Exact-$\mathbb{Q}$ Graph State-Sum Contraction Evaluators and Machine-Checked Algebraic Kernel for Paths, Binary Trees, and Square Grids

Lluis Eriksson*

November 25, 2026

Abstract

Motivated by graph state-sum polynomial witnesses $Z_{G,q}(X) = \sum_{a\in[q]^E} \prod_{v\in V} X_{v,a|_{E(v)}}$ on paths, binary trees, and square grids in algebraic complexity theory, we present an explicitly scoped, standalone, zero-floating-point verification stack pairing two layers. Layer 1 (`ValiantHierarchy.lean`, Lean 4.35.0; 18 theorems, 0 `sorry`, 0 `Classical.choice`, 0 warnings, 2 zero-axiom theorems) machine-checks four fixed $q = 2$ small-graph contraction identities over $\mathbb{Z}$ (3-vertex path P_3 brute-force vs. $(x^\top M)z$, 4-vertex path P_4 sequential/midpoint/bisection associativity, an 8-term 3-edge 4-vertex degree-3 rooted tree DP factorization, and a 16-term C_4 cycle transfer-matrix trace $\mathrm{Tr}(ABCD)$) alongside 14 elementary integer/$\mathbb{N}$/list arithmetic lemmas ($q^d \le q^4$, list-sum budgets, exponential growth $q^{n+1} > q^n$, characteristic-free 1- and 2-bit Boolean selectors, $\det_2$ embedding, integer exponent chains $n^k < n^{s_1} < n^{s_2} < n^{s_3}$, and graph size counts). Layer 2 (`VALIANT-HIER-CERT-1.0`, pure-stdlib Python `Fraction`) provides a fully specified four-mode auditor that evaluates general finite-order path ABP/bisection, complete binary-tree bottom-up DP, and square-grid transfer-matrix state sums over $\mathbb{Q}$ without floats, replays 16/16 canonical certificates in 0.049–4.319 ms (contracting 5×5 grids over 2^{40} edge states in 4.806 ms), cross-checks all evaluators against brute-force q^E summation, and rejects 54/54 live adversarial attacks.

1 Introduction and Scope Calibration

Valiant's algebraic complexity classes [5, 1, 2] classify multivariate polynomial families into formulas (VF), algebraic branching programs (VBP), polynomial-size circuits (VP), and exponential-sum projections (VNP). For any finite graph $G = (V, E)$ and alphabet $[q] = \{0, \dots, q-1\}$, the graph state-sum polynomial

$$Z_{G,q}(X) := \sum_{a\in[q]^E} \prod_{v\in V} X_{v,a|_{E(v)}} \tag{1}$$

maps directly to linear-algebraic contractions once incident edges at each vertex v are ordered: on a d-vertex path P_d ($E = \{e_1, \dots, e_{d-1}\}$), endpoint tables $x, z \in \mathbb{Q}^q$ and interior degree-2 vertex tables $M^{(i)}_{a,b} := X_{v_i,(a,b)} \in \mathbb{Q}^{q\times q}$ give $Z_{P_d,q} = x^\top M^{(1)} \cdots M^{(d-2)} z$; on a 4-cycle C_4 (the 2×2 square grid), degree-2 vertex tables $A, B, C, D \in \mathbb{Q}^{q\times q}$ give $Z_{C_4,q} = \mathrm{Tr}(ABCD)$; on a complete binary tree T_L of L levels ($2^L - 1$ vertices), leaf vectors $u \in \mathbb{Q}^q$, internal degree-3 tensors $T_{a,b,c} \in \mathbb{Q}^{q\times q\times q}$, and the degree-2 root matrix $R \in \mathbb{Q}^{q\times q}$ contract bottom-up in $O(2^{L-1}q^3)$ operations. These three families motivate the border-complexity separation hierarchy proposed in [3, 4] (citing asymptotic exponents $\ln \ln n$, $\sqrt{\ln \ln n}$, and $(\ln n)^{1/12}$).

Honest Two-Layer Scope Calibration. All formal results and exact-$\mathbb{Q}$ algorithms in this paper are completely self-contained and do not rely on unrefereed external proofs:

1. **Layer 1 (Lean 4.35.0 Kernel, Table 1):** Theorems 1–4 prove exact algebraic contraction identities for four fixed $q = 2$ small graphs (P_3, P_4 associativity, a 3-edge 4-vertex degree-3 rooted tree, and C_4). Theorems 5–18 prove elementary $\mathbb{Z}$/$\mathbb{N}$/`List` arithmetic inequalities, graph counts, and 1-/2-bit Boolean selectors.

