(2)}$ identifies them. By Proposition 3, what $R_K^{(2)}$ forgets is the distinction between a cell that is present and a cell that can never be inserted. Even the number of present cells is not determined by the future (Section 8.4).

A legal move changes both the current state and its future language:

$$s \longrightarrow U(s, i), \mathcal{L}(s) \longrightarrow \mathcal{L}(U(s, i)).$$

At the same time,

$$\text{Avail}(s) \longrightarrow \text{Avail}(U(s, i)).$$

The minimal operational picture is therefore

current structure	$\rightarrow$ legal move	$\rightarrow$ accessibility deformation
		$\rightarrow$ new future language.

Main Theorem E shows that the one-bit relative-side label is insufficient to determine the collateral part of this deformation. Predicting the collateral effect requires more geometric information than the coarse label alone.

11.2. Formal verification and provenance boundary

The formal modules belong to a framework named Difference Barrier. The present paper does not assume that Difference Barrier is a history-memory device. It uses only a current-state legality mechanism,

$$\boxed{\text{current local structure} \longrightarrow \text{legal / illegal discrimination},}$$

from which

$$\text{legality} \longrightarrow \text{future language} \longrightarrow \text{future quotient}$$

is constructed.

Main Theorems A–C are supported by the existing B2E/B2H future-language core. Main Theorem D and the theorem consequence of Main Theorem E are supported by Waterway-specific BUILD-verified overlays. For Main Theorem F, acyclicity and terminal collapse are supported by the existing B2G/B2K core, while the sharpened remaining-capacity bound is supplied by a BUILD-verified Waterway overlay. The numerical payload $C_1 = 2, C_2 = 1$ for Main Theorem E is independently source-audited rather than computed from raw JSON inside Lean:

$$\boxed{\text{source-catalogue audit} \rightarrow (2, 1) \rightarrow \text{Lean-certified consequence}.}$$

Main Theorem G is supported by two further modules (Section 6.2): part (i) by the extended dead-slot normalisation module, and part (ii) by the reachable-witness module, whose hypothesis is the local contact certificate. Its provenance has the same structure as that of Main Theorem E:

$$\boxed{\text{frozen catalogue audit} \rightarrow \text{local contact certificate} \rightarrow \text{Lean-certified consequence}.}$$

Propositions 3 and 4, the counts of Section 6.5, and the presence-rank example of Section 8.4 are not formal certificates; their status is stated where they appear and collected in Appendix 13.

The parity-register chain of [19], the central-residue chain of [21], the later PFHC2 release/reuse chain, and the growing-catalogue D1b–G chain are outside the theorem inputs of Main A–G and outside the recursive import closure of the clean Waterway certificates. Supplementary Provenance S1 records theorem identifiers, module hashes, build status, source-audit hashes, and the dependency firewall.

12. Conclusion

This paper studies how far current finite geometry determines the complete legal future of single-face attachment on a frozen 691-cell AKN-type rhombohedral patch.

First, the complete current residue $R_{\text{cat}}(s)$ collects a three-component local legality signature over all move slots. It exactly decodes one-step legality and evolves autonomously under a legal insertion, so equality of residues forces equality of the complete finite-word future language. A finite residue automaton synchronises word by word with geometry-level legality, and the realised code signatures coincide with the canonical future-signature range.

Second, this exact code is not minimal, and part of it can be erased without losing any future. Replacing every permanently dead slot by one canonical value gives a coarser code $R_K^{(2)}$ that is still exact and autonomously updated. A formally verified witness, conditional on a finite local contact certificate audited against the frozen catalogue, shows two distinct states reachable from the same seed, both with legal moves, that $R_K^{(2)}$ identifies; their configurations differ only in which of two mutually exclusive rows is present and which is permanently blocked. An exhaustive search to depth eight finds 562 such classes, none containing a dead end.

