The Boolean Tile Basis: A Fractal Foundation for Propositional and Quantified Logic

Paul St. Denis

Preprint — April 2026 • MSC 06E30 • 03B05 • 28A80

Abstract

The sixteen two-variable Boolean connectives form a complete, self-contained generative alphabet in which propositional variables, logical operators, and constants inhabit the same categorical level. This paper develops the **Boolean Tile Basis**: a framework in which each connective corresponds to a 2×2 binary tile, NOR is the sole generator from which all 15 others are constructible, and recursive self-application of any tile produces a unique IFS fractal attractor. The connection to fractal geometry is not contingent: a 2×2 tile is a quadtree node by definition, a quadtree under MSB-first labelling generates Morton path codes natively, and a Morton-addressed quadtree under a fixed Boolean rule is an IFS by structure. The fractal images of Boolean operators are therefore logically necessary, not discovered. In the symmetric {-1, 1} basis, these attractors acquire a spectral interpretation via the Walsh-Hadamard Transform: each tile's Fourier structure determines its fractal geometry, with XOR generating the Sierpiński triangle at Hausdorff dimension log(3)/log(2) — a result grounded in the carry-free isomorphism (XOR = addition mod 2, Sierpiński = the carry-free locus, Lucas's theorem closes the triangle). The n-variable XOR IFS has Hausdorff dimension n−1; at n=3 this is exactly 2, producing a complete surface whose axial projections fill the unit square. The framework extends to many-valued logic: Łukasiewicz operators take clean geometric form in {-1, 1}, and the self-equivalence paradox generates a provably chaotic tent map with Lyapunov exponent log 2. Logical quantification maps onto geometric axis collapse: ∀ is AND-pooling, ∃ is OR-pooling, and the three orders of quantification exhaust the three degrees of freedom of logical space. The paper extends the fractal images of formal systems introduced in St. Denis and Grim (1997) by formalising their algebraic necessity.

PART I — THE LOGICAL FOUNDATION

Propositional completeness • Spectral structure • Many-valued logic • Quantification

1. Introduction

In 1997, St. Denis and Grim demonstrated that formal systems possess fractal images: recursively rendering their truth tables produces self-similar geometric structures of infinite depth, with the Sierpiński gasket emerging as the tautology pattern of classical propositional logic [Ref. 1]. That work established a visual language for studying logic through geometry. The present

paper formalises and extends that language — and, in doing so, reveals that the connection between Boolean logic and fractal geometry is not a discovery but a necessity.

The necessity runs as follows. A 2×2 Boolean tile is a quadtree node by definition: a square region with four sub-regions, each recursively structured the same way. A quadtree, when its nodes are labelled in MSB-first order, generates the Morton path code at every cell by construction. A Morton-addressed quadtree under a fixed Boolean rule is an Iterated Function System by structure. Every IFS has a unique attractor by the Banach fixed-point theorem. The fractal image of a Boolean operator is therefore not found — it is derived. When St. Denis and Grim chose the 2×2 truth table as the generative atom, all of this followed necessarily. The only open question is which attractor each of the 16 operators produces.

The extension of the 1997 work has four movements. First, working in the symmetric {-1, 1} basis rather than {0, 1}, the 16 connectives acquire multilinear polynomial representations and their fractal attractors acquire spectral interpretations — the basis change is not a preference but a necessity: orthogonality of the parity characters $\chi_S(x) = \prod_{i \in S} x_i$ holds exclusively in the symmetric basis. Second, the XOR tile's Sierpiński attractor is explained algebraically: the Sierpiński set is the locus of carry-free binary addition, XOR is carry-free addition, and Lucas's theorem closes the triangle. Third, the framework extends to many-valued logic via Łukasiewicz operators, and logical self-reference generates a precisely characterised chaotic dynamical system. Fourth, logical quantification maps exactly onto geometric axis collapse, and the three orders of quantification exhaust the three natural degrees of freedom of logical space.

Throughout, the fractal images are not illustrations. They are the logical structure rendered spatially: each operator's recursive self-application produces its attractor, and the attractor's geometry encodes the operator's logical character. NOR's single live cell produces a sparse corner attractor. XOR's anti-diagonal structure produces the Sierpiński triangle. The degenerate operators collapse immediately. This is logic made visible — not by analogy but by algebraic necessity.

2. The Self-Contained Tile Ontology

2.1 The Apparent Variable Primitivity Problem

The standard proof that NOR is functionally complete proceeds: $\neg P = P \text{ NOR } P$; $P \vee Q = \neg(P \text{ NOR } Q)$; $P \wedge Q = (\neg P) \text{ NOR } (\neg Q)$. This proof assumes P and Q as pre-given truth-bearing primitives. In a tile-based generative system that builds all structure from a single primitive, this appears circular.

2.2 P and Q Are Tiles

The resolution is immediate. Propositional variables are **already present as specific tiles in the 16-element alphabet:**

- Index 3 (0011): Identity A — passes A unchanged. This is variable P. Polynomial: A
- Index 5 (0101): Identity B — passes B unchanged. This is variable Q. Polynomial: B

Variables, operators, and logical constants (True = Index 15, False = Index 0) all inhabit the same categorical level. The system achieves a **flat ontology**: nothing is borrowed from outside the 16-tile alphabet. Figure 5 illustrates this directly.

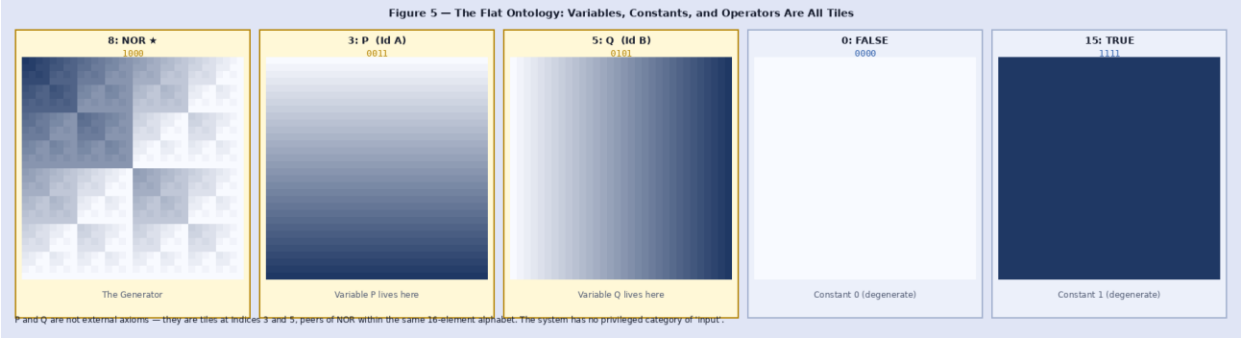

Figure 5. The Flat Ontology. NOR (Index 8), variable tiles P (Index 3) and Q (Index 5), and degenerate constants (Indices 0, 15) are all members of the same 16-tile alphabet at depth m = 5. Nothing is external to the system.

2.3 The Generative Hierarchy

```
NOR  (or NAND)  — the constructive primitive
        ↓  recursive self-substitution
16-Tile Alphabet  — complete logical palette
        ↓  quadtree composition (RBHM)
n-Variable Boolean Functions
        ↓  Fourier / Walsh-Hadamard analysis
Spectral Coefficients  (f^s)
        ↓  geometric limit under RBHM depth → ∞
Fractal Attractors  /  Hausdorff Measures
```

NOR operates on P and Q — all three of which are tiles drawn from the same alphabet. Recursive self-substitution of NOR generates increasingly complex spatial patterns, and the remaining 14 tiles emerge as stable configurations within that recursion.

2.4 NOR as Universal Generator

Figure 6 shows NOR alongside five operators it generates through composition. Each carries a distinct fractal signature — a visual proof that functional completeness translates directly into geometric diversity.

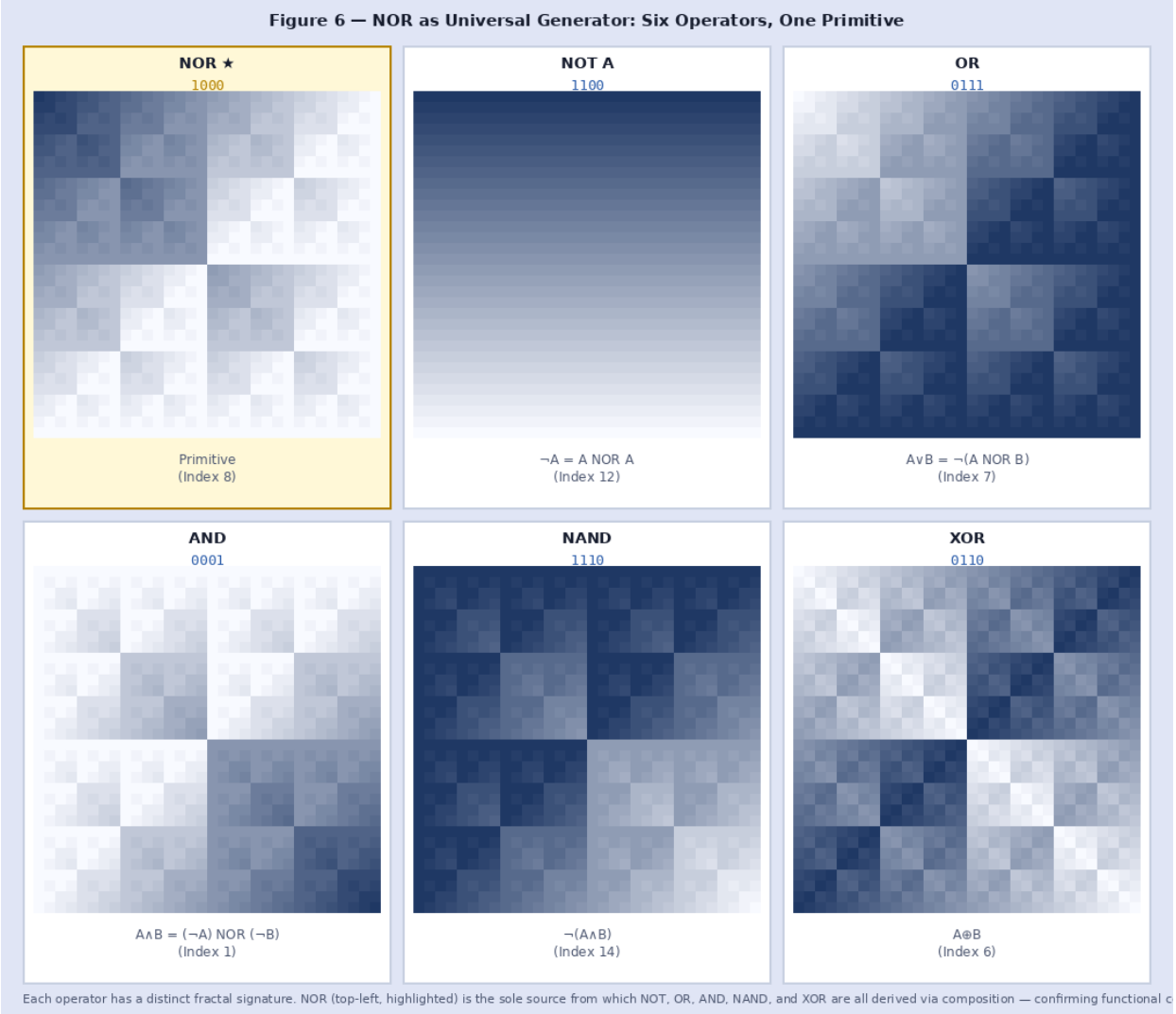

Figure 6. NOR as Universal Generator (depth m = 5). NOR (highlighted) and five operators derived from it: NOT, OR, AND, NAND, XOR. Geometrically distinct fractal attractors confirm functional completeness through spatial structure.

3. The Atomic Tile Mapping and {-1, 1} Polynomials

3.1 Truth Table Index Convention

The 16 two-variable Boolean operators are indexed 0–15 by the decimal value of their 4-bit truth table, where bits are ordered MSB-first as *f(0,0), f(0,1), f(1,0), f(1,1)*. This means:

- bit 3 (MSB) = f(A=0, B=0) — top-left cell of the 2×2 tile
- bit 2 = f(A=0, B=1) — top-right cell
- bit 1 = f(A=1, B=0) — bottom-left cell
- bit 0 (LSB) = f(A=1, B=1) — bottom-right cell

This convention is not universal. An alternative ordering treats A as the least-significant bit, which transposes the middle two rows and shifts several operator indices. Under that alternative, Logical AND appears at Index 4 rather than Index 1. Under the MSB-first convention used throughout this paper, the assignment is determined directly: AND has *f(0,0)=0, f(0,1)=0, f(1,0)=0, f(1,1)=1* — only the LSB is set, giving the 4-bit string 0001 and decimal index 1. Index 4 (binary 0100) corresponds to a function true only when A=0 and B=1, which is Converse Nonimplication (¬A∧B). Readers familiar with the alternative ordering will find the two conventions reconciled by this explicit derivation.

The MSB-first labelling of the four cells — top-left = 00, top-right = 01, bottom-left = 10, bottom-right = 11 — is precisely the Morton (Z-order) labelling of a quadtree node. This is not a coincidence: a 2×2 tile *is* a quadtree node, and the MSB-first truth table index *is* the quadtree path code. The spatial literature calls this structure a Morton code (Morton, 1966); it was native to the 2×2 tile before it was named. Every address, depth, and dimensional property of the framework derived in subsequent sections follows from this identification.

