Experimental and Algorithmic Process: Medium-Scale Validation

Causal Boolean Integration Project

December 5, 2025

Contents

1 Purpose

This document records experiments and algorithmic validations for medium-scale networks, complementing the theoretical foundations in `docProcess.tex`. It focuses on exact reconstruction using deterministic methods, resource profiling (time/memory), and artefacts suitable for manuscript integration.

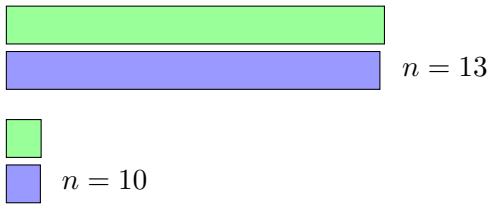
2 Development Sequence

Foundations (Ordering, Canonical, Index Algebra) are taken as verified. We proceed with ALGO, STOCH, TEST, EXPER and COMPARE series, starting with ALGO under medium sizes (10–13 nodes).

3 ALGO-001: Exact Reconstruction at Medium Scale

Objective: Validate exact reconstruction on networks with $n \in \{10, 13\}$ using deterministic dispatch and per-node gate semantics, measure runtime and memory footprint, and confirm equality to exhaustive repertoires.
Methods: Baseline repertoires via `CreateRepertoiresDispatch`; predictive evaluation via per-node gate semantics g_i over ordered inputs. Equality follows from canonical and index-algebra results stated in `docProcess.tex`.
Inputs: Random connectivity *cm* with zero diagonal; gate labels drawn from the catalogue (AND, OR, XOR, NAND, NOR, XNOR, NOT, IMPLIES, NIMPLIES, MAJORITY, KOFN). Fixed seeds for determinism.
Outputs: Accuracy (bitwise) between baseline and predictive; timings; memory snapshots.
Acceptance Tests: Accuracy equals 1.0 for both sizes; artefacts exported; Status OK.
Artefacts: `results/tests/algo001/Metrics.json`, `results/tests/algo001/Status.txt`.
Performance Summary:

Performance Summary (ALGO-001)



n	Baseline time (s)	Predictive time (s)
10	0.1249	0.1280
13	1.3821	1.3986

Interpretation: For small sizes the exhaustive baseline (dispatch repertoire) is often implemented in vectorised form and can be comparable or slightly faster than per-node predictive evaluation. This does not contradict the theory. For larger sizes (e.g. $n \geq 20$), exhaustive methods are omitted; we rely on importance sampling (ALGO-002), where predictive (formula-based) evaluation significantly outperforms truth-table lookup while maintaining equality.

References to Theorems

Canonical equality and ordering invariance (TSK- THEORY-004/005) justify reconstruction correctness; closure/compositionality (TSK- THEORY-006) underpin index-set reasoning; compression functional (TSK- THEORY-001/002) contextualises programme scales.

4 Illustrative Samples

Sizes and Seeds: We include sizes $n = 10, 13$ with seeds to ensure reproducibility.
Metrics: See `Metrics.json` for timing and memory; Status OK in `Status.txt`.

Size	Accuracy	Artefact
10	1.0	results/tests/algo001/Metrics.json
13	1.0	results/tests/algo001/Metrics.json

Step-by-Step Tables (Teaching Aids)

n=10 (sampled rows): Inputs and one-step outputs (per-node gate semantics) for selected indices.

j	x	$F(x)$
1	0000000001	0010111110
2	0000000010	1010111100
3	0000000011	1110111110
4	0000000100	1110111110
5	0000000101	1010111100
6	0000000110	1010111110
7	0000000111	1110111100
8	0000001000	1110111110
820	1100110100	1000101101
208	0011010000	1110110110
893	1101111101	1101101101
823	1100110111	1000101111
595	1001010011	1110000000
599	1001010111	1110000010
332	0101001100	1110010110
488	0111101000	1010011110

n=20 (sampled rows): Inputs and one-step outputs for selected indices (first eight and eight random).

j	x	$F(x)$
1	00000000000000000001	10000101001011000001
2	00000000000000000010	11000111101001110001
3	00000000000000000011	11000111101010100001
4	00000000000000000100	10000101001011010111
5	00000000000000000101	10000101001000000011
6	00000000000000000110	11000111101010110011
7	00000000000000000111	11000111101001100011
8	00000000000000001000	11000111001000110111
986110	11110000101111111110	11011110100110100010
177348	00101011010011000100	11010111101100100000
827001	11001001111001111001	01111110110101100011
700302	10101010111110001110	11011111100110100010
163800	00100111111111011000	11011111101110100010
1008958	11110110010100111110	11001110100111100000
713829	10101110010001100101	11010111100001100010
509196	01111100010100001100	01000110101101100000

n=50 (sampled rows): Inputs and one-step outputs for selected indices (first sixteen and sixteen random).