Third, catalogue-cell availability decreases strictly at every legal insertion. After removing the compulsory loss of the executed cell, a source-audited pair of moves with the same coarse relative-side label has collateral accessibility costs 2 and 1, so the coarse label alone does not determine the collateral effect.

Fourth, the catalogue-present rank increases by exactly one at each legal step; every legal word satisfies

$$|w| \leq 691 - r_{\text{pres}}(s),$$

and nonempty legal cycles are impossible. This bound guarantees that every run stops. A separate proposition shows more: along the audited branch, a row becomes permanently non-insertable after the first move, so every subsequent run stops before the catalogue is exhausted. The bound is not an invariant of the future, since future-equivalent states can differ in the number of present cells.

Thus the central chain is

finite local geometry $\rightarrow$ explicit future-sufficient residue $\rightarrow$ exact finite residue automaton $\rightarrow$ finite exact canonical future-signature presentation.

followed by

exact residue $\rightarrow$ erasure of permanently dead slots $\rightarrow$ strictly coarser exact code on reachable states.

together with

legal update $\rightarrow$ strict accessibility loss and coarse-label-insufficient collateral deformation $\rightarrow$ acyclic terminal structure; along the audited branch, termination before saturation.

Nerode minimality of the compressed code, physical release, reuse, Capacity Recycling, catalogue growth and time-dependent move alphabets, Persistent Reconfiguration, No-Dead-End realisation, and infinite continuation remain separate problems (Section 10.3).

Appendix A. Claim firewall

Table A1 lists each claim of the paper with its evidential status. The status keys are as follows.

Lean. A Lean theorem about the frozen catalogue object K , with no hypothesis beyond the definitions.

Lean + certificate. A Lean theorem with an explicit finite hypothesis about the catalogue; that the frozen file satisfies the hypothesis is established by an external audit, not inside Lean.

Lean + payload. A Lean consequence of numerical values that are supplied by an external audit of the frozen file.

Audit. An external audit of the frozen catalogue file.

Paper. A proof given in this paper and not formalised.

Computation. An audited exhaustive computation over the explored states only.

Cited. A result of a companion paper, used in a remark only and never as a theorem input.

Open / not claimed. Stated as an open problem, or explicitly outside the claims.

Lean identifiers are given where the main text does not already give them; module hashes and build records are listed in Supplementary Provenance S1.

Claim	Section	Status	Basis
<i>Substrate</i>			
Generation by the acceptance rule; covering without overlap, at most two cells per face, vertex-set identification, connectivity	2.3	Audit	Checks at generation (Table 1); the face property is re-confirmed in Section 6.5
Exact reproduction in $\mathbb{Q}(\sqrt{5})$ arithmetic	2.3	Audit	Reproducibility check of the same construction
Identification with the standard infinite AKN model set; matching-rule or local-rule property	2.3, 2.4	Not claimed	—
<i>Main Result I</i>			
Exact one-step legality factorisation (Proposition 1)	5	Lean	differenceBarrierD1bCompleteCatalogueActualGeometryLegalityFactorizationV1, in the module of Main Theorem A
Exact autonomous residue update (Proposition 2)	5	Lean	differenceBarrierD1bCompleteCatalogueStateResidueUpdateCorrectV1, in the module of Main Theorem A
Main Theorems A, B, C	5	Lean	Identifiers in Section 5
Nerode minimality of R_{cat} ; a bounded-radius code for the complete legal future	4	Not claimed	—
<i>Main Result II</i>			
Main Theorem G(i): future sufficiency and autonomous update of $R_K^{(2)}$	6.1, 6.2	Lean	Identifiers in Sections 6.1 and 6.2
Main Theorem G(ii): reachable witness	6.2, 6.3	Lean + certificate	Hypothesis: the local contact certificate; established for <code>t7_ref2_tiles.json</code> by a frozen external audit, which also checks the cell and face literals of the Lean module
Proposition 3 (which states $R_K^{(2)}$ identifies)	6.4	Paper	Uses the at-most-two-cells-per-face property
Counts of the depth-8 search (720, 473 states, 562 classes, no dead ends, 165 classes with different presence ranks)	6.5	Computation	Per-class recomputation does not depend on Proposition 3; that the 562 classes are all identifications within depth 8 does
Nerode minimality of $R_K^{(2)}$ on reachable states	6.6, 10.3	Open	—
<i>Main Result III</i>			
Main Theorem D: strict accessibility loss	7	Lean	Identifier in Section 7