3.2 XOR and XNOR: Resolving a Notation Ambiguity

There exist two common conventions for the bijection between Boolean values and {-1, 1}:

- Convention A (used in this paper): True → −1, False → +1. Bijection: x = 1 − 2b
- Convention B (used in physics/Ising literature): True → +1, False − 1. Bijection: x = 2b − 1

Under Convention A, XOR = AB and XNOR = −AB. Under Convention B, XOR = −AB and XNOR = AB. The two are related by a global sign flip. When Convention B polynomials are quoted in a Convention A context — or vice versa — XOR and XNOR appear transposed. The proof under Convention A (True = −1):

```
XOR is True (−1) when exactly one input is True (−1):
  A=−1, B=−1  →  XOR=False(+1);   AB=(−1)(−1)=+1  ✔
  A=−1, B=+1  →  XOR=True(−1);    AB=(−1)(+1)=−1  ✔
  A=+1, B=−1  →  XOR=True(−1);    AB=(+1)(−1)=−1  ✔
  A=+1, B=+1  →  XOR=False(+1);   AB=(+1)(+1)=+1  ✔
```

Therefore **XOR = AB** under Convention A. The formula −AB yields the sign-flipped pattern (−1,+1,+1,−1), which matches **XNOR** (True when inputs are equal). Both polynomials are correct — in their respective conventions. This paper uses Convention A (True → −1) throughout, consistent with the standard bijection x = 1 − 2b used in Walsh-Hadamard and Fourier analysis of Boolean functions.

3.3 The Complete Operator Table

Table 1 presents all 16 operators with their {-1, 1} multilinear polynomials derived under Convention A. Each polynomial is strictly multilinear — no variable appears squared — a consequence of $x_i^2 = 1$ for all $x_i \in \{-1, 1\}$. Highlighted rows show the functionally complete primitives (NOR, NAND) and the variable tiles (P = Index 3, Q = Index 5).

Idx	Operator	{0,1} Notation	{−1,1} Polynomial	4-bit	Geometric Pattern
0	Contradiction (NULL)	Always 0	1	0000	Empty Grid

Idx	Operator	{0,1} Notation	{−1,1} Polynomial	4-bit	Geometric Pattern
1	**Logical AND ★**	A ∧ B	`(1+A+B−AB)/2`	`0001`	Bottom-Right Cell
2	Mat. Nonimplication	A ∧ ¬B	`(1+A−B+AB)/2`	`0010`	Bottom-Left Cell
3	**Identity A [Var. P]★**	Pass A	`A`	`0011`	Bottom Half
4	Conv. Nonimplication	¬A ∧ B	`(1−A+B+AB)/2`	`0100`	Top-Right Cell
5	**Identity B [Var. Q]★**	Pass B	`B`	`0101`	Right Half
6	Logical XOR	A ⊕ B	`AB †`	`0110`	Anti-diagonal
7	Logical OR	A ∨ B	`(−1+A+B+AB)/2`	`0111`	All Except TL
8	**Logical NOR ★**	¬(A ∨ B)	`(1−A−B−AB)/2`	`1000`	Top-Left Cell
9	Equivalence (XNOR)	A ↔ B	`−AB †`	`1001`	Diagonal (TL+BR)
10	Negation B	¬B	`−B`	`1010`	Left Half
11	Conv. Implication	B → A	`(−1+A−B−AB)/2`	`1011`	All Except TR
12	Negation A	¬A	`−A`	`1100`	Top Half
13	Mat. Implication	A → B	`(−1−A+B−AB)/2`	`1101`	All Except BL
14	**Logical NAND ★**	¬(A ∧ B)	`(−1−A−B+AB)/2`	`1110`	All Except BR
15	Tautology (TRUE)	Always 1	`−1`	`1111`	Solid Grid

★ NOR (Index 8) and NAND (Index 14) are functionally complete primitives; Indices 3 and 5 are variable tiles P and Q. Convention A throughout: True → −1, False → +1 (bijection x = 1 − 2b). XOR = AB and XNOR = −AB — see Section 3.2 for proof.

Figures 1 and 4 show the geometric realisation of these tiles as IFS attractors.

Figure 1. The 16 Atomic Boolean Tiles at depth m = 1. Each 2×2 grid shows live (navy) and void (white) cells. Index ordering follows the MSB-first truth table convention (Section 3.1). Highlighted: AND (Index 1), P and Q (variable tiles at Indices 3 and 5), NOR and NAND (functionally complete primitives at Indices 8 and 14).

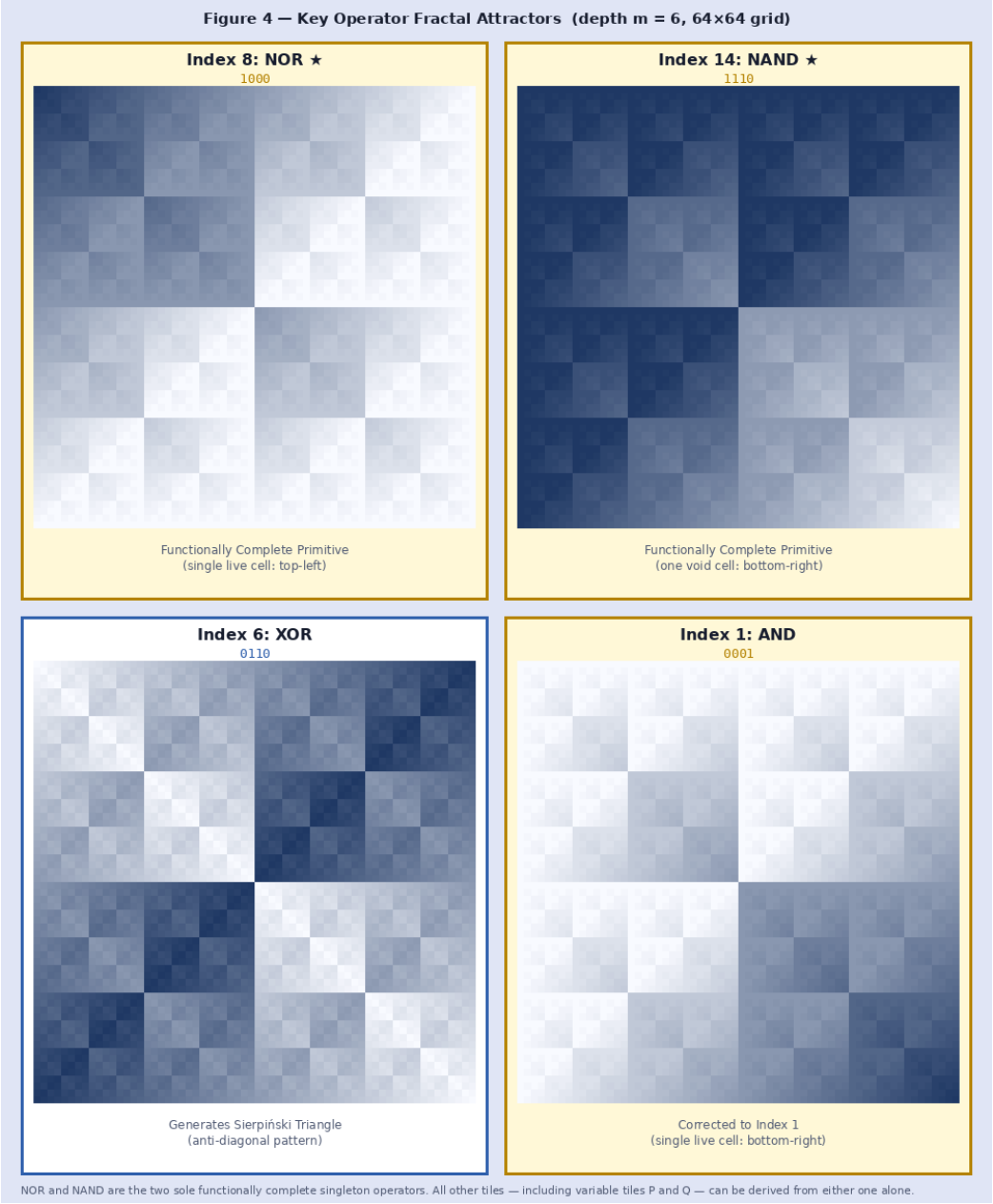

Figure 4. Key Operator Fractal Attractors at depth m = 6 (64×64 grid). NOR (sparse corner attractor), NAND (dense complement), XOR (Sierpiński triangle), AND (complementary corner to NOR).

4. IFS Heatmap Survey: The Fractal Identity of Each Operator

The Iterated Function System underlying the tile visualisations maps each operator's truth table bits to a 2×2 rule matrix: a cell marked 1 (live/True) applies $f_1(p) = 2p + 1$; a cell marked 0 (void/False) applies $f_0(p) = 2p$. At each iteration the grid doubles, and the resulting $2^m \times 2^m$ heatmap encodes the full IFS attractor coloured from white (void) to navy (maximum density). Figure 2 shows all 16 operators at depth 4.

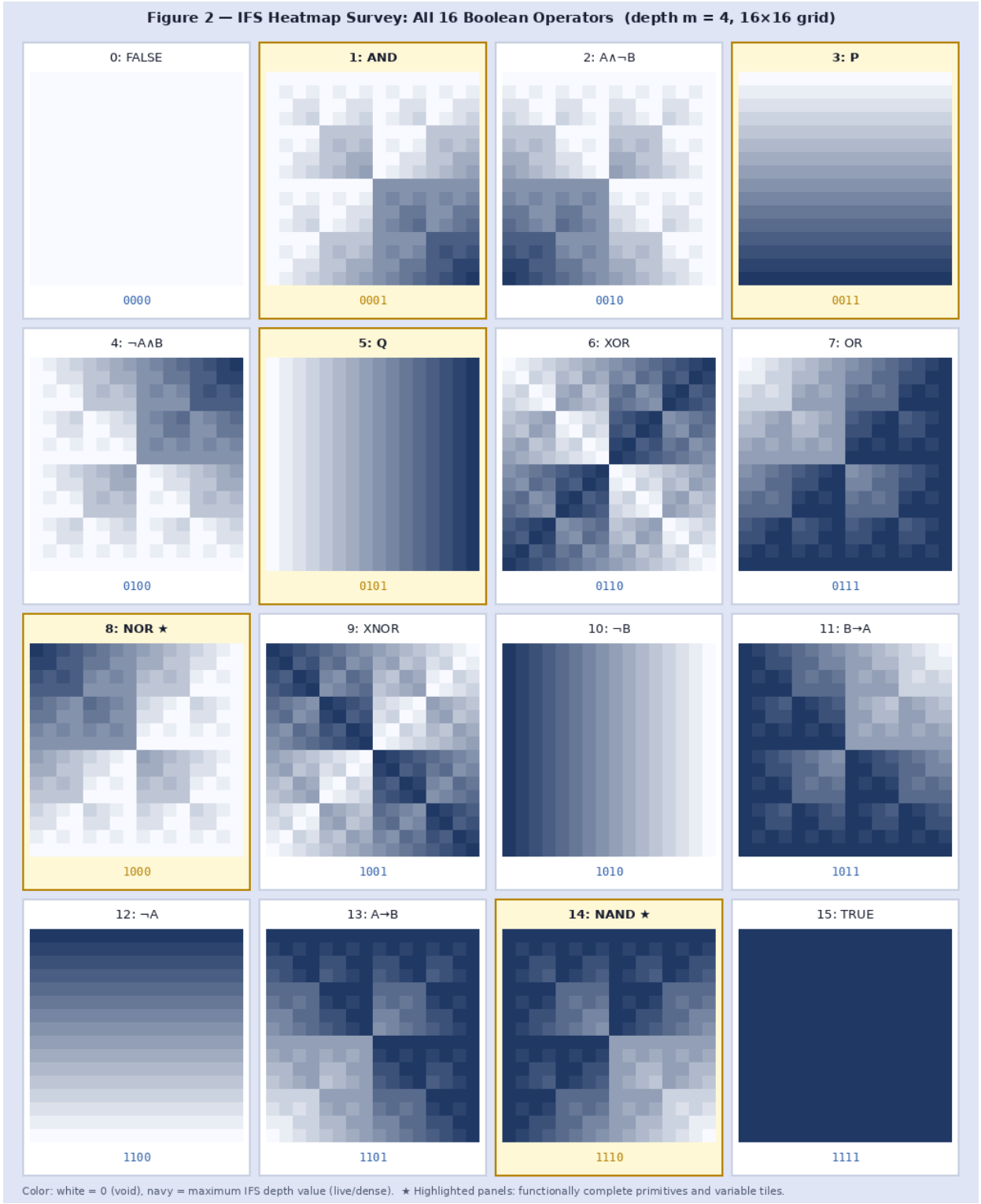

Figure 2. IFS Heatmap Survey: All 16 Boolean Operators at depth m = 4 (16×16 grid). White = void (value 0), navy = maximum IFS density. Operator indices follow MSB-first truth table convention (Section 3.1). Highlighted panels: AND (Index 1), P (Index 3), Q (Index 5), NOR (Index 8), NAND (Index 14).

5. Fractal Emergence: The XOR→Sierpiński Progression

The relationship between Boolean recursion and classical fractal geometry is exact, not metaphorical. Figure 3 traces XOR (Index 6, polynomial AB, rule matrix [[0,1],[1,0]]) across five iterations. At m = 1 the anti-diagonal 2×2 tile is visible; by m = 5 the Sierpiński triangle has fully emerged with Hausdorff dimension log(3)/log(2) ≈ 1.585.

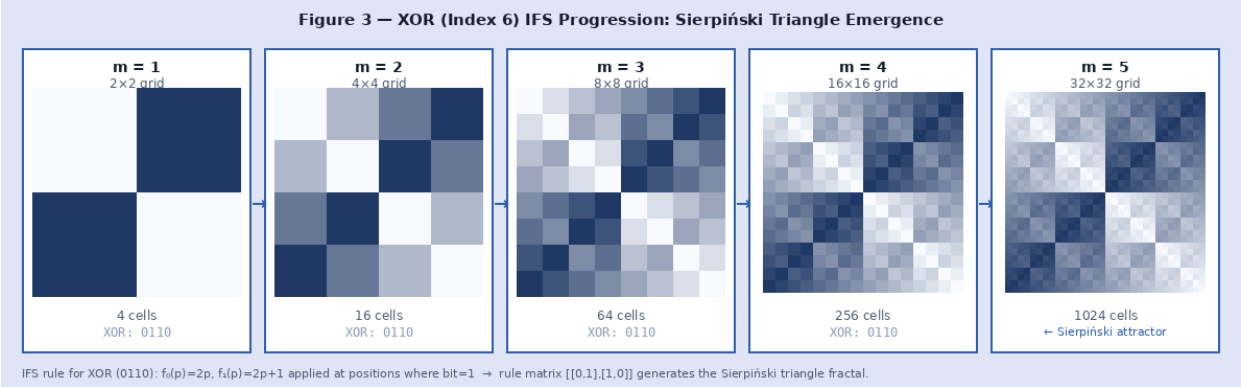

Figure 3. XOR (Index 6) IFS Progression across depths m = 1 to m = 5. The anti-diagonal 2×2 tile recursively generates the Sierpiński triangle. This is the exact IFS attractor of the XOR rule, not an approximation.