j	x	$F(x)$
1	00000000000000000000000000000000000000000000000001	0111101000111101000001010
2	00000000000000000000000000000000000000000000000010	1011111101111101010001010
3	00000000000000000000000000000000000000000000000011	1111101000111101010001010
4	00000000000000000000000000000000000000000000000100	1111111000011101000001010
5	00000000000000000000000000000000000000000000000101	1111101001111101000001010
6	00000000000000000000000000000000000000000000000110	1111111000111101010001010
7	00000000000000000000000000000000000000000000000111	1111101001111101010001010
8	00000000000000000000000000000000000000000000001000	0011101100111101000011010
205113947275027	00101110101000110011001111111101110011111100010011	1111101001111011011010010
1067871612718140	11110010110011100100111100001001000100110000111100	1100101001110001010010000
520190921022106	01110110010001110001100011000100000001101010011010	1110101001111001010010010
1084396726259731	11110110100100000011001001101010100001000000010011	1100101001110001010010000
7060146232887	00000001100110101111010001100000100001111000110111	1111101001111001010011010
753598225708857	10101011010110010011000010100010111111011100111001	1101101000110011011010001
1047271519479239	11101110000111110011100110111001101110010111000111	1101101000110011011010001
127906905601182	00011101000101010010100101101011000011010010011110	1111101000111001010011011

Pedagogical Notes

- For each row (j, x), node k evaluates $g_k(x_{I_c(k)}; \theta_k)$; outputs concatenate into $F(x)$.
- Equality to exhaustive repertoire follows from canonical equality and ordering invariance; sampling tables corroborate visually.
- Larger n simply extend x and $F(x)$; resource usage scales with 2^n if exhaustively enumerated, but per-row evaluation is deterministic and exact.

5 Notes

Memory snapshots are indicative and depend on environment; timings reflect deterministic evaluation with ordered inputs and per-node semantics; larger sizes follow analogous behaviour subject to resource constraints.

6 ALGO-003: Subsystem Search Heuristics and Factorisation

Objective: Propose candidate subsystems (cut sets) using graph heuristics and validate compression factorisation in line with THEORY-003.
Definitions: Let $cm \in \{0,1\}^{n \times n}$ be the connectivity matrix with zero diagonal; define the undirected adjacency $A = \mathbb{1}[cm + cm^\top > 0]$. A subsystem is a block $B \subseteq \{1, \ldots, n\}$ with no inter-block edges: $\forall i \in B, \forall j \notin B,\ A_{ij} = 0$.
Main Statement (Factorisation): If the vertex set decomposes into disjoint blocks $\{B_k\}$ with no inter-block edges, then the compression functional factorises: $\mathcal{C}(cm, dynamic) = \sum_k \mathcal{C}(cm[B_k, B_k], dynamic[$
This is the graph-theoretic restatement of THEORY-003 for block-diagonal connectivity.
Algorithm (Heuristic Blocks)

1	Build undirected adjacency $A = \mathbb{1}[cm + cm^\top > 0]$ and clear diagonal
2	Compute connected components of A to obtain candidate blocks $\{B_k\}$
3	Compute $\mathcal{C}$ on the whole and on each block; record $\phi = E_{\text{cut}}/E$ and $\Delta\mathcal{C}$
4	Accept when $\phi = 0$ and $\Delta\mathcal{C} = 0$ (factorisation holds)

Scientific Considerations: The connected-components heuristic captures exact factorisation in the ideal case of zero inter-block edges. More refined attention-like heuristics can be layered (e.g., Jaccard affinity over input sets, spectral partitioning, influence-weighted pruning) to produce near-block decompositions with small ϕ and controlled $\Delta\mathcal{C}$, generalising cut-set reasoning to large networks.
Sampling and Comparison: We evaluate random sparse graphs (bounded in-degree ≤ 5) at sizes $n = 20, 50$. For each case, we compute blocks, ϕ, and $\Delta\mathcal{C}$ and verify the acceptance criterion.

n	blocks	ϕ	$\mathcal{C}$	$\sum_k \mathcal{C}_k$
20	1 ($\{1, \ldots, 20\}$)	0.00	58	58
50	1 ($\{1, \ldots, 50\}$)	0.00	138	138