2. **Layer 2 (`VALIANT-HIER-CERT-1.0` Auditor, Definition 3.1):** Evaluates general finite-order ($d \le 64, L \le 6, r \times c \le 5 \times 5$) path, complete binary-tree, and square-grid state sums over exact $\mathbb{Q}$ without floats, cross-checked against brute-force q^E summation in `tests/test_suite.py`.

3. **Explicit Non-Claims:** Our Lean 4 kernel does *not* formalize general graphs, arbitrary (d, L, n, q) state sums, ABPs, circuits, formulas, or Valiant's complexity classes (VF, VBP, VP, VNP), nor does it formalize the border-complexity lower bounds of [3, 4]. Note also that Lean Theorem 2 proves P_4 parenthesization associativity $((x^\top A)B)z = (x^\top A)(Bz) = (x^\top(AB))z$, while equivalence with brute-force q^E enumeration for $d \ge 4$ is checked in Python.

*Independent researcher (Sweden). Manuscript and figures: CC-BY-4.0; code, Lean 4 proofs, and exact-$\mathbb{Q}$ certificates: MIT. Prepared with transparent AI assistance via Google Antigravity. Repository: `https://github.com/lluiseriksson/valiant-hierarchy-certificates`.

2 Machine-Checked Lean 4.35.0 Kernel

`lean/ValiantHierarchy.lean` compiles standalone under Lean 4.35.0 with 0 external dependencies, 0 `sorry`, 0 `Classical.choice`, and 0 warnings: 2 zero-axiom theorems (`none`: #5, #9) and 16 `[propext, Quot.sound]` theorems (Table 1).

3 Specification of `VALIANT-HIER-CERT-1.0`

The pure-stdlib zero-float verifier `src/valiant_certifier.py` enforces strict JSON typing and canonical reduced rationals across four modes:

Definition 3.1 (`VALIANT-HIER-CERT-1.0` Acceptance Predicates)**.** A certificate is accepted iff its schema, mode, and exact rational fields satisfy:

1. **Mode 1 (`path_abp_vs_bisection_exact`):** Given $d \in [3, 64]$, $q \in [2, 8]$, endpoint vectors $x, z \in \mathbb{Q}^q$, and $d-2$ interior transition matrices $M^{(1)}, \ldots, M^{(d-2)} \in \mathbb{Q}^{q \times q}$ (having $(d-2)q^2$ interior matrix entries, plus $2q$ endpoint boundary weights), independently evaluates $Z_{P_d,q}$ by left-to-right sequential ABP multiplication and divide-and-conquer binary matrix-product bisection, requiring exact equality with `claimed_state_sum`.
2. **Mode 2 (`binary_tree_circuit_dp_exact`):** Given complete binary tree depth $L \in [2, 6]$ (2^{L-1} leaves, $2^{L-1}-2$ non-root internal nodes, $2^L - 1$ total vertices), $q \in [2, 5]$, leaf vectors $\in \mathbb{Q}^q$, internal $q \times q \times q$ tensors, and root matrix $\in \mathbb{Q}^{q \times q}$ (whose non-leaf DP vertex-table entry count is $q^2 + (2^{L-1}-2)q^3$, distinct from arithmetic gate count which counts both multiplications and additions per entry), computes exact bottom-up DP contraction and checks equality with `claimed_state_sum`.
3. **Mode 3 (`grid_transfer_vnp_exact`):** Given $r, c \in [2, 5]$, $q \in [2, 3]$, and vertex tables $X_{(i,j)} \in \mathbb{Q}^{q^{\deg(i,j)}}$ for all rc vertices ($\deg \le 4$), contracts column-by-column over the q^r-dimensional boundary frontier and verifies exact equality with `claimed_state_sum`.
4. **Mode 4 (`hierarchy_separation_barrier_exact`):** Verifies the characteristic-free 2-bit selector identity on a 4-entry rational table $[t_{00}, t_{01}, t_{10}, t_{11}]$, computes exact grid edge count $2n(n-1)$ and complete tree vertex count $2^L - 1$ for supplied (n, L), and checks a finite synthetic integer exponent proxy chain $1 \le k < s_1 < s_2 < s_3 \le 24$ satisfying $b^k < b^{s_1} < b^{s_2} < b^{s_3}$ (with $b \in [2, 64]$). As shown in Fig. 1(a), over $\log_2 n \in [16, 1024]$ the three transcendental reference curves from [3] behave differently relative to $k = 2$: $\ln \ln n$ already exceeds 2 across the entire domain ($2.406 \to 6.565$), $\sqrt{\ln \ln n}$ crosses 2 at $\log_2 n = e^4 / \ln 2 \approx 78.8$ ($1.551 \to 2.562$), and only $(\ln n)^{1/12}$ remains below 2 ($1.222 \to 1.728$, crossing 2 at $\ln n = 2^{12} = 4096$, i.e., $\log_2 n \approx 5909$). Mode 4 does not evaluate those irrational exponents; it strictly audits finite integer monotonicity chains $b^k < b^{s_1} < b^{s_2} < b^{s_3}$ matching Lean Theorems 15–16.