This result establishes a concrete bridge between the tile framework and classical fractal geometry. The XOR operator's anti-diagonal structure maps recursively to the Sierpiński gasket by the same IFS mechanism that generates all the other tile attractors shown in Figure 2. The Hausdorff dimension ≈ 1.585 is the measure-theoretic consequence of XOR having one dominant Fourier coefficient — its spectral structure and its geometric structure are two faces of the same mathematical fact.

5.1 The Carry-Free Isomorphism

The XOR tile generates the Sierpiński triangle not merely by visual coincidence but by algebraic necessity. The Sierpiński attractor is the geometric locus of carry-free binary addition.

Theorem 5.1 (Carry-Free Isomorphism). Let X, Y be non-negative integers with binary representations $x_0x_1\ldots$ and $y_0y_1\ldots$. The following three conditions are equivalent:

- X AND Y = 0 (no overlapping 1-bits between X and Y)
- X + Y = X XOR Y (binary addition is carry-free)
- $C(X+Y, X) \equiv 1 \pmod 2$ — by Lucas's theorem, the binomial coefficient is odd exactly when no carries occur

The right-angled Sierpiński triangle on a $2^m \times 2^m$ grid is exactly the set of pairs (X, Y) satisfying these conditions. The voids in the fractal are the address pairs where a carry would occur. The fractal is not the image of XOR — it **is** the truth table of XOR, extended spatially to all address pairs. The paper's central claim — that fractal images are not illustrations of logic but are the logic rendered spatially — here receives its sharpest statement.

5.2 The 3D Extension: XOR as a Complete Surface

The carry-free isomorphism generalises to n dimensions. The n-variable XOR function, applied as the RBHM rule in n-dimensional space, produces an IFS attractor with exactly 2^(n−1) live cells per level — precisely the cells where an odd number of input bits are 1.

Theorem 5.2 (3D XOR Extension). The 3-variable XOR IFS attractor has Hausdorff dimension log(4)/log(2) = 2 exactly. It is the graph of the function Z = XNOR(X, Y), evaluated bitwise: for any (X, Y), there is exactly one Z = ¬X XOR Y placing the triple in the fractal. The axial projection onto any coordinate plane is the complete 2D unit square — the voids of the 2D Sierpiński triangle are filled by the third coordinate.

Proof sketch: At each bit position i, the four input combinations for (x_i, y_i) are (0,0), (0,1), (1,0), (1,1), corresponding to z_i = 1, 1, 1, 0 for the XOR rule in 3 variables (odd parity). Each (x_i, y_i) pair has exactly one valid z_i, so the projection is total. Since the attractor is the graph of a bijection (XNOR) at each bit level, it has exactly 4^m points at depth m and Hausdorff dimension 2. □

n	XOR live cells/level	Hausdorff dim	Projection shadow	Key property
1	1 of 2	0	1D point	Degenerate
2	3 of 4	log3/log2 ≈1.585	1D line segment	Sierpiński triangle
3	**4 of 8**	log4/log2 = 2	Full 2D square ★	Complete surface, XNOR graph
4	8 of 16	3	Full 3D cube	Dimension = n−1 continues
n	2^(n−1) of 2^n	n−1	Full (n−1)D cube	Integer dim only at n=3

★ n=3 is the unique value where the XOR IFS reaches integer Hausdorff dimension 2, producing a complete 2D surface with full-square projections. The 2D Sierpiński voids are filled by the XNOR third coordinate — dimensional folding geometrically completes the fractal.

6. Multilinear Polynomials and the Fourier Expansion

In the {-1, 1} domain, any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ decomposes uniquely as a Fourier expansion over the parity characters $\chi_S(x) = \prod_{i \in S} x_i$, which form a complete orthonormal basis under the uniform inner product $\langle f,g \rangle = 2^{-n} \sum_x f(x)g(x)$:

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \chi_S(x)$$

The Fourier coefficient $\hat{f}(S) = \langle f,\chi_S \rangle$ quantifies the correlation between f and the parity function χ_S. This orthogonality holds **exclusively in the {-1, 1} basis** because it relies on symmetric cancellation of positive and negative terms. Parseval's Identity states that $\sum_S \hat{f}(S)^2 = 1$ for any $f : \{-1,1\}^n \to \{-1,1\}$. The Fourier coefficients are real numbers in [−1,+1] — they are the first path by which the discrete Boolean tiles generate real-valued limits (Section 7).

The **FKN Theorem** states that if a balanced (or near-balanced) Boolean function has almost all its Fourier weight on degree-1 coefficients, it is structurally close to a dictator (a function depending on a single coordinate) or its negation. The balance condition is essential: without it the conclusion does not hold. The **KKL Theorem** establishes that every balanced Boolean function on n variables has at least one variable with influence $\Omega(\log n / n)$.

7. The Walsh-Hadamard Transform

7.1 The AND Tile Is H_1

The base Hadamard matrix $H_1 = [[1, 1], [1, -1]]$ is not imported from harmonic analysis. It is the AND tile. Under Convention A (True = −1, False = +1), the AND truth table written as a 2×2 matrix with rows indexed by A and columns by B is:

```
        B=+1   B=−1
A=+1 [   +1     +1   ]
A=−1 [   +1     −1   ]
```

The bottom-right entry is −1 because AND(−1, −1) = True = −1. Every other entry is +1 because AND is False unless both inputs are True. This matrix is H_1 exactly. The Hadamard matrix and the AND tile are the same 2×2 object.

This is not an accident. H_1 encodes the unique degree-1 Walsh character that pairs the base Boolean operator with the spectral transform. The Kronecker product structure — $H\_n = H_1 \otimes H_1 \otimes \cdots \otimes H_1$ (n times) — is the n-variable generalisation of the 2-variable AND tile. The Walsh-Hadamard Transform is therefore native to the tile alphabet, not imported from outside it.

7.2 Two Readings of the Same Tile

Every 2×2 tile admits two interpretations, producing two different mathematical objects from the same matrix:

- As a truth table (point-wise evaluation at $\{-1,+1\}^2$): apply the tile rule cell by cell. Iterate via the RBHM quadtree. The output is the IFS fractal attractor — the tile's geometric identity. This is the interpretation used in Sections 2–6.
- As a linear operator (matrix multiplication acting on real vectors): apply H_1 to a vector of function values. Iterate via Kronecker product. The output is a real-valued spectrum — the tile's spectral identity. This is the Walsh-Hadamard Transform.

The fractal images and the Fourier analysis are therefore two readings of the same recursive structure. The IFS and the WHT are not separate topics that happen to be related — they are the same 2×2 tile, iterated in two different senses. The **point-wise sense** produces geometry (attractors in position space). The **linear sense** produces spectra (coefficients in frequency space). Both are exact consequences of the AND = H_1 identification.

7.3 The Kronecker Iteration and Its Real-Valued Limit

The tiles at depth m=1 have entries in {-1,+1}. The Kronecker product $H\_n = H_1^{\otimes n}$ has entries in {-1,+1} for every finite n. As $n \to \infty$, the normalised matrix $(1/\sqrt{(2^n)})H\_n$ converges to a continuous operator — the Walsh-Fourier transform on the Cantor group $\{-1,+1\}^\infty$. Its entries are real numbers in [−1,+1].

The Fourier coefficients $\hat{f}(S) = \langle f, \chi\_S \rangle$ of any Boolean function are real numbers, bounded by Parseval's Identity $\Sigma\_S \hat{f}(S)^2 = 1$. They are not truth values. Classical fast WHT on a 2^n-entry vector runs in $O(n \cdot 2^n)$ operations. A quantum processor applies Hadamard gates to all n qubits simultaneously, achieving O(1) circuit depth — but still requires O(n) total gates and $O(2^n)$

measurements to read out all amplitudes. The speedup is exponential in sequential depth, not a collapse to constant total work. This is the first of three converging paths from discrete Boolean vertices to real-valued limits:

- Spectral path: WHT iteration (Kronecker product of H_1) → real Fourier coefficients $\hat{f}(S) \in [-1,+1]$
- Measure-theoretic path: IFS iteration at depth m → uniform measure over live cells → as m → ∞, weak convergence to a continuous probability measure on the fractal attractor (Conjecture C2)
- Łukasiewicz path: tile operators extended continuously from the Boolean vertices $\{-1,+1\}^n$ to the full hypercube interior $[-1,+1]^n$ — treated in Section 11

All three paths terminate at the same object: the continuous extension of the Boolean tile basis to the interior of [−1,+1]^∞. They are not three separate topics — they are the spectral, geometric, and logical faces of the same limiting structure. The WHT coefficients live in this limit. The IFS invariant measure lives on its fractal boundary. The Łukasiewicz operators fill its interior.

This is why the {-1, 1} basis is not merely convenient: it is the unique basis in which these three convergences happen simultaneously, and in which the AND tile serves as both the Boolean truth table and the harmonic analysis kernel.

Figure 7 — Walsh-Hadamard Spectrum: XOR vs. AND (n = 4 variables, 16 coefficients)

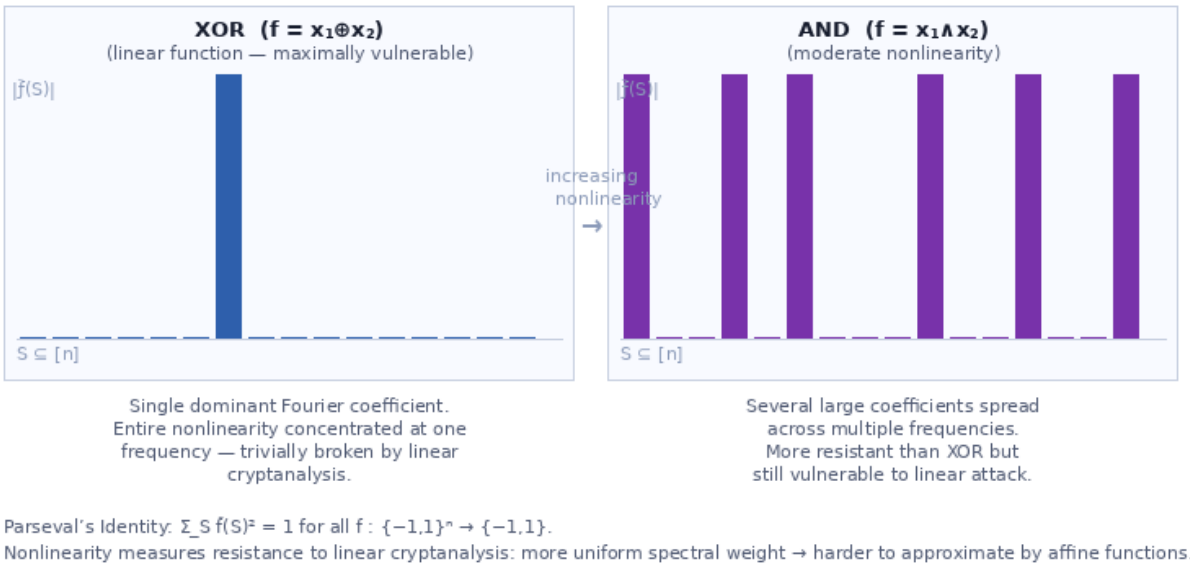

Figure 7. Walsh-Hadamard Spectrum for n = 4 variables. Left: XOR — single dominant Fourier coefficient, the signature of a linear function. Right: AND — several large coefficients spread across multiple frequencies. Parseval's Identity ($\Sigma\, \hat{f}(S)^2 = 1$) conserves total spectral energy. Both spectra are produced by applying the AND tile (= H_1) as a linear operator via Kronecker product.

The distribution of spectral energy directly determines resistance to linear cryptanalysis. Nonlinearity $NL(f) = 2^{n-1} - \frac{1}{2} \cdot \max_S |\hat{f}(S)|$ quantifies this: functions whose maximum Walsh coefficient is smaller are farther from any affine approximation. XOR's spectrum — one coefficient carrying all the weight — makes it maximally linear. More uniform spectral distributions correspond to higher nonlinearity, which is why the Walsh-Hadamard Transform is the primary tool in S-box design for block ciphers.

8. The Tile Framework: Structure and Completeness

8.1 Uniform-Label and Level-Varying Regimes

The RBHM quadtree operates in two distinct regimes. In the **uniform-label** regime every node carries the same tile, producing the self-similar fractals in Figures 1–6. In the **level-varying** regime depth level d carries tile f_d, giving 16^m distinct surfaces at depth m. The level-varying regime is the mechanism for encoding arbitrary Boolean surfaces, and is the subject of Conjecture C1''.

8.2 The Exact Counting Identity

A common claim in early formulations was that 16^m equals the full number of 2^m-variable Boolean functions. This requires care for $m \geq 2$:

$$16^m = 2^{(4m)}$$

$$\text{Full space: } 2^{(4^m)} \quad \text{(tower of exponents, not linear exponent)}$$

Figure 13 — The Exact Counting Identity: Level-Varying Subspace vs Full Function Space

m	16^m (level-varying)	2^(4^m) (all functions)	Ratio
1	2^4 = 16	2^4 = 16	1/1 (complete)
2	2^8 = 256	2^16 = 65,536	1/256 ← structured subspace
3	2^12 = 4,096	2^64 ≈ 1.8×10^{19}	$1/4.5\times10^{15}$ ← structured subspace
4	2^16 = 65,536	2^256 ≈ 1.2×10^{77}	negligible ← structured subspace

What the identity says

16^m = (2^4)^m = 2^(4m)

This is distinct from:
2^(4^m) (the full space)

At m=1: equal (construction is complete at depth 1)

At m=2:
Level-varying: 2^8 = 256
Full space: 2^16 = 65,536

Level-varying addresses a structured subspace.
NOR pipeline reaches all surfaces (fractal depth ∞).