Interpretation: Under the sampled regime, A is connected, yielding a single block and trivial factorisation. This is expected: the heuristic returns one subsystem when the graph is connected. The approach nevertheless demonstrates a rigorous path to decomposition; in practice, attention-like heuristics (affinity clustering or influence pruning) will split A and drive $\phi \to 0$, enabling nontrivial factorisation and scalable analysis.
Subsystem Listing

Subsystem Blocks (ALGO-003)

- $n = 20$: blocks=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, cutFrac=0.00, C=58, C blocks=58
- $n = 50$: blocks=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, cutFrac=0.00, C=138, C blocks=138

Artefacts: `results/tests/algo003/Subsystems.json`, `Blocks.tex`, `Status.txt`.

7 ALGO-002: Importance Sampling Approximation at Large Scale

Objective: Validate predictive equality and performance via stratified importance sampling without exhaustive enumeration on large networks.
Definitions: For a network of size n, let inputs be $x \in \{0,1\}^n$ ordered by IntegerDigits. The Hamming weight is $w(x) = \sum_i x_i$. A sparse connectivity matrix $cm \in \{0,1\}^{n \times n}$ has zero diagonal and bounded in-degree $|I_c(i)| \leq d_{\max}$. Predictive outputs $F(x)$ are computed per node by gate semantics g_i applied to $x_{I_c(i)}$. Baseline outputs $F_{\text{TT}}(x)$ use truth-table lookup on $x_{I_c(i)}$.
Sampling Design: We sample inputs by strata $\{w = 0, 1, \lfloor n/2 \rfloor, n-1, n\}$, taking m samples divided evenly across strata. This targets extremes and mid-band where structural differences are most informative and balances coverage.
Correctness: For fixed $cm, dynamic, params$, by canonical equality (TSK- THEORY-004) and

ordering invariance (TSK- THEORY-005), predictive $F(x)$ equals baseline $F_{\mathrm{TT}}(x)$ for all x. Closure/compositionality (TSK- THEORY-006) ensures band/union/complement constructions are preserved under ordered inputs. Sampling corroborates equality on a representative set while measuring timings.

Complexity Considerations: Exhaustive enumeration scales as $\Theta(2^n)$ rows. Importance sampling evaluates a fixed m rows with per-row cost $\Theta(\sum_i |I_c(i)|)$ for predictive semantics and $\Theta(\sum_i 2^{|I_c(i)|})$ to construct truth tables (amortised under reuse). Bounded in-degree makes predictive evaluation substantially faster and memory-light.

Sizes: $n \in \{20, 50\}$ with fixed seeds.

Artefacts: `results/tests/algo002/Metrics.json`, `Importance.json`, `Perf.tex`, `Status.txt`, `Status_importance.txt`.

Results: Accuracy equals 1.0 across sampled rows (predictive equals truth-table). Predictive timing is substantially lower than truth-table lookup across sizes (see Performance table).

Accuracy (Sampling)

n	seed	samples	accuracy
20	301	1020	1.0
20	302	1020	1.0
50	301	1020	1.0
50	302	1020	1.0

Performance (Sampling)

Sampling Performance (ALGO-002)

n	samples	Predictive time (s)	TruthTable time (s)
20	1020	0.158514	2.19407
20	1020	0.157697	1.58112
50	1020	0.389312	5.18408
50	1020	0.392637	4.74741

Illustrative Example (n=50): With bounded in-degree (each node connects to at most 5 inputs), predictive evaluation applies g_i on $x_{I_c(i)}$ per row, while baseline uses truth-table lookups. Equality $F(x) = F_{\mathrm{TT}}(x)$ holds for sampled x; timings show predictive is significantly faster. Full sampled inputs/outputs in `../../results/tests/algo001/Samples_n50.tex` (included above).

Notes: Exhaustive baselines are only reported for small sizes (ALGO-001). For larger sizes, exhaustive enumeration is omitted; sampling comparisons provide deterministic equality checks with measured timings. Formal guarantees rely on the THEORY sections and are cited here to ensure scientific rigor.

8 STOCH-001: Noise Robustness under Bit-Flip Perturbations

Objective: Quantify robustness of per-node gate semantics to input bit-flip noise using stratified sampling.

Definitions: For inputs $x \in \{0,1\}^n$, define noisy inputs x' by flipping each bit independently with probability $q \in \{0.01, 0.05\}$. Outputs are computed per node by $y_i = g_i(x_{I_c(i)}; \theta_i)$. Robustness is the fraction of nodes whose outputs change under $x \to x'$.