Claim	Section	Status	Basis
Main Theorem E: label equality and the consequences of $C_1 \neq C_2$	7.4	Lean + payload	Payload $C_1 = 2, C_2 = 1$ from WATERWAY_MAINE_SOURCE_AUDIT_v1_1.py on t7_ref2_tiles.json
Same slot with different collateral costs in different states	7.4, 10.3	Open	—
Every dead end has the terminal language $\{\epsilon\}$	7	Paper	Immediate from the definition of $\mathcal{L}(s)$
Saturated states are future-equivalent	7, 8.2	Lean	differenceBarrierD1bCatalogueSaturatedFutureEquivalentV1
<i>Fixed-capacity boundary</i>			
Main Theorem F: no nonempty legal cycle	8.2	Lean	differenceBarrierD1bNoNonemptyLegalCatalogueCycleV1
Main Theorem F: $ w \leq 691 - r_{\text{pres}}(s)$	8.1, 8.2	Lean	differenceBarrierD1bCompleteCatalogueLegalWordLengthLeRemainingCapacityV1
Main Theorem F: finiteness of the reachable subgraph	8.2	Paper	Immediate from the bound; not a separate certificate
Proposition 4 (premature termination)	8.3	Paper	Combines Main Theorems D and F with the audited fact $92 \in \text{CollateralLost}_1$
Presence rank is not a function of the future ($r_{\text{pres}} = 7$ and 8)	8.4	Computation	Exact comparison of $R_K^{(2)}$ for the two states
<i>Companion papers</i>			
$\Phi(s_0) = \Phi(s_2)$ on the audited branch	7.4, 10.1	Cited	[19], conditional on its registered source certificate
$\mathcal{L}(s_0) \neq \mathcal{L}(s_2)$	10.1	Paper	The move i_1 is illegal at s_2 because its target cell is present
Future-equivalent states have equal Φ	10.3	Open	—
Any use of Φ or of the central residue of [21] as a theorem input	10.1	Not claimed	—
<i>Beyond the fixed catalogue</i>			
Release, reuse, catalogue growth, infinite legal runs; physical release, erasure, or recycling of capacity	10.2, 10.3	Not claimed	—

Table A1. Claims and their evidential status.

Statements and Declarations

Funding

This research received no external funding.

Potential Competing Interests

The author declares no competing interests.

Ethics Approval and Consent to Participate

Not applicable. This work is theoretical and formal and does not involve human participants, animals, clinical data, or identifiable personal data.

Data and Code Availability

No experimental or participant data were generated. The frozen catalogue, the formal code, the computational audits, and Supplementary Provenance S1 are deposited at Zenodo, <https://doi.org/10.5281/zenodo.23427992>; each archive is identified below by its SHA256 digest. The formal code corresponding to Main Theorems A–F is provided in the archive:

WATERWAY_LEAN_v11_CORRESPONDING.zip

c9cec7b75daa9e356d29850b69defaf08bef654c84716ee4ec5456c837b02461

The archive contains a theorem-to-certificate map, a SHA256 manifest, the recursive Lean import closure required by the paper (62 Lean files in total, including the entry point), and a self-contained Waterway paper entry point, `WaterwayPaperV11.lean`. The extracted dependency set excludes the P5A/P5B/P5C parity-register branches, the History2I central-residue branch of [21], the PFHC2 release/reuse branch, and the growing-catalogue D1b–G branch. Supplementary Provenance S1 records the theorem identifiers, module provenance, build status, and the boundary between externally audited source facts and Lean-certified theorem consequences. The formal code and computational audits for Main Theorem G are provided in two further archives of the same record:

WATERWAY_v14_S2_LEAN_MAIN_G.zip

08d6c5531d1ee8034fdbfe48d1c525b373a9da815dfa57adc579f96fe513bf3c

WATERWAY_v14_S3_COMPUTATIONAL_AUDITS.zip

f4ff8b41fd82cf1a9fa04e2072a9c60e2d903fb4da97e360daa8dd113f47a07e

Archive S2 contains the recursive A5 import closure of the Main Theorem G modules (34 Lean files, importing only `Mathlib.Tactic` outside A5), the entry point `WaterwayPaperV14MainG.lean`, the axiom audits, a theorem map, and a SHA256 manifest; its closure contains none of the branches excluded above, and the 31 of its files that also occur in the archive for Main Theorems A–F are byte-identical to them. Archive S3 contains the frozen catalogue, Python audit scripts without third-party dependencies, their recorded outputs, and a script that reproduces the audits of Sections 6.5, 7.4, 8.3, and 8.4 and the local contact certificate of Section 6.2. The authoritative 691-cell source catalogue `t7_ref2_tiles.json` used for the Main Theorem E audit and the independent source-audit verifier `WATERWAY_MAINE_SOURCE_AUDIT_v1_1.py` are deposited in the same Zenodo record; the audit input and relevant artefacts are identified there by SHA256 digest.

Author Contributions

M. Numagaki is the sole author and is responsible for the conception of the study, the mathematical analysis, the D1b/GROW-D1 construction, the formalisation strategy, provenance decisions, interpretation of the results, and writing and revision of the manuscript.

Use of Generative AI

OpenAI ChatGPT was used during manuscript preparation for English translation and copyediting, LaTeX formatting, organisation and readability improvements, and assistance in drafting or revising formal-code text. Anthropic Claude was used during the final revision for consistency checks of the manuscript against the companion papers, the Lean source files, and the catalogue data; for re-running the supplied audit scripts; for locating bibliographic details; and for drafting revised English passages and LaTeX edits. The mathematical claims, source identities, formal-proof acceptance decisions, build-status assertions, scientific interpretations, and conclusions were reviewed and approved by the author. The author takes full responsibility for the manuscript and the accompanying formal source package.

Acknowledgements

The author thanks the Lean 4 and Mathlib developer communities for the open-source formal tools used in this work ^{[23][24]}.