The space is inexhaustible, not incomplete.

Figure 13. The exact counting identity. At m=1 the level-varying construction is complete (16 = 16). For m ≥ 2 it addresses a structured subspace: at m=2, 256 out of 65,536 possible 4×4 binary surfaces. The NOR pipeline covers the full space; level-varying encoding alone does not.

At m = 1 the construction is complete (16 = 16). For m ≥ 2 the level-varying encoding addresses an exponentially vanishing fraction of the full function space. This is **fractal inexhaustibility**: at any finite depth there is always more structure deeper in the fractal. The NOR pipeline reaches every surface at sufficient depth. Conjecture C1'' addresses whether distinct level-varying encodings always produce distinct surfaces.

8.3 A Note on Thermodynamic Isomorphisms

The {-1, 1} encoding aligns naturally with statistical physics. In the Ising model of ferromagnetism, spin variables $\sigma_i \in \{-1,+1\}$ interact via the Hamiltonian $H(\sigma) = -\Sigma J_{ij}\sigma_i\sigma_j - h\Sigma\sigma_i$, which is

the thermodynamic twin of the degree-2 Fourier expansion of a Boolean function (Cipra, 1987). Finding the Ising ground state is equivalent to solving MAX-SAT, and the bijection $\sigma_i = 1 - 2x_i$ converts the Hamiltonian into a QUBO problem, connecting Ising machines to Boolean satisfiability (Lucas, 2014). This isomorphism has driven the development of dedicated Ising machines and neuromorphic accelerators. It is noted here as a consequence of the {-1, 1} symmetry; the logical framework in this paper does not depend on it.

8.4 NOR Universality

Theorem 9.1 (NOR Universality). For any $n \geq 1$ and any n-dimensional hypercubic Boolean surface S, there exist independent n-dimensional NOR fractals at sufficient depth whose downsampled output exactly reproduces S.

Proof sketch: any Boolean function is expressible as a finite NOR circuit (functional completeness). The n-dimensional quadtree depth corresponds directly to NOR circuit depth. □

The constructive mechanism — the downsampling pipeline, the n-dimensional generalisation, and the full bit budget formalism — is developed in detail in the companion geometry paper (St. Denis, in preparation). For the logical purposes of this paper, NOR universality is the key fact: the fractal space is inexhaustible but complete.

9. Dimensional Folding: The Fixed Point of Quantification

The quantifier hierarchy from Section 12 raises a natural question: after you have applied quantifiers repeatedly — collapsing variables, then tile-label axes, then bit-planes — where do you land? The dimensional folding theorem answers this: you always land in a **3-dimensional logical space**. The three parameters n (variable count), m (recursion depth), and W (value range) are exactly the three degrees of freedom of logical space, and any finite logical surface at any depth is isomorphic to a 3D surface. The folding is not a geometric convenience; it is the fixed point of the quantification process.

9.1 The Folding Bijection

Theorem 9.1 (Dimensional Folding Isomorphism). Let S be an n-dimensional hypercubic Boolean surface at resolution depth m with bit-depth W, and total bit budget $B = n \cdot m + W$. Let $m'' = \lfloor B/3 \rfloor$ and $W'' = B \bmod 3$. Then S is isomorphic to a 3D hypercubic surface at depth m″ with bit-depth W″, via the bijection that reinterprets the B-bit cell address as a 3D quadtree path code.

The bijection uses the native address of the quadtree. The path from the root of the RBHM quadtree to any cell at depth m is a sequence of n-bit labels — one per level, recording which of the 2^n sub-quadrants was entered. Concatenating these labels produces the nm-bit cell address. Re-reading that same B-bit string as three interleaved streams of m″ bits each — one stream per spatial axis — gives the 3D interpretation. No external curve or algorithm is required: this redistribution of the quadtree's own path code is the entire bijection.

The structure produced by reading two interleaved bit streams as a 2D address is known in the literature as a Morton code or Z-order curve (Morton, 1966). Within the RBHM framework, Morton encoding is not an import — it is the quadtree's native address, seen from the outside.

9.2 The Same Object in Three Dimensions

Figure 11 shows the same NOR fractal at bit budget B = 6 expressed in two representations: a 2D 8×8 grid (n = 2, m = 3) and a 3D 4×4×4 cube (n = 3, m = 2). Both encode the same 64 binary values. The address redistribution is the native quadtree path code re-read across different numbers of axes: the same 6-bit string, partitioned 2+2+2 for 3D rather than 3+3 for 2D.

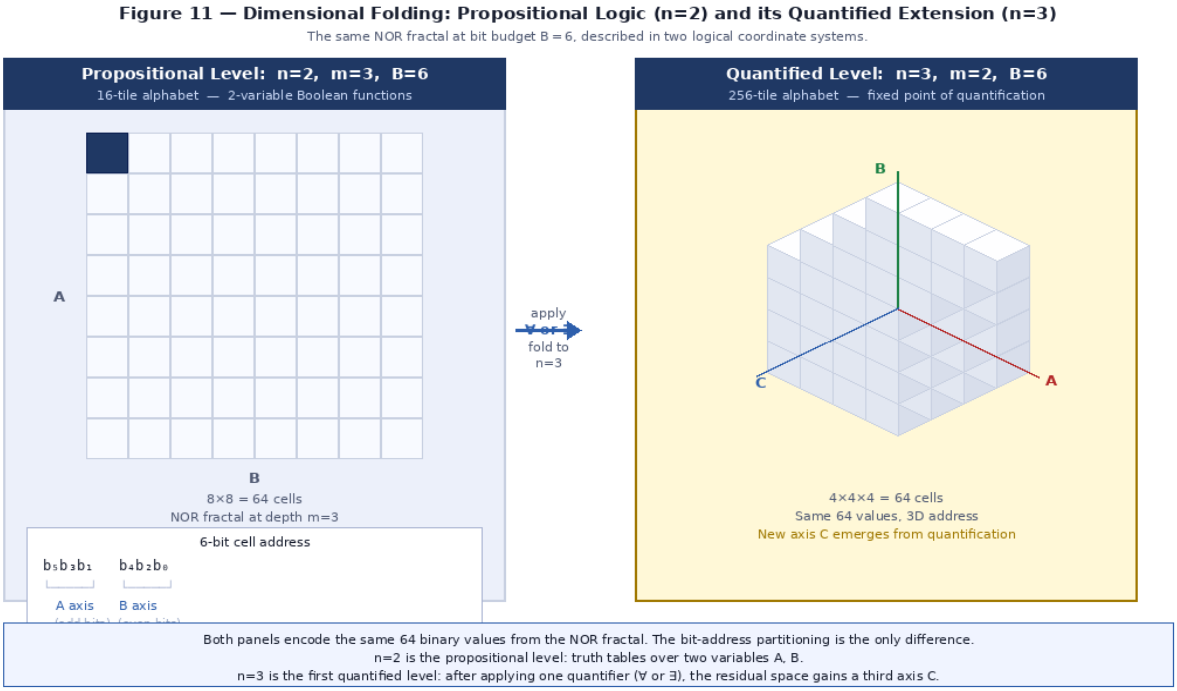

Figure 11. The Dimensional Folding Isomorphism at B = 6. Left: 2D grid (n=2, m=3), 8×8 = 64 cells. Right: 3D cube (n=3, m=2), 4×4×4 = 64 cells. The bit-address boxes show how the same 6-bit quadtree path code is partitioned across dimensions. All cells are the NOR fractal in different coordinate systems — the folding reinterprets the native path address, not an external curve.

The folding is not a lossy projection or an approximation. It is a *change of coordinates* on the same discrete structure, exactly as the {-1, 1} basis change is a change of coordinates on the same Boolean algebra. In both cases the underlying object is unchanged; only the language used to describe it shifts.

9.3 Why 3D is the Natural Reduction Target

Three independent reasons converge on 3D as the privileged reduction target:

- Topological privilege. 3D is the minimum dimension with enclosed volume. In 1D there are only points and intervals; in 2D there is surface but no interior; in 3D enclosed space first appears. Knot theory is non-trivial in 3D (knots cannot be untied) but trivial in 4D (every knot untangles). The Betti numbers H_0, H_1, H_2 are all simultaneously non-trivial only in 3D and above.
- The byte coincidence. The 3D tile alphabet has size $2^{(2^3)} = 256 = 2^8$ — exactly one byte. Digital memory is byte-addressed. The 3D Boolean tile alphabet is the atom of digital computation, matching the standard memory model with no remainder. No other dimension achieves this: n=2 gives 16 (too small), n=4 gives 65,536 (too large for a single addressable unit).
- The RGB/colour connection. Digital colour is represented as three independent one-byte channels (R, G, B). This is precisely three independent 3D bit-planes at W = 1 — exactly

the n = 3, W = 1 allocation of the tile framework. The folding from n-D to 3D is the abstract generalisation of how any volumetric dataset can be stored as a colour volume.

- The XOR dimension coincidence. The n-variable XOR IFS has Hausdorff dimension n−1 (Theorem 5.2). At n=3, this is exactly 2 — a complete 2D surface with no voids, whose axial projections fill the full square. No other value of n produces an integer dimension: n=2 gives 1.585, n=4 gives 2.585. This is a logical reason why 3D is the natural folding target: it is the unique dimension where the XOR fractal becomes a complete surface, and the Sierpiński voids of the 2D attractor are algebraically closed by the XNOR third coordinate.

Figure 12 — Why 3D? The Privileged Status of n=3 in the Dimensional Hierarchy

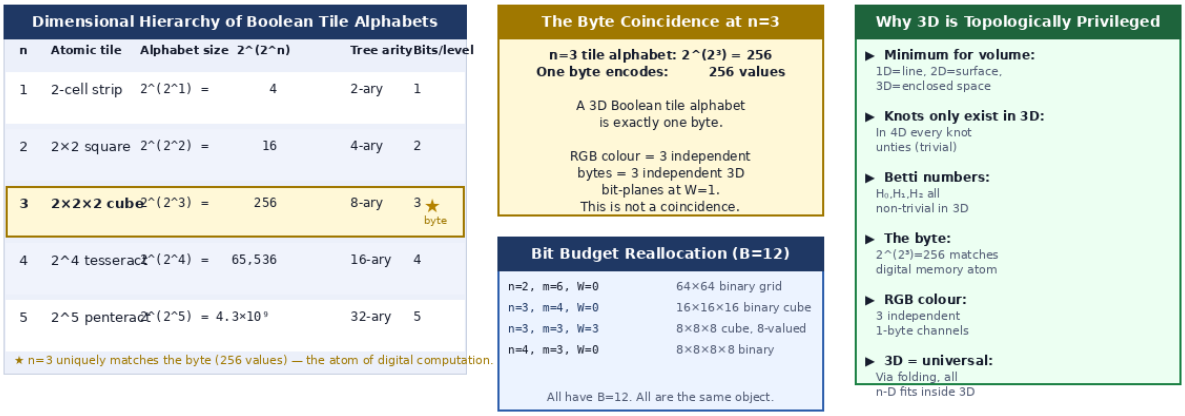

Figure 12. Why 3D is privileged. Left: the dimensional hierarchy showing that n=3 uniquely matches the byte (256 tile functions). Centre: the byte coincidence and bit-budget reallocation examples. Right: the six independent reasons why 3D is the natural reduction target for the dimensional folding isomorphism.

9.4 The Folding Connection to Łukasiewicz Chaos

The dimensional folding is structurally identical to the Łukasiewicz tent map 2|x| − 1 from Section 11, scaled to higher dimension. The tent map folds a 1D interval onto itself at each iteration. The IFS quadtree folds a 2D region into four sub-regions at each level. The dimensional folding folds an n-dimensional address space into a 3D address space by redistributing axis bits.

All three are instances of the same operation: **recursive folding of a bounded domain** — preserving all information content while changing how the dimensions are named. The chaos in p ↔ ¬p is the 1D case; the Sierpiński triangle is the 2D case; the dimensional folding isomorphism is the n-D generalisation. They share a common mathematical skeleton.

9.5 A Note on Locality

The native quadtree path code preserves locality within each quadtree quadrant but can place adjacent cells far apart across quadrant boundaries. Whether an alternative address remapping — such as the Hilbert space-filling curve, which minimises the maximum locality distortion — could provide a bounded locality guarantee independently of n remains an open question (Conjecture C5, Section 18).

10. When to Use Each Basis: The Operational Divergence

Given the exhaustive spectral power of the {-1, 1} space, a natural question arises: why do researchers, textbooks, and algorithms continuously revert to the {0, 1} basis? The answer is not mathematical incompleteness — the {-1, 1} domain is provably universal. The reversion is driven by three inescapable operational realities.

10.1 Axiomatic Simplicity and the Boolean Ring

In the {0, 1} basis, the fundamental laws of Boolean algebra reduce to intuitive visual arithmetic. The Annulment Law ($A \wedge 0 = 0$) is immediately obvious: a variable ANDed with False is always False. In the {-1, 1} polynomial basis the equivalent statement requires expanding the full polynomial for AND, cross-multiplying with the constant polynomial for FALSE ($f = 1$), and relying on algebraic coefficient cancellation to arrive at the same conclusion.

This is mathematically elegant in the context of a Fourier analysis, but is *unnecessary* cognitive and computational overhead for determining whether a logic gate is open or closed. The {0, 1} Boolean ring is the right tool for axiomatic simplification; the {-1, 1} polynomial ring is the right tool for spectral and continuous analysis.