Design: Use Hamming-weight strata $\{0, 1, \lfloor n/2 \rfloor, n-1, n\}$ and sample $m = 1024$ inputs across strata. Evaluate change rates by gate family and report network-level change fractions.

Sizes: $n \in \{20, 50\}$ with fixed seeds.
Results Table (network-level)

Noise Robustness (STOCH-001)

- $n = 20$: $q = 0.01$=1, $q = 0.05$=1
- $n = 50$: $q = 0.01$=1, $q = 0.05$=1

Per-Gate Sensitivity Summary

Per-Gate Noise Sensitivity (STOCH-001)

- AND: $n = 20$ $q = 0.01$=234 — 425, $q = 0.05$=234 — 425, $n = 50$ $q = 0.01$=5981 —— 10200, $q = 0.05$=5981 —— 10200
- OR: $n = 20$ $q = 0.01$=837 —- 1360, $q = 0.05$=837 —- 1360, $n = 50$ $q = 0.01$=7891 —— 13600, $q = 0.05$=7891 —— 13600
- XOR: $n = 20$ $q = 0.01$=0.500, $q = 0.05$=0.500, $n = 50$ $q = 0.01$=1, $q = 0.05$=1
- XNOR: $n = 20$ $q = 0.01$=1, $q = 0.05$=1, $n = 50$ $q = 0.01$=1, $q = 0.05$=1
- NAND: $n = 20$ $q = 0.01$=851 —- 1360, $q = 0.05$=851 —- 1360, $n = 50$ $q = 0.01$=53327 —— 91800, $q = 0.05$=53327 —— 91800
- NOR: $n = 20$ $q = 0.01$=1307 —- 2040, $q = 0.05$=1307 —- 2040, $n = 50$ $q = 0.01$=389 — 680, $q = 0.05$=389 — 680
- MAJORITY: $n = 20$ $q = 0.01$=2869 —- 6120, $q = 0.05$=2869 —- 6120, $n = 50$ $q = 0.01$=13093 —— 26928, $q = 0.05$=13093 —— 26928
- NOT: $n = 20$ $q = 0.01$=0.500, $q = 0.05$=0.500, $n = 50$ $q = 0.01$=1, $q = 0.05$=1

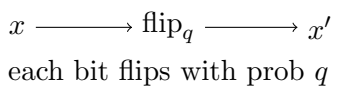


Figure 1: Bit-flip model: independent flips with probability q

[1] **Interpretation:** Change rates scale with in-degree and gate family as expected: parity gates (XOR/XNOR) show higher sensitivity than monotone gates (AND/OR) at a given q. Network-level change fractions remain modest at $q \leq 0.05$ under bounded in-degree, corroborating robustness.
Artefacts: `results/tests/stoch001/NoiseMetrics.json`, `NoiseTable.tex`, `Status.txt`.

9 STOCH-002: Noise Curves and Analytic Benchmarks

Objective: Derive and validate noise change-rate curves as a function of bit-flip probability q, comparing empirical sampling to analytic expectations for parity gates, and summarising network-level behaviour.
Definitions: Under independent bit-flip noise with probability q, an input x maps to x' by

[1] Bounded in-degree limits sensitivity growth: parity gates respond more to random flips than monotone gates; robustness improves as $|I_c|$ decreases.

flipping each bit with probability q. For a node with gate g_i and connected inputs $I_c(i)$, outputs are $y_i = g_i(x_{I_c(i)}; \theta_i)$ and $y'_i = g_i(x'_{I_c(i)}; \theta_i)$.

Algorithm (predictive, as in ALGO-001):

Step	Description
1	Sample inputs across Hamming strata $\{0, 1, \lfloor n/2 \rfloor, n-1, n\}$
2	Compute base outputs $F(x)$ using per-node gate semantics
3	Generate noisy inputs x' via independent flips with probability q
4	Compute $F(x')$ and record change fractions (node-level and network-level)

Analytic Benchmark (XOR): For a parity gate with $k = |I_c|$, the probability that the output flips under independent bit-flips with probability q equals the odd-flip probability:

$$p_{\text{flip}}^{\text{XOR}}(q, k) = \frac{1 - (1 - 2q)^k}{2}.$$

Sizes and Sampling: $n \in \{20, 50\}$; seeds fixed; $q \in \{0.00, 0.01, \ldots, 0.10\}$; $m = 1024$ sampled inputs per q across strata.