References

1. ^a ^bKramer P, Neri R (1984). "On Periodic and Non-Periodic Space Fillings of $\mathbb{R}^m$ Obtained by Projection." *Acta Crystallogr A*. **40**:580–587. doi:[10.1107/S0108767384001203](https://doi.org/10.1107/S0108767384001203).
2. ^aOguey C, Duneau M, Katz A (1988). "A Geometrical Approach of Quasiperiodic Tilings." *Commun Math Phys*. **118**:99–118. doi:[10.1007/BF01218479](https://doi.org/10.1007/BF01218479).
3. ^aBaake M, Grimm U (2013). *Aperiodic Order. Vol. 1: A Mathematical Invitation*. Cambridge University Press.
4. ^aBuganski I, Wolny J, Takakura H (2020). "The Atomic Structure of the Bergman-Type Icosahedral Quasicrystal Based on the Ammann–Kramer–Neri Tiling." *Acta Crystallogr A*. **76**(2):180–196. doi:[10.1107/S2053273319017339](https://doi.org/10.1107/S2053273319017339).
5. ^a ^b ^cKatz A (1988). "Theory of Matching Rules for the 3-Dimensional Penrose Tilings." *Commun Math Phys*. **118**(2):263–288. doi:[10.1007/BF01218580](https://doi.org/10.1007/BF01218580).
6. ^a ^bLe T, Piunikhin S (1995). "Local Rules for Multi-Dimensional Quasicrystals." *Differ Geom Appl*. **5**:10–31.
7. ^a ^bBédaride N, Fernique T (2020). "Canonical Projection Tilings Defined by Patterns." *Geom Dedicata*. **208**:157–175.
8. ^a ^b ^cHammock D, Fang F, Irwin K (2018). "Quasicrystal Tilings in Three Dimensions and Their Empires." *Crystals*. **8**(10):370. doi:[10.3390/cryst8100370](https://doi.org/10.3390/cryst8100370).
9. ^a ^bOnoda G, Steinhardt P, DiVincenzo D, Socolar J (1988). "Growing Perfect Quasicrystals." *Phys Rev Lett*. **60**(25):2653–2656.
10. ^a ^bHann C, Socolar J, Steinhardt P (2016). "Local Growth of Icosahedral Quasicrystalline Tilings." *Phys Rev B*. **94**(1):014113. doi:[10.1103/PhysRevB.94.014113](https://doi.org/10.1103/PhysRevB.94.014113).
11. ^a ^bEvans J (1993). "Random and Cooperative Sequential Adsorption." *Rev Mod Phys*. **65**:1281–1329. doi:[10.1103/RevModPhys.65.1281](https://doi.org/10.1103/RevModPhys.65.1281).
12. ^a ^bRothmund P, Winfree E (2000). "The Program-Size Complexity of Self-Assembled Squares (Extended Abstract)." *ACM*. doi:[10.1145/335305.335358](https://doi.org/10.1145/335305.335358).
13. ^a ^bDoty D (2012). "Theory of Algorithmic Self-Assembly." *Commun ACM*. **55**(12):78–88. doi:[10.1145/2380656.2380675](https://doi.org/10.1145/2380656.2380675).
14. ^a ^bMyhill J (1957). "Finite Automata and the Representation of Events."
15. ^a ^bNerode A (1958). "Linear Automaton Transformations." *Proc Am Math Soc*. **9**(4):541–544. doi:[10.1090/S0002-9939-1958-0135681-9](https://doi.org/10.1090/S0002-9939-1958-0135681-9).
16. ^a ^bPark D (1981). "Concurrency and Automata on Infinite Sequences." doi:[10.1007/BFb0017309](https://doi.org/10.1007/BFb0017309).
17. ^a ^bMilner R (1989). *Communication and Concurrency*. Prentice Hall.
18. ^aCousot P, Cousot R (1977). "Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints." *ACM*. doi:[10.1145/512950.512973](https://doi.org/10.1145/512950.512973).

19. ^a ^b ^c ^d ^e ^f ^g ^h ⁱNumagaki M (2026). "Exact Current-State Compilation of a History-Derived Parity in an $\mathcal{A}_{\leq 5}$ -Seeded State-Transition System." *Qeios*. doi:[10.32388/L7NTKX](https://doi.org/10.32388/L7NTKX).
20. ^AKemeny J, Snell J (1976). *Finite Markov Chains*. Springer-Verlag.
21. ^a ^b ^c ^dNumagaki M (2026). "Admissible History Deformations and Central 2-Torsion Residues: The $\mathcal{A}_{\leq 5}$ Case." *Qeios*. doi:[10.32388/8WF48](https://doi.org/10.32388/8WF48).
22. ^APaige R, Tarjan R (1987). "Three Partition Refinement Algorithms." *SIAM J Comput.* **16**(6):973–989. doi:[10.1137/0216062](https://doi.org/10.1137/0216062).
23. ^Ade Moura L, Ullrich S (2021). *The Lean 4 Theorem Prover and Programming Language*. Springer. doi:[10.1007/978-3-030-79876-5_37](https://doi.org/10.1007/978-3-030-79876-5_37).
24. ^AThe mathlib Community (2020). "The Lean Mathematical Library." Association for Computing Machinery. doi:[10.1145/3372885.3373824](https://doi.org/10.1145/3372885.3373824).

Supplementary data: available at <https://doi.org/10.32388/VB34WS>

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.