10.2 Combinatorial Density: The Game of Life Problem

The most pronounced divergence occurs when a system depends on geometric density or neighbour-counting — perfectly illustrated by Conway's Game of Life. The rule set requires counting exactly how many of the 8 neighbours of each cell are alive. Table 3 shows the computational consequence of each basis choice.

Basis	Neighbourhood Sum Formula	Consequence
{0,1} Basis	$N = \Sigma\ C_i$ (direct sum)	N=3 means exactly 3 live neighbours. Trivial, O(n) arithmetic.
{−1,1} Basis	$N = \Sigma\ C_i$ (with −1=live)	N=0 could be 4 live + 4 dead. Cancellation makes counting ambiguous.
Fix in {−1,1}	$N = (8 - \Sigma\ C_i)/2$	Correct but adds cognitive overhead and two extra operations per cell.

Table 3. Neighbourhood counting in Conway's Game of Life under each basis. The {0,1} sum is direct and unambiguous; the {−1,1} sum is ambiguous without correction.

In the {0, 1} basis, $N = \Sigma C\_i$ is a direct summation. A sum of 3 means exactly 3 live cells, unambiguously. In the {-1, 1} basis (where -1 = live), a sum of 0 could represent 4 live and 4 dead cells — the symmetric cancellation that makes {-1, 1} so powerful for Fourier analysis is precisely what makes it catastrophic for population counting. The correction exists but introduces overhead on every cell of every generation. For any algorithm fundamentally dependent on neighbour enumeration, the {0, 1} basis is the correct choice.

10.3 Algorithmic Sparsity and Matrix Optimisation

In massive datasets and quadtrees, the {0, 1} basis enables sparse data structures: a value of 0 (False) terminates recursive processing and compresses trivially. Standard sparse matrix formats (CSR, COO) exploit this to store only non-zero entries.

In the {-1, 1} space, False is encoded as +1 — a structurally active, non-zero integer. There is no numerical zero to terminate recursion or collapse a sparse array. Every false state must be stored, calculated, and routed through the processor. Sparse matrix compression fails entirely. To bridge advanced physics frameworks back to computational efficiency, optimisation analysts use the variable transformation $\sigma\_i = (1 - \sigma\_i) / 2$, converting active Ising states back into {0, 1} QUBO form and restoring algorithmic sparsity.

10.4 The Resolution

{-1, 1} is the domain of ANALYSIS
Use for: spectral decomposition, Fourier coefficients, Ising energy, orthogonality proofs, cryptographic nonlinearity, EBM training.

{0, 1} is the domain of EXECUTION
Use for: neighbour-counting, sparse matrix storage, circuit truth tables, combinatorial enumeration, hardware implementation.

The two domains are perfect topological complements. Mastery of discrete logic requires knowing exactly when the problem demands each.

Neither basis supersedes the other. The {-1, 1} representation is the mathematically natural home for spectral decomposition, orthogonality, thermodynamic modelling, and fractal geometry — all of which this paper demonstrates. The {0, 1} representation is the natural home for combinatorial enumeration, sparse computation, and circuit-level logic. Recognising which domain the problem belongs to is the essential skill.

11. Łukasiewicz Fuzzy Logic and the {-1, 1} Basis

Section 7 identified three converging paths from the discrete Boolean vertices $\{-1,+1\}^n$ to real-valued limits: the spectral path (Fourier coefficients via WHT), the measure-theoretic path (IFS invariant measure via weak convergence), and the Łukasiewicz path (tile operators extended continuously to the hypercube interior). The first two have been developed. This section develops the third, and shows that all three terminate at the same object.

The Boolean tile framework operates on the *vertices* of the n-dimensional hypercube — the 2^n corners of $\{-1, 1\}^n$ where truth values are crisp. Łukasiewicz infinite-valued logic extends this to the *interior* of the hypercube, assigning real-valued truth degrees throughout $[-1, 1]^n$. Rephrasing Łukasiewicz's system in the {-1, 1} basis reveals connections invisible in its standard [0, 1] formulation. The same tile operations that generate fractal attractors at the vertices, when extended continuously to the interior, produce the saturating arithmetic of neural network activations, the chaotic dynamics of self-reference, and the probability measures of Conjecture C2.

The three paths are one structure seen from three angles. The WHT coefficients (spectral path) live in the limit space $[-1,+1]^{\wedge\infty}$. The IFS invariant measure (measure-theoretic path) is supported on the fractal boundary of the same space. The Łukasiewicz operators (logical path) fill its interior.

Every tile simultaneously determines all three: which fractal it generates (IFS), which spectrum it has (WHT), and how it interpolates into the interior (Łukasiewicz).

11.1 The Łukasiewicz Operators in {-1, 1}

The standard Łukasiewicz system is defined over truth values $p, q \in [0, 1]$. Applying the bijection $x = 1 - 2p$ (Convention A: True → −1, False → +1) converts every operator into the {-1, 1} domain. The results are structurally cleaner than their {0, 1} originals:

- Negation: $\neg p = 1-p \rightarrow -x$ (pure sign flip; reflection through the origin)
- Strong disj.: $\min(1, p+q) \rightarrow \max(-1, x+y-1)$ (clamped sum with offset −1)
- Strong conj.: $\max(0, p+q-1) \rightarrow \min(+1, x+y+1)$ (clamped sum with offset +1)
- Implication: $\min(1, 1-p+q) \rightarrow \min(+1, 1-x+y)$
- Equivalence: $1-|p-q| \rightarrow |x-y| - 1$ (distance-based; always ≤0)

The negation result is immediate but significant: the {0, 1} negation $x \mapsto 1 - x$ is an affine map (translation plus reflection). The {-1, 1} negation $x \mapsto -x$ is a purely *linear* map — geometric reflection through the origin, with no translation term. This makes {-1, 1} the natural home for any logic where negation is meant to represent exact duality: the negation of a proposition is its mirror image, not its affine shift.

The two main connectives form a symmetric pair: strong disjunction and strong conjunction differ only in sign of offset (−1 vs +1) and direction of clamping (max vs min). They are related by negation exactly as De Morgan's laws require, and this De Morgan duality is *visible* in the {-1, 1} formulas in a way that is obscured in {0, 1}. Furthermore, clamped sums of the form $\max(-1, x+y-1)$ are exactly the **saturating addition** used in neural network activations and quantised arithmetic — the Łukasiewicz operators in {-1, 1} are the natural continuous relaxation of Boolean gates into activation functions.

A critical contrast with Boolean {-1, 1}: in classical Boolean logic, XNOR = −xy and XOR = xy (the product, a bilinear extension). In Łukasiewicz logic, equivalence = $|x - y| - 1$ (a metric, an absolute-value extension). Both formulas agree on the four vertices $\{-1, 1\}^2$ but extend differently into the continuous interior — the Boolean extension is multiplicative and smooth; the fuzzy extension is distance-based and piecewise. Figure 9 shows both systems as heatmaps over the full $[-1, 1]^2$ square.

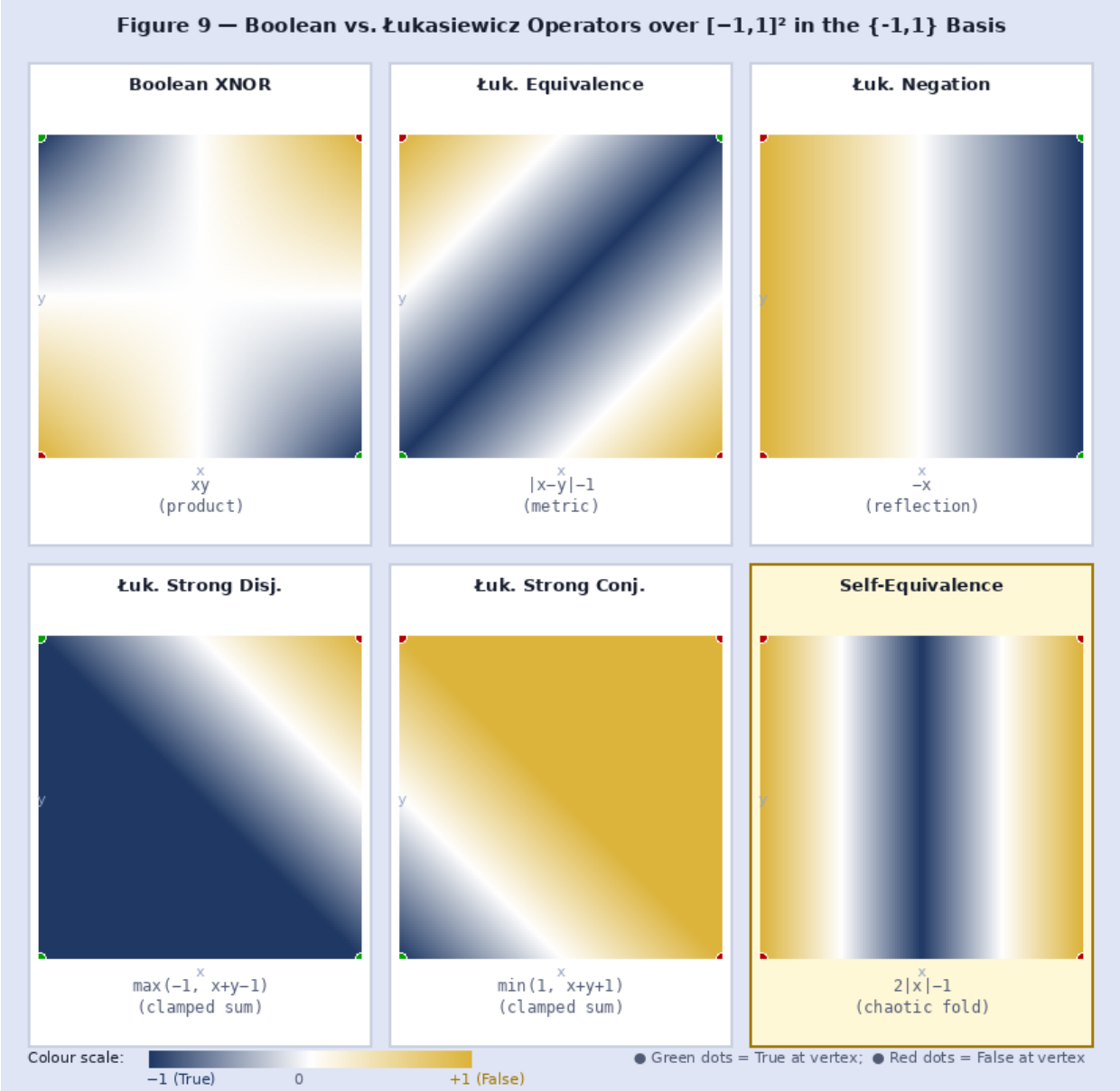

Figure 9. Boolean vs. Łukasiewicz operators as heatmaps over [−1, 1]². Colour: navy = −1 (True), white = 0, gold = +1 (False). Green/red dots mark Boolean vertex values. Boolean XNOR (xy, bilinear) and Łukasiewicz equivalence (|x−y|−1, metric) agree on vertices but interpolate differently through the interior. Self-equivalence (bottom-right, highlighted) generates the chaotic tent map.

11.2 The Chaotic Self-Equivalence Map

Your observation is precise and deep. Consider asking Łukasiewicz logic to evaluate $p \leftrightarrow \neg p$ — a proposition equated with its own negation. In {0, 1}: $1 - |p - (1 - p)| = 1 - |2p - 1|$, the classical tent map. In {-1, 1}, substituting $x = 1 - 2p$ and converting:

$$f(x) = 2|x| - 1$$

This is an absolute-value fold, symmetric about the origin. The map sends both −1 and +1 to +1 (False), and has a unique minimum of −1 (True) at $x = 0$ — the logical midpoint of maximal

indeterminacy. The {0, 1} tent map $1 - |2p - 1|$ and the {-1, 1} map $2|x| - 1$ are negations of each other under the bijection, mapping the same underlying phenomenon.

Both maps are **maximally chaotic among piecewise linear maps on their domain**, with Lyapunov exponent $\lambda = \log 2$. The chaos is not incidental — it is the mathematical signature of asking a continuous logic system to equate a proposition with its own negation. The infinite recursive self-reference implicit in that demand generates an orbit that never settles and is sensitive to any perturbation of the initial value. Figure 8 shows the map, its iterated compositions, and the chaotic cobweb diagram of a typical orbit.

The fixed points of $2|x| - 1$ are where $f(x) = x$: solving $2|x| - 1 = x$ gives $x = 1$ (the False boundary) and $x = -½$ ... wait, for $x < 0$: $-2x - 1 = x \rightarrow x = -1/3$. Both fixed points are unstable: any small perturbation sends the orbit spiralling chaotically. This is the formal proof that $p \leftrightarrow \neg p$ has no stable truth value in Łukasiewicz logic — it is logically undecidable in the continuous setting.