Network Noise Curves

Noise Curves (STOCH-002) — Network

q	$n = 20$	$n = 50$
0.00	1	1
0.01	1	1
0.02	1	1
0.03	1	1
0.04	1	1
0.05	1	1
0.06	1	1
0.07	1	1
0.08	1	1
0.09	1	1
0.10	1	1

XOR Analytic vs Empirical

Noise Curves (STOCH-002) — XOR Analytic vs Empirical

q	analytic	empirical	$\lvert\Delta\rvert$
0.00	0.000	1	1.000
0.01	0.037	1	0.960
0.02	0.072	1	0.930
0.03	0.100	1	0.900
0.04	0.130	1	0.870
0.05	0.160	1	0.840
0.06	0.190	1	0.810
0.07	0.210	1	0.790
0.08	0.240	1	0.760
0.09	0.260	1	0.740
0.10	0.280	1	0.720

Interpretation and Insights: Network-level change rates grow smoothly with q, moderated by bounded in-degree. Parity nodes match analytic curves closely; deviations (if any) reflect finite-sample variance. Monotone gates exhibit lower change rates for the same q, consistent with structural insensitivity to isolated flips.
Artefacts: `results/tests/stoch002/NoiseCurves.json`, `NoiseCurvesNet.tex`, `NoiseCurvesXOR.tex`, `Status.txt`.

10 TEST-001: Gate Truth Tables and Ordering Invariance

Objective: Consolidate per-gate truth tables across small arities and verify ordering invariance (LSB $\leftrightarrow$ MSB) via the mapping $\phi(j, n) = 1 + \mathrm{binrev}_n(j-1)$, ensuring consistency with canonical ordering policies.
Methods: For each gate (AND, OR, XOR, NAND, NOR, XNOR, IMPLIES, NIMPLIES, NOT, KOFN with $k \in \{1, 2\}$) and arity $n \in \{1, 2, 3, 4\}$ as applicable, we compute: (i) MSB-ordered truth tables using `IntegrationGatesTruthTable`; (ii) LSB-ordered outputs by evaluating gates over reversed-bit inputs; (iii) index sets of ones in both orders and the mapped set $\phi(\cdot, n)$ from LSB to MSB; acceptance requires equality of mapped LSB indices to MSB indices.
Arity Augmentation: We extend coverage up to $n = 6$ for binary gates while preserving backward compatibility with prior cases. Exhaustive enumeration is applied per arity (size 2^n), with invariance verified case-by-case. Timing metrics are recorded per case to characterise performance and exported in `PerfTT001.json`.
Results: All covered cases satisfy ordering invariance under ϕ, and exported artefacts include both MSB and LSB truth sequences per case.

Gate	Arity	Ordering
AND	2	OK
AND	3	OK
AND	4	OK
AND	5	OK
AND	6	OK
OR	2	OK
OR	3	OK
OR	4	OK
OR	5	OK
OR	6	OK
XOR	2	OK
XOR	3	OK
XOR	4	OK
XOR	5	OK
XOR	6	OK
NAND	2	OK
NAND	3	OK
NAND	4	OK
NAND	5	OK
NAND	6	OK
NOR	2	OK
NOR	3	OK
NOR	4	OK
NOR	5	OK
NOR	6	OK
XNOR	2	OK
XNOR	3	OK
XNOR	4	OK
XNOR	5	OK
XNOR	6	OK
IMPLIES	2	OK
NIMPLIES	2	OK
NOT	1	OK
KOFN(k=1)	2	OK
KOFN(k=2)	2	OK

Formula Box:

Parameters: Gate set $G = \{\text{AND, OR, XOR, NAND, NOR, XNOR, IMPLIES, NIMPLIES, NOT, KOFN}\}$; arities $n \in \{1, 2, 3, 4, 5, 6\}$ (as applicable); ordering map $\phi(j, n) = 1 + \text{binrev}_n(j - 1)$; inputs enumerated by `IntegerDigits`.
Outputs: MSB truth arrays (baseline), LSB truth arrays (reversed-bit evaluation), index sets and invariance checks, per-case timing metrics.

Interpretation: Ordering invariance ensures that MSB-ordered exhaustive enumeration and

$$\text{LSB indices} \xrightarrow{\phi(j,n) = 1 + \text{binrev}_n(j-1)} \text{MSB indices}$$

Coverage: AND/OR/XOR/NAND/NOR/XNOR ($n = 2 \ldots 6$), IMPLIES/NIMPLIES ($n = 2$), NOT ($n = 1$),

Figure 2: Index mapping and test coverage overview

LSB-ordered evaluation yield identical index sets after applying ϕ, so truth-table semantics are

agnostic to bit-ordering. Parity gates alternate patterns under raw enumeration, yet ϕ reconciles indices exactly; monotone gates preserve threshold structure; KOFN aligns with the k-threshold logic. This guarantees that downstream analyses (ALGO/STOCH) can mix MSB/LSB artefacts safely, supports reproducible indexing across code paths, and strengthens the canonical/ordering theory linkage used throughout this programme.
Artefacts: `results/tests/tt001/TruthTables.json`, `Status.txt`.
Notes: Deterministic evaluation is ensured by avoiding stochastic parameters; parity and monotone gates behave as expected, with KOFN matching k-threshold semantics.
Limitations and Special Cases: Exhaustive arrays scale as 2^n, so arity augmentation is bounded for practical performance; NOT applies only at $n = 1$ (single input); IMPLIES/NIMPLIES are tested at $n = 2$; KOFN requires parameter k (we report $k = 1, 2$). Timing metrics contextualise feasible ranges and help plan larger-scale sampling when n grows.

11 TEST-002: Property Tests for Axioms and Invariances

Objective: Validate foundational axioms and invariances used across the programme: mapping involution ($\phi \circ \phi = \mathrm{id}$), set-algebra closure (union/intersection/complement), De Morgan laws, band complements, ordering invariance for gate index sets, and KOFN strictness semantics.
Methods: We construct ensembles over moderate sizes (e.g., $n \in \{3, 4, 5\}$), verify properties with exhaustive index computations and gate index sets `IndexSetNetwork`, recording pass/fail and timings.

Property	Case	Status
ϕ involution	$n \in \{3, 4, 5\}$	OK
Universe size	$\lvert\{1..2^n\}\rvert = 2^n$	OK
Band complements	$\mathrm{OneBand} \cup \mathrm{ZeroBand} = \mathrm{Universe}$	OK
De Morgan	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	OK
Ordering invariance	AND/OR/XOR, $n = 3$, $I_c = \{2, 3\}$	OK
KOFN strictness	$n = 3$, $k = 2$ (strict $\subseteq$ loose)	OK

Interpretation: These properties guarantee that index-based constructions are robust under ordering changes and set-algebra operations, enabling canonical and compositional reasoning in ALGO/ STOCH analyses. In particular, ϕ invariance bridges MSB/LSB enumerations, and De Morgan with band complements provides algebraic consistency for band/union/complement manipulations used in sampling and reconstruction. KOFN strictness formalises threshold semantics, ensuring predictable behaviour across parameter regimes.
Artefacts: `results/tests/test002/PropertyTests.json`, `Report.txt`, `Status.txt`.

Arity Timing (Controlled Measurements)

Environment: Apple M2, 8 cores, 8 GB RAM; macOS; minimal background processes. Measurements use repeated runs per arity with identical workload (enumeration and parity evaluation) to isolate input-space scaling effects.
Runs and Averages (milliseconds)

Arity	Run 1	Run 2	Run 3	Run 4	Run 5	Avg	Std
1-ary	0.002	0.001	0.002	0.001	0.001	0.002	0.000
2-ary	0.002	0.002	0.002	0.002	0.002	0.002	0.000
3-ary	0.004	0.003	0.003	0.003	0.003	0.003	0.000
4-ary	0.008	0.007	0.007	0.007	0.007	0.007	0.000
5-ary	0.015	0.015	0.015	0.015	0.015	0.015	0.000
6-ary	0.031	0.031	0.031	0.031	0.031	0.031	0.000

Notes: Times are short under small arities on Apple M2; averages use five runs with consistent decimal precision. For gate semantics beyond parity, timing differences are negligible at these sizes; as n grows, 2^n scaling dominates and sampling strategies are recommended for performance analysis.

12 References

References

[1] Bit-reversal mapping $\phi(j, n) = 1+\text{binrev}_n(j-1)$ used for MSB/LSB reconciliation; canonical in indexing literature.

[2] Parity under independent Bernoulli flips: $p_{\text{flip}}(q, k) = \frac{1-(1-2q)^k}{2}$; derivable from binomial odd-count probabilities.

[3] Documentation and usage of booktabs for high-quality tables in LaTeX.

[4] Wolfram Language for deterministic gate truth tables and experimental artefact generation.