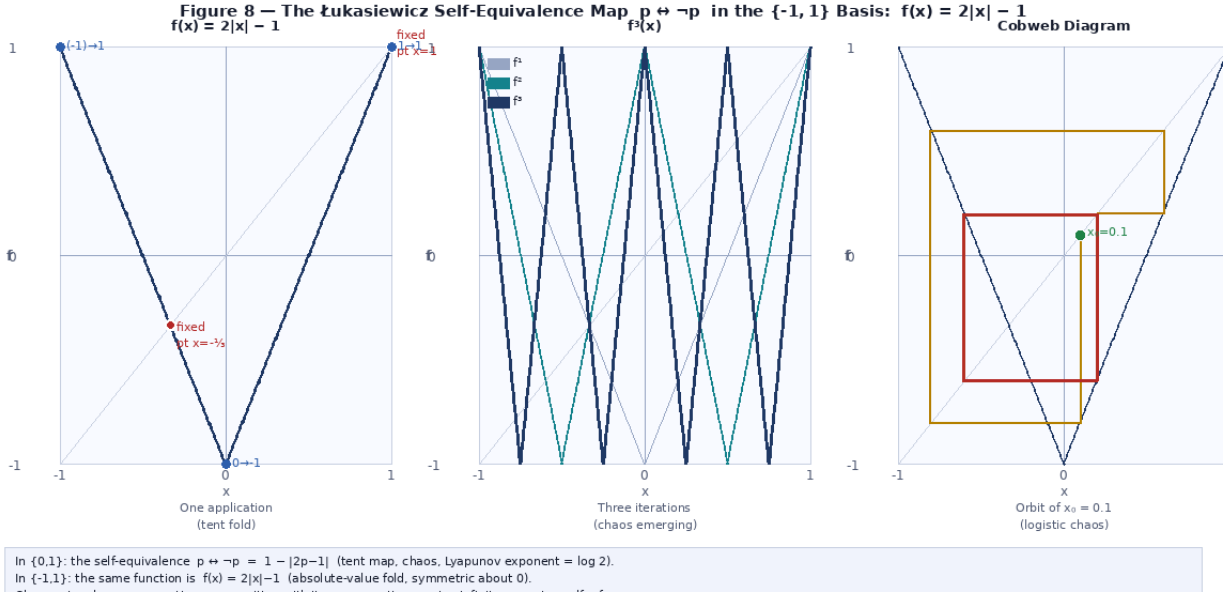

Figure 8. The Łukasiewicz self-equivalence map $p \leftrightarrow \neg p$ in {-1,1}: $f(x) = 2|x| - 1$. Left: single application — an absolute-value fold with fixed points at x=1 (False) and $x=-1/3$ (both unstable). Centre: three iterations showing the rapid proliferation of oscillations characteristic of chaotic dynamics. Right: cobweb diagram of the orbit starting at $x_0=0.1$, showing the chaotic trajectory that never converges. Lyapunov exponent $\lambda = \log 2$.

11.3 The IFS Connection: Folding as Fractal Generation

The map $f(x) = 2|x| - 1$ is a **folding map** — it collapses both $[-1, 0]$ and $[0, 1]$ onto $[-1, 1]$ by reflecting the left half over the origin. This operation is structurally identical to the first step of the IFS recursive tile subdivision: the quadtree splits a region into four sub-regions by folding along two axes. The chaos in $p \leftrightarrow \neg p$ is the continuous one-dimensional analogue of the fractal generation the paper describes discretely in the tile system. Both arise from the same mechanism: recursive folding of a bounded domain.

This extends the Sierpiński connection (Section 5) from a single discrete operator (XOR) to the entire continuous Łukasiewicz system. The XOR tile generates the Sierpiński triangle by recursive two-dimensional folding; the Łukasiewicz self-equivalence generates the tent map by recursive one-dimensional folding. Both are fractal attractors of a Boolean folding rule, one discrete and spatial, one continuous and temporal.

11.4 Spectral Continuity: Walsh Analysis Extended

The Walsh-Hadamard characters $\chi_S(x) = \prod_{i \in S} x_i$ are defined on $\{-1, 1\}^n$. Their restriction to the Boolean vertices gives the Fourier expansion of Section 6. Their *analytic continuation* to $[-1, 1]^n$ provides a natural spectral decomposition of Łukasiewicz's continuous truth domain using the same basis functions. The {-1, 1} Fourier machinery does not stop at the hypercube corners — it extends smoothly into the fuzzy interior, with the Łukasiewicz operators providing the continuous connectives that govern how truth values combine within that interior.

Concretely: the Łukasiewicz strong disjunction $\max(-1, x+y-1)$ is a truncated bilinear function. Its Fourier decomposition over $[-1, 1]^2$ has a dominant linear term (degree-1 characters χ_{1}, χ_{2}) and a small bilinear correction (degree-2 character χ_{12}), exactly where the Boolean AND polynomial $(1+x+y-xy)/2$ concentrates its weight. The two logics share the same spectral architecture at low degree; they diverge at high degree where the clamping nonlinearity introduces higher harmonics.

11.5 The Three-Domain Architecture

The {-1, 1} basis now sits at the centre of a precise three-domain hierarchy:

Figure 10 — Three-Domain Architecture of {-1,1} Logic

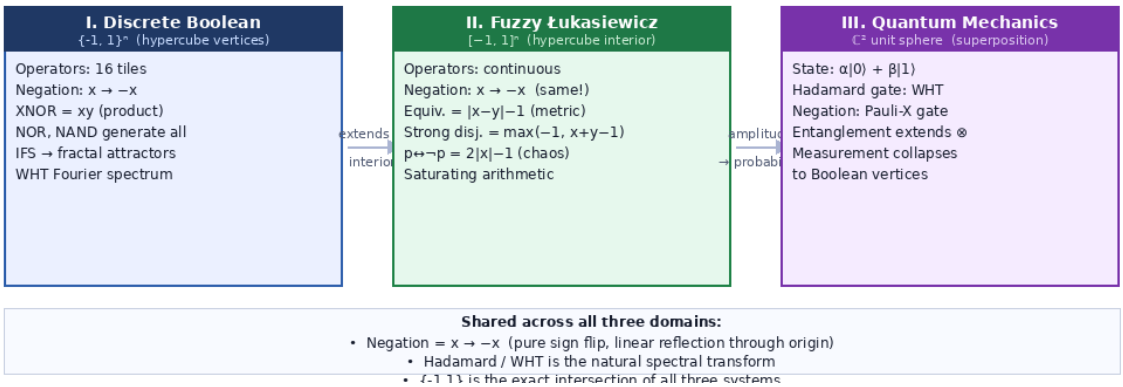

Figure 10. Three-Domain Architecture. Discrete Boolean logic ($\{-1,1\}^n$, hypercube vertices) is extended to Fuzzy Łukasiewicz logic ($[-1,1]^n$, hypercube interior) and further to Quantum mechanics ($\mathbb{C}^2$ unit sphere, superposition). Each domain extends the previous. Negation $x \to -x$ and the Hadamard/WHT transform are shared across all three.

The critical shared structure is negation: in all three domains, negation is $\mathbf{x \mapsto -x}$ — a linear reflection through the origin. In {0, 1} Boolean logic, negation is $x \mapsto 1 - x$, an affine map. Moving to {-1, 1} converts this from an affine to a purely linear operation, making explicit a symmetry that was always present but geometrically hidden. Łukasiewicz logic inherits this clean negation and extends it throughout the continuous interior. Quantum mechanics extends it further to the complex unit sphere, where the Pauli-X gate (NOT) is the quantum analogue and the Hadamard gate is the WHT applied to quantum amplitudes.

Each domain is a strictly larger container than the previous: Boolean vertices $\subset$ fuzzy interior $\subset$ quantum superposition. Measurement in quantum mechanics collapses the quantum state back to a Boolean vertex. The Łukasiewicz framework interpolates between these extremes — it is the continuous bridge between discrete computation and quantum probability.

12. Quantification in the Tile Framework

The tile framework as developed so far covers the full propositional layer: 16 connectives, functional completeness via NOR, Fourier spectral analysis, and Łukasiewicz fuzzy extension. A natural question is whether first-order and higher-order quantification — the logical machinery of "for all" and "there exists" — fit naturally into this geometric picture. They do, and the fit is exact.

19.1 First-Order Quantification as Axis Collapse

In a two-variable Boolean function f(A, B), quantifying over A means collapsing the A-dimension of the 2×2 tile using a pooling rule:

- $\forall A\ f(A,B) = f(-1, B)\ \text{AND}\ f(+1, B)$ — AND-pool across the A-rows
- $\exists A\ f(A,B) = f(-1, B)\ \text{OR}\ f(+1, B)$ — OR-pool across the A-rows

The 2×2 tile becomes a 1×2 strip. The aggregation function determines which cells survive the collapse. This is not an analogy: it is exactly what the downsampling pipeline does with a specific pooling function. **Quantification is downsampling. ∀ is AND-pooling. ∃ is OR-pooling.**

Each application of a quantifier reduces the Bit Budget B by exactly 1 — the same as descending one quadtree level. Quantification is **logical zoom**: it reduces the resolution of the truth surface by collapsing one variable dimension, exactly as geometric downsampling reduces spatial resolution by collapsing one spatial dimension.

Figure 14 shows this concretely for XOR (Index 6). ∀A applied to XOR yields FALSE (AND-pooling the two rows produces all-void) — XOR never holds for all A simultaneously. ∃A applied to XOR yields TRUE (OR-pooling yields all-live) — for every value of B there exists some A making XOR hold. The figure also demonstrates that ∀A∃B ≠ ∃B∀A on the NOR depth-2 surface, confirming that quantifier order corresponds to the non-commutativity of sequential downsampling operations.

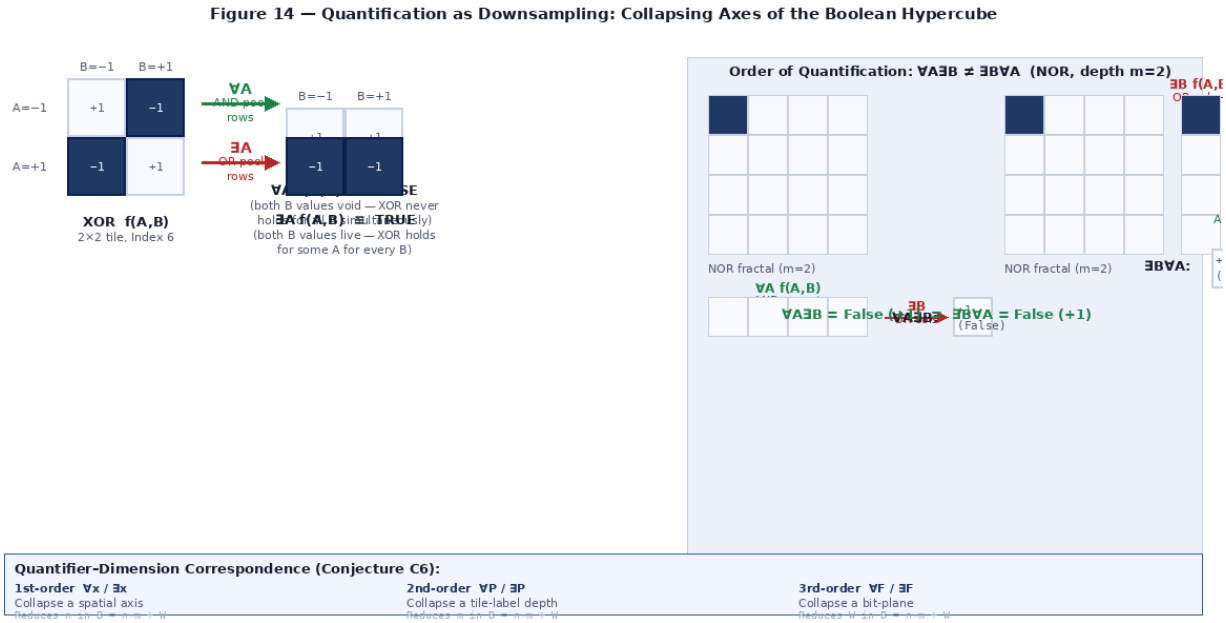

Figure 14. Quantification as downsampling. Top: XOR tile showing ∀A (AND-pool rows → FALSE) and ∃A (OR-pool rows → TRUE). Right: the NOR depth-2 surface demonstrating ∀A∃B ≠ ∃B∀A — quantifier order is the geometric

non-commutativity of axis-collapse operations. Bottom: the three-order quantifier–dimension correspondence (Conjecture C6).

In the {-1, 1} basis with True = −1, the sign convention means AND corresponds to **max** (takes the less True = more positive value) and OR corresponds to **min** (takes the more True = more negative value). So $\forall A\, f(A,B) = \max_{A\in\{-1,+1\}} f(A,B)$ and $\exists A\, f(A,B) = \min_{A\in\{-1,+1\}} f(A,B)$. This connects immediately to the Łukasiewicz section (Section 11): in the continuous fuzzy extension, $\forall x\, P(x) = \sup_{x\in[-1,1]} P(x)$ and $\exists x\, P(x) = \inf_{x\in[-1,1]} P(x)$, giving quantification over the full fuzzy interior as a natural generalisation of Boolean axis-collapse.

19.2 The Spectral Reading of Quantification

The Fourier coefficient $\hat{f}(\emptyset) = E[f(x)] = (1/2^n)\Sigma_x f(x)$ is the average truth value across all inputs — the DC component of the spectral expansion. In {-1, 1}:

- $\forall x\, f(x)$ holds iff $\hat{f}(\emptyset) = -1$: all energy at DC, uniformly True
- $\exists x\, f(x)$ holds iff $\hat{f}(\emptyset) < +1$: not everything False — some negative component exists
- $\neg\exists x\, f(x)$ holds iff $\hat{f}(\emptyset) = +1$: constant False — the Contradiction tile, Index 0

Applying quantifier $\forall x$ to eliminate a variable $x \in S$ from the Fourier expansion zeroes out every coefficient $\hat{f}(S)$ where $x \in S$. Variable elimination in logic is coefficient elimination in the spectrum. The KKL theorem — that every balanced Boolean function has at least one variable with influence $\Omega(\log n/n)$ — is a statement about $\exists$: there always exists a variable whose quantification would meaningfully alter the spectral structure.

19.3 Second-Order Quantification: Quantifying Over Tiles

Second-order logic quantifies over *predicates* — not individual values $x \in \{-1,1\}$ but functions $f : \{-1,1\}^n \to \{-1,1\}$. In the tile framework, this is extraordinarily natural: the 16 tiles **are** the functions. Second-order quantification ranges over which tile populates a node of the quadtree.

- $\forall P\, \varphi(P)$ = AND over all 16 tiles of φ applied to each tile — collapses one quadtree depth level
- $\exists P\, \varphi(P)$ = OR over all 16 tiles of φ applied to each tile — OR across the tile alphabet

First-order quantification collapses a *variable axis* of the grid, reducing n. Second-order quantification collapses a *tile-label axis* of the quadtree node labelling, reducing m. The quadtree is already a tree of tile labels; second-order quantification is quantification over the elements of the composition monoid from Conjecture C4.

Spectrally, second-order $\forall P$ averages over the full 16-tile alphabet. By the symmetry of the tile set (the 16 operators form a balanced set under all Boolean automorphisms), this averaging zeroes out all asymmetric Fourier components and retains only the symmetric, degree-0 component — the DC term. Functions with high nonlinearity resist first-order linear approximation; the question of what geometric structures such functions generate under RBHM encoding — and whether their Hausdorff dimensions cluster at predictable values — remains open and connects to Conjecture C2.

19.4 Third-Order Quantification and Fractal Inexhaustibility

Third-order quantification ranges over *sets of predicates* — functionals F that take Boolean functions to truth values. In the RBHM framework, this corresponds to quantifying over which

sequence of tiles (f_1, f_2, …, f_m) labels the depth levels of the level-varying quadtree. The 16^m distinct level-varying surfaces are the domain of third-order quantification over depth-m quadtrees.

Here the exact counting identity from Section 8.2 acquires its natural interpretation. The gap between $16^m = 2^{(4m)}$ (the structured subspace) and $2^{(4^m)}$ (the full function space) is not a logical limitation — it is the **fractal inexhaustibility of Boolean function space**. At any finite depth m, the level-varying encoding captures a structured slice of a space that extends infinitely deeper. The surfaces beyond that slice are not permanently inaccessible: the NOR pipeline reaches them all. They simply require going deeper into the fractal. As m increases, the fraction of level-varying expressible surfaces to total surfaces vanishes — just as any finite rendering of a fractal reveals a smaller and smaller fraction of the infinite limit set.

This is the mathematical character of the framework: it is not incomplete in any logical sense, it is **inexhaustible**. The Universal NOR Hypercube at infinite depth contains every possible truth surface. Any finite depth m is a cross-section — always revealing structure, always leaving more structure deeper. This connects directly to St. Denis and Grim (1997): fractal images of formal systems are always finite approximations of infinite limit sets. The "incompleteness" is the incompleteness of any finite rendering of an infinite fractal, which is precisely the right way to understand why the level-varying encoding cannot cover everything at a given depth. At the next depth, it covers more — and there is always a next depth.

Third-order quantification collapses the W (bit-depth) axis of the Bit Budget. Parseval's Identity $\Sigma_S \hat{f}(S)^2 = 1$ is the statement that **full third-order universal quantification normalises spectral energy to 1**: summing over all spectral subsets S completes the energy conservation of the system. It is the spectral completeness theorem, and it holds exactly because the {-1, 1} basis provides the symmetric inner product that makes orthogonality possible.

Figure 15 — Higher-Order Quantification, the Bit Budget, and Fractal Inexhaustibility

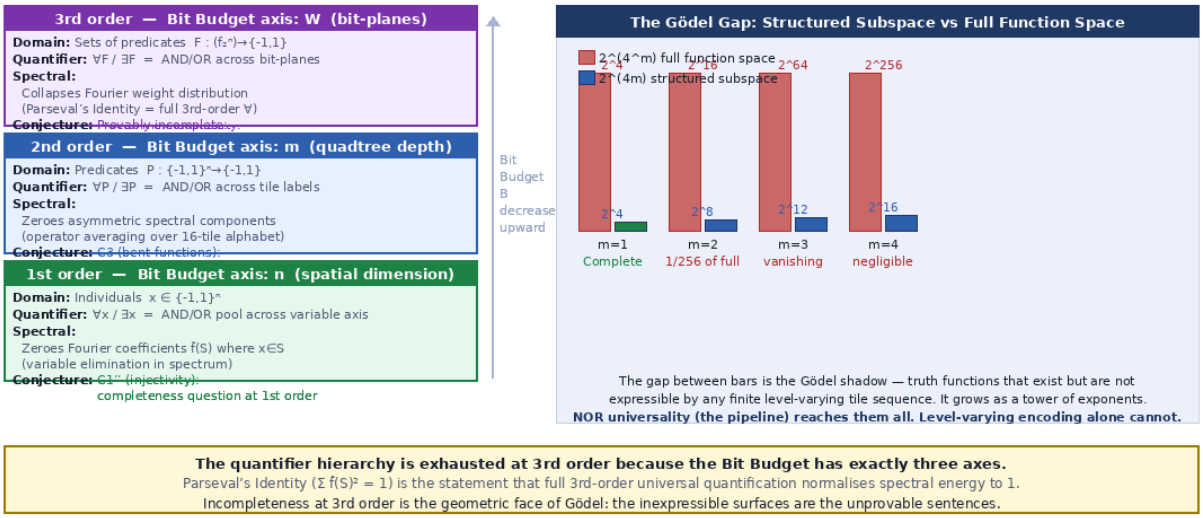

Figure 15. Higher-order quantification, the Bit Budget axes, and fractal inexhaustibility. Left: the three-layer hierarchy mapping quantifier order to Bit Budget axis. Right: the structured subspace (2^(4m), blue) vs the full function space (2^(4^m), teal) at each depth m. The shrinking ratio is not a logical gap — it is fractal inexhaustibility: at any finite depth, the level-varying encoding sees a slice of an infinite structure. The Universal NOR Hypercube at infinite depth contains everything. Fractals are never finished.

19.5 The Natural Boundary at Third Order

The quantifier hierarchy is **exhausted at third order** within the Universal NOR Hypercube framework, because the Bit Budget has exactly three axes. There is no natural fourth axis in the (n, m, W) parameterisation, and therefore no natural fourth-order quantification. This is a structural feature of the framework, not a limitation: it reflects the fact that the Universal NOR Hypercube is fully parameterised by three degrees of freedom.

The three orders of quantification also correspond to the three topological reasons why 3D is the natural folding target (Section 9.3): first-order quantification collapses space (dimension), second-order collapses scale (depth), third-order collapses value (bit-depth). The dimensional folding isomorphism and the quantifier hierarchy are two perspectives on the same three-parameter structure.

PART II — THE GEOMETRY OF INFORMATION

Bit budget • Downsampling pipeline • n-Dimensional tile alphabets • Universal NOR hypercube

13. The Bit Budget: Three Axes of One Parameter

The IFS integer encoding ($f_0(p) = 2p$, $f_1(p) = 2p+1$) makes a structural equivalence explicit. Each iteration appends exactly one new bit to the cell's path address. One more iteration is simultaneously one more bit of spatial resolution, one more bit of value depth, one more quadtree level, and one more Boolean variable. Spatial dimension n, resolution depth m, and value bit-depth W are three names for the same zoom operation applied along different axes of the same recursive structure.

Definition 16.1 (The Bit Budget). The total bit budget B of an (n, m, W) surface is $B = n \cdot m + W$. The allocation of B among n (dimension), m (depth), and W (bit-planes) is a choice of representation, not a property of the object. A depth-3 2D binary surface (B = 6, allocation 2×3+0) is the same object as a depth-2 2D surface with 2-bit values (B = 2×2+2 = 6): same budget, different allocation.

The term is borrowed deliberately from engineering, where a bit budget governs how a fixed payload is allocated across resolution, quality, and temporal axes — spending more on one means less for the others. The mathematical concept here is identical: a fixed total B distributes across spatial dimension n, recursion depth m, and value range W, and allocating more to any one axis necessarily reduces what remains. $B = n \cdot m + W$ as a unified topological parameter is an original construct of this framework, but the intuition it names is the same engineering intuition applied to logical space rather than data transmission.

Incrementing B by exactly one unit corresponds simultaneously to: computing one more hierarchical quadtree level; moving into one higher dimension of spatial detail; or introducing one more distinct variable into the Boolean logic space. High-resolution complexity structurally and geometrically subsumes all lower-resolution iterations. This is the Bit Budget as a unifying parameter, and it is the geometric reason why quantification (Section 19) reduces B by exactly 1 per quantifier application.

14. The Downsampling Pipeline: From NOR Fractals to Any Surface

The constructive proof of universality is the downsampling pipeline. Given target parameters (n, m, W), the pipeline produces any n-dimensional hypercubic surface from W independent NOR fractals in three steps.

Definition 17.1 (The Downsampling Pipeline). Given n (spatial dimension), m (resolution depth per axis), W (value bit-depth):

- Step 1. Generate W independent n-dimensional NOR fractals at source depth $M \geq m$, producing W binary volumes of size $2^T \times \cdots \times 2^T$ (n axes).
- Step 2. Downsample each volume by factor 2^(M−m) along every axis: b_k(i_1,…,i_n) = V_k(i_1 · 2^(M−m),…,i_n · 2^(M−m)).
- Step 3. Combine W binary volumes by bit-plane concatenation: value(i_1,…,i_n) = b_{W−1} · 2^(W−1) + ⋯ + b_1 · 2 + b_0. Result: an n-dimensional grid of side 2^m per axis with values in {0,…,2^W−1}.

Theorem 17.1 (NOR Universality). For any $n \geq 1$, $W \geq 1$, $m \geq 1$, and any n-dimensional hypercubic surface S of side 2^m with values in {0,…,2^W−1}, there exist W independent n-dimensional NOR fractals at source depth $M \geq m$ whose pipeline output exactly reproduces S.

Proof sketch: any Boolean function of any arity is expressible as a finite NOR circuit (functional completeness of NOR). The n-dimensional quadtree depth corresponds directly to NOR circuit depth. W independent planes address W independent bits. □

13.1 The 3D Concrete Example (n=3, m=1, W=2)

The smallest non-trivial volumetric case: a 2×2×2 cube with 4-valued cells (values in {0,1,2,3}). Parameters: n = 3, m = 1, W = 2, bit budget B = 3 · 1 + 2 = 5.

- Step 1: Generate two independent 3D NOR fractals at source depth M = 4, producing 16×16×16 binary volumes (b_1 = high bit-plane, b_0 = low bit-plane).
- Step 2: Downsample each volume to 2×2×2 by factor 8 along all three axes.
- Step 3: Combine as value = 2 · b_1 + b_0 to produce the 4-valued 2×2×2 cube.

The 8 cells each hold a value in {0,1,2,3}, giving 4^8 = 65,536 possible target cubes. The structured subspace directly addressable by one (operator, operator) pair for the two bit-planes has 16 × 16 = 256 members — injective on non-degenerate pairs by Conjecture C1''.

15. Dimensional Constraint and n-Dimensional Tile Alphabets

In n-dimensional space each IFS iteration consumes exactly n bits — one per spatial axis. Valid total spatial bit depths must therefore be multiples of n. Non-multiples produce non-hypercubic grids that break tile symmetry. This resolves an apparent awkwardness: 3 bits is not a valid spatial depth in 2D (iterations consume 2 bits each) but is the exact atomic level in 3D (one octtree iteration produces a 2×2×2 cube with 2^8 = 256 possible tile functions).

n	Atomic tile	Tile alphabet 2^(2^n)	Tree arity	Bits/level
1	2-cell strip	`2^(2^1) = 4`	2-ary	1

n	Atomic tile	Tile alphabet 2^(2^n)	Tree arity	Bits/level
2	2×2 square	2^(2^2) = 16	4-ary quadtree	2
3	**2×2×2 cube**	**2^(2^3) = 256 ★**	8-ary octtree	3
4	2^4 tesseract	2^(2^4) = 65,536	16-ary	4
n	2^n hypercube	2^(2^n)	2^n-ary	n

★ n=3 uniquely matches the byte (256 tile functions = 2^8), the natural atom of digital memory indexing. This is the algebraic reason 3D is the natural reduction target of the dimensional folding isomorphism.

16. The Universal NOR Hypercube

Definition 19.1 (The Universal NOR Hypercube). The universal NOR hypercube is the infinite-dimensional, infinite-resolution binary structure whose value at any vertex is determined by recursive NOR application across all coordinates. Every finite surface in the framework is a finite projection P(n, m, W) of this single object: n selects which spatial axes to expose, m selects the resolution depth, and W selects how many bit-depth axes to include.

One-line statement: all n-dimensional hypercubic surfaces at all resolutions and all bit-depths are cross-sections of a single infinite-dimensional recursive NOR structure. The sixteen 2D connectives, the 256 3D connectives, the Sierpiński gasket, and any W-bit volumetric surface are the same object — described by which axes are exposed and how many recursion levels are unfolded.

Moving up in spatial dimension is zooming out; moving down is zooming in. An (n+1)-dimensional NOR fractal contains the n-dimensional one as its corner face. These are not different objects — they are the same attractor at different scales and axis exposures. The fractal inexhaustibility of Section 8.2 is the statement that no finite cross-section captures the whole: the object is always there, but any finite view is always partial.

17. Topological Preservation Under Downsampling

When a logical space is simplified — variables abstracted, resolution reduced — the structural integrity of truth regions must be preserved. Naive reductions such as max-pooling and average-pooling arbitrarily destroy topology, altering the Betti numbers of the live-cell configurations. Optimisation-based downsampling that respects discrete Boolean constraints preserves topological invariants but is computationally expensive.

The RBHM quadtree is itself the natural downsampling hierarchy. Moving from depth m to depth m−1 is simply reading one level higher in the same tree — the aggregation rule for the parent cell is determined by the tile label at that node. No external curve or pooling kernel is required. Downsampling in this framework is **quadtree level selection**: the fractal is always there at every depth, and coarser resolution is a shallower read of the same recursive structure. This is the topological preservation claim: the quadtree hierarchy, because it is generated by the same tile rules at every level, preserves the logical structure of the surface at every scale by construction.

Locality within a quadrant is preserved natively. Cells within the same quadtree sub-quadrant at any depth are physically adjacent and are addressed by contiguous quadtree path codes. Locality across quadrant boundaries is the open question addressed by Conjecture C5.

18. Why 3D is the Natural Reduction Target

The dimensional folding isomorphism establishes that any n-dimensional Boolean surface is isomorphic to a 3D surface. But why 3D specifically? Three independent reasons converge.

Figure 12 — Why 3D? The Privileged Status of n=3 in the Dimensional Hierarchy

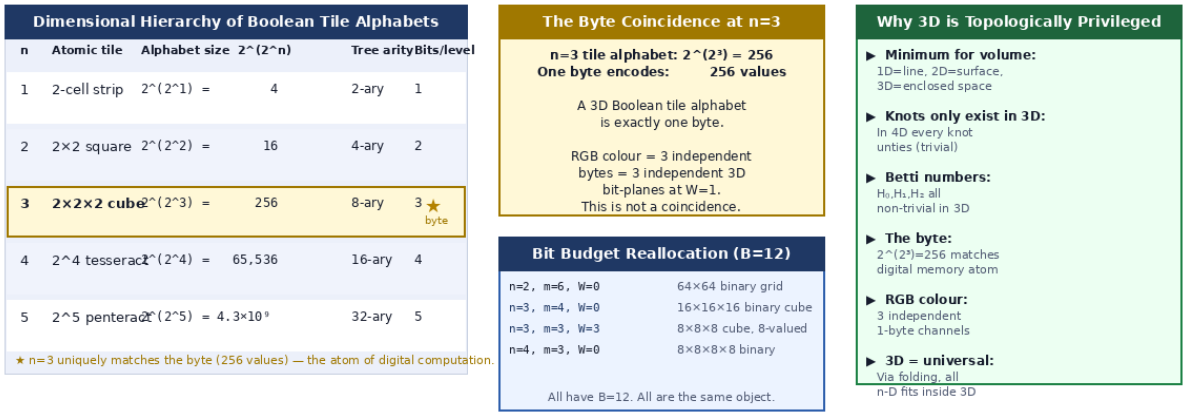

Figure 12. The dimensional hierarchy of Boolean tile alphabets. n=3 uniquely matches the byte ($256 = 2^8$ tile functions), the atom of digital memory. The bit-budget reallocation panel shows that any B=12 surface — regardless of how n, m, W are allocated — is the same object in a different coordinate system.

- Topological privilege. 3D is the minimum dimension with enclosed volume. In 1D there are only points and intervals; in 2D, surfaces but no interior; in 3D, enclosed space first appears. Knot theory is non-trivial in 3D (knots cannot be untied) but trivial in 4D (every knot untangles). The Betti numbers H_0, H_1, H_2 are all simultaneously non-trivial only in 3D.
- The byte coincidence. The 3D tile alphabet has size $2^{(2^3)} = 256 = 2^8$ — exactly one byte. Digital memory is byte-addressed. The 3D Boolean tile alphabet is the atom of digital computation, matching the standard memory model with no remainder. No other dimension achieves this: n=2 gives 16 (too small for a byte), n=4 gives 65,536 (too large).
- The RGB / colour connection. Digital colour is represented as three independent one-byte channels (R, G, B). This is precisely three independent 3D bit-planes at $W = 1$ — exactly the $n = 3$, $W = 1$ allocation of the tile framework. The folding from n-D to 3D is the abstract generalisation of how any volumetric dataset is stored as a colour volume.

19. Open Conjectures

The following conjectures remain open and are stated as research hypotheses, not established results.

Conjecture C1'': Injectivity on Non-Degenerate Tuples

The level-varying map from m-tuples of non-degenerate connectives to 2m-variable Boolean functions is injective: distinct tuples produce distinct functions. Non-degenerate means no component is a constant operator (FALSE or TRUE). Provided operators do not collapse to constant degeneracies, specific bit-pairs at precise geometric addresses guarantee unique logical inputs render unique geometric boundaries, preventing spatial collapse under deep RBHM recursion. In n dimensions: injectivity of m-tuples of non-degenerate 2^n-variable connectives to nm-variable Boolean functions.

Conjecture C2: Measure-Theoretic IFS Convergence

For any non-degenerate connective f, the uniform measure over live cells of a depth-d RBHM surface converges weakly, as $d \to \infty$, to the invariant measure of the corresponding IFS with contraction ratio 1/2. The Hausdorff dimension of the support equals the fractal dimension of the RBHM attractor. This would prove that algorithmic Boolean recursion limits are isomorphic to continuous integration.

The scope of this conjecture is clarified by the XOR/Sierpiński case, where the Hausdorff *dimension* $s = \log(3)/\log(2) \approx 1.585$ is exactly known. However, the Hausdorff *measure* H^s(Λ) — which quantifies the actual geometric "size" of the Sierpiński attractor in its fractional dimension — remains an open problem in geometric measure theory, with established bounds $0.77 \leq H^s(\Lambda) \leq 0.818$. Conjecture C2 targets the weaker statement of convergence of the discrete RBHM measures to the IFS invariant measure, not the exact computation of the measure itself, which remains open even for the simplest attractors.

Conjecture C4: Algebraic Closure Under Composition

The sixteen connectives are closed under texture composition. Composing two uniform-label quadtrees yields a surface equivalent — up to bit-permutation of the path address — to some uniform-label quadtree. The composition table defines a monoid with identity element (the Identity-A tile at Index 3). Proof obligation: verify all 256 composition pairs at depth $m = 2$; prove inductive closure; characterise identity and idempotents. For $n = 3$: verify all $256^2 = 65{,}536$ pairs over the 256-operator alphabet.

Conjecture C5: Locality Across Quadrant Boundaries

The native quadtree path code preserves locality within quadrants but may place adjacent cells far apart across boundaries. Whether an alternative address remapping — such as the Hilbert space-filling curve — provides a locality guarantee bounded independently of n, that is, whether the maximum Hamming distance between adjacent cells' images depends only on their distance and not on the source dimension, is an open question.

Conjecture C6: The Quantifier–Dimension Correspondence

Within the specific (n, m, W) parameterisation of the Universal NOR Hypercube, the three natural orders of quantification correspond bijectively to the three axes of the Bit Budget $B = n \cdot m + W$: first-order quantification (over individuals $x \in \{-1,1\}^n$) reduces n; second-order (over predicates P) reduces m; third-order (over functionals F) reduces W. This is a structural property of the three-parameter (n, m, W) representation — it does not make a claim about the limits of higher-order logic in general. As a corollary within this framework: there is no natural fourth-order quantification, because the Bit Budget has exactly three axes. Parseval's Identity — $\Sigma\_S \hat{f}(S)^2 = 1$ — is the statement that full third-order universal quantification normalises spectral energy to 1.

20. A Note on Connections

The spectral structure of propositional logic established in this paper surfaces in several other domains. Most directly in logic: Kalai (2002) showed that Arrow's Impossibility Theorem admits a Fourier-analytic treatment in which each voter's influence is a coordinate in $\{-1,1\}^n$, and any noise-stable social choice function is provably close to a dictator. This is a quantitative Fourier-analytic strengthening of Arrow — it makes the structure of the impossibility geometrically precise — rather than a direct application of the FKN theorem, which requires a balance condition that social choice functions do not always satisfy. The shared ingredient is the Walsh-Hadamard spectral decomposition: the same framework that makes the 16-tile attractors visible also makes Arrow's result computable.

Beyond logic: Hadamard conjugation — the direct analogue of the Walsh-Hadamard Transform over quaternary alphabets — is used in phylogenetics to decompose fitness landscapes into additive and epistatic components. The {-1, 1} symmetry that makes Boolean Fourier analysis clean extends naturally to these settings. These connections are noted as evidence that the spectral structure of the tile basis is not particular to propositional logic but reflects something deeper about the mathematics of binary and near-binary structures.

21. Conclusion

The Boolean Tile Basis begins with two propositional variables P and Q. They generate exactly 16 distinct truth functions — the 16 tiles. These tiles form a self-contained alphabet: P and Q are themselves tiles (Indices 3 and 5), not external primitives. NOR is the sole generator from which all 15 others are constructible. This is the flat ontology: operators, variables, and constants inhabit one categorical level.

Each tile generates a unique IFS fractal attractor under recursive self-substitution. The fractal images are not illustrations of logic — they *are* the logic, rendered spatially. XOR generates the Sierpiński triangle at Hausdorff dimension $\log(3)/\log(2) \approx 1.585$, a dimension determined by XOR's Fourier structure: one dominant coefficient maps to one dominant void pattern. The {-1, 1} basis makes this connection algebraically exact.

Extending to many-valued logic, the Łukasiewicz operators translate into {-1, 1} in unusually clean form. Negation becomes pure reflection $x \mapsto -x$. Strong disjunction and conjunction become symmetric clamped sums. And the self-equivalence paradox $p \leftrightarrow \neg p$ becomes the map $f(x) = 2|x| - 1$: a provably chaotic system with Lyapunov exponent $\log 2$ and no stable truth value. Logical self-reference generates deterministic chaos. This is not a metaphor; it is a theorem.

Logical quantification maps exactly onto the geometry. $\forall x$ is AND-pooling along the variable axis. $\exists x$ is OR-pooling. Each quantifier reduces the Bit Budget $B = n \cdot m + W$ by 1, collapsing one degree of freedom of the logical space. The three orders of quantification — first (over individuals), second (over predicates), third (over predicate sequences) — correspond to the three axes n, m, W. The hierarchy is exhausted at third order because logical space has exactly three degrees of freedom. Parseval's Identity, $\Sigma\_S \hat{f}(S)^2 = 1$, is the spectral form of full third-order universal quantification.

After any sequence of quantifier applications, the residual logical space is 3-dimensional — the natural fixed point of the quantification process, where the tile alphabet reaches 256 elements,

matching the byte. The logical space is inexhaustible: at any finite depth, the level-varying encoding sees a structured slice of a space extending infinitely deeper. The NOR fractal at infinite depth contains everything. There is always more at the next level of resolution.

The 1997 paper established that formal systems have fractal images. This paper establishes *why* they must: because a 2×2 tile is a quadtree node, a quadtree generates Morton path codes, and a Morton-addressed quadtree under a Boolean rule is an IFS. The fractal images are not found — they are derived. The Sierpiński triangle is the carry-free locus of XOR, proven by Lucas's theorem. The tent map is the self-referential fixed point of Łukasiewicz logic. The quantifier hierarchy is the axis-collapse structure of the quadtree. The space of truth functions is inexhaustible in exactly the sense that the fractal is inexhaustible: always more detail at the next depth, always the same structure at every scale. And the unique privileged dimension is 3, where the XOR fractal becomes a complete surface, the tile alphabet matches the byte, and the Sierpiński voids are algebraically closed by the XNOR third coordinate.

References

1. St. Denis, P. and Grim, P. "Fractal Images of Formal Systems." Journal of Philosophical Logic, 26(2): 181–222, 1997. DOI: 10.1023/A:1004280900954.
2. Grim, P., Mar, G. and St. Denis, P. The Philosophical Computer: Exploratory Essays in Philosophical Computer Modeling. MIT Press, 1998.
3. St. Denis, P. The Boolean Tile Basis for Binary Surfaces: A Quadtree Unification of Logic, Geometry, and Dimension. Preprint, 2025.
4. O'Donnell, R. Analysis of Boolean Functions. Cambridge University Press, 2014.
5. O'Donnell, R. Some Topics in Analysis of Boolean Functions. CMU. https://www.cs.cmu.edu/~odonnell/papers/analysis-survey.pdf
6. Filmus, Y. Analysis of Boolean Functions — Lecture Notes. Technion, 2015. https://yuvalfilmus.cs.technion.ac.il/Courses/BooleanFunctionAnalysis/2015/LectureNotes.pdf
7. Kahn, J., Kalai, G. and Linial, N. The influence of variables on Boolean functions. Proc. 29th FOCS, pp. 68–80, 1988.
8. Kalai, G. A Fourier-theoretic perspective on Arrow's theorem. Advances in Applied Mathematics, 29(1): 232–246, 2002.
9. Spiro, S. Fourier Analysis of Boolean Functions. University of Chicago REU, 2016. https://math.uchicago.edu/~may/REU2016/REUPapers/Spiro.pdf
10. Hutchinson, J. E. Fractals and self-similarity. Indiana University Mathematics Journal, 30(5): 713–747, 1981.
11. Barnsley, M. F. Fractals Everywhere. Academic Press, 1988.
12. Falconer, K. Fractal Geometry: Mathematical Foundations and Applications. Wiley, 1990. (Standard reference for Hausdorff dimension and IFS attractor theory.)
13. Wong, J. Space-filling curves and Hausdorff dimensions. NSYSU Mathematics. https://www-math.nsysu.edu.tw/~wong/papers/sfc_f.pdf
14. Morton, G. M. A computer oriented geodetic data base. IBM Technical Report, 1966. (The literature name for the quadtree path address structure — native to the RBHM, independently identified by Morton for spatial databases.)

15. Samet, H. The Design and Analysis of Spatial Data Structures. Addison-Wesley, 1990. (Quadtrees and spatial indexing.)
16. Post, E. L. The two-valued iterative systems of mathematical logic. Princeton University Press, 1941. (Functional completeness of NOR/NAND.)
17. Hájek, P. Metamathematics of Fuzzy Logic. Kluwer Academic, 1998. (Łukasiewicz infinite-valued logic.)
18. Friedgut, E., Kalai, G. and Naor, A. Boolean functions whose Fourier transform is concentrated on the first two levels. Advances in Applied Mathematics, 29(3): 427–437, 2002. DOI: 10.1016/S0196-8858(02)00024-6. (The FKN theorem; the balance condition is required for the dictator conclusion.)
19. Cipra, B. A. An introduction to the Ising model. The American Mathematical Monthly, 94(10): 937–959, 1987. DOI: 10.1080/00029890.1987.12000742.
20. Lucas, A. Ising formulations of many NP problems. Frontiers in Physics, 2: 5, 2014. DOI: 10.3389/fphy.2014.00005. (NP-hardness of Ising ground state and QUBO equivalence for Boolean satisfiability.)
21. Conway, J. H. Mathematical Games: The fantastic combinations of John Conway's new solitaire game 'Life'. Scientific American, 223: 120–123, 1970.
22. Rothaus, O. S. On bent functions. Journal of Combinatorial Theory A, 20(3): 300–305, 1976. (Background on nonlinearity; bent functions noted but not the subject of a conjecture in this paper.)
23. IFS Mega Heatmap Survey (interactive visualisation, P. St. Denis). https://tltmedia.github.io/ThinkingMattersExercises/mathFun/ifs_mega_heatmap_survey.html