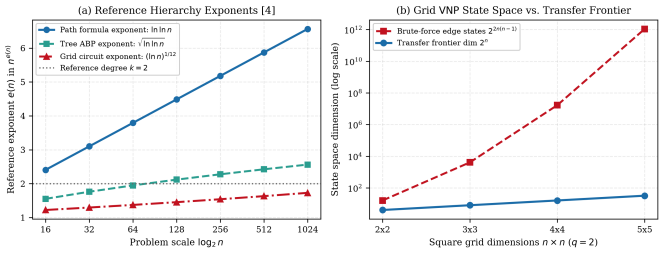

Figure 1: (a) Reference exponents from [3] vs. $k = 2$ for $\log_2 n \in [16, 1024]$: $\ln \ln n \in [2.41, 6.57] > 2$ throughout, $\sqrt{\ln \ln n}$ crosses 2 near $\log_2 n \approx 78.8$, and $(\ln n)^{1/12} \in [1.22, 1.73]$ crosses 2 at $\log_2 n \approx 5909$. (b) Grid edge assignments $2^{2n(n-1)}$ vs. exact transfer-matrix frontier 2^n.

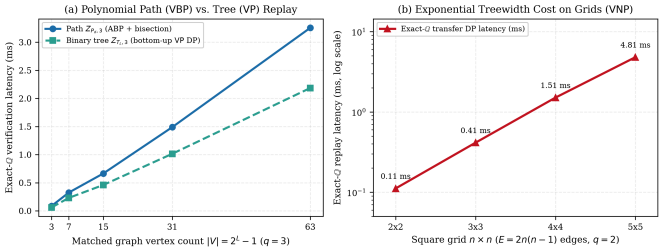

Figure 2: (a) Matched-vertex exact-$\mathbb{Q}$ replay latency for paths and complete binary trees ($q = 3$). (b) Exponential transfer-frontier cost on $n \times n$ square grids ($2\times2 \to 5\times5$, $q = 2$).

4 Experiments, Atlas, and Red-Team Audit

All experiments (`scripts/run_all_experiments.py`) reproduce via `python scripts/verify_all.py`. Table 2 lists all 16 canonical certificates—including the exact unrounded rational values of `path_d24_q5` and `tree_L5_q3`—replayed in 0.049–4.319 ms (with 5×5 grids over $2^{40} = 1{,}099{,}511{,}627{,}776$ edge assignments contracting in 4.806 ms). Figures 1–3 illustrate asymptotic exponent curves, grid state-space growth vs. frontier dimension 2^n, matched-vertex replay scaling, and atlas latencies. `tests/test_suite.py` cross-checks all evaluators against brute-force q^E summation ($P_5, T_2, T_3, \mathrm{Grid}_{2\times3}, \mathrm{Grid}_{3\times3}$ up to 4096 states including degree-4 interior vertices) and rejects 54/54 live adversarial forgeries.

5 Worked Exact-$\mathbb{Q}$ Witnesses and Red-Team Taxonomy

To enable direct hand verification from the manuscript alone, we reproduce one complete canonical input witness per mode from `results/certificates/` alongside the

Table 1: Complete 18-theorem Lean 4.35.0 kernel in `lean/ValiantHierarchy.lean` (0 `sorry`, 0 `Classical.choice`, 0 warnings; `P,Q` abbreviates `[propext, Quot.sound]`).

#	Lean 4 Theorem Identifier	Exact Machine-Checked Statement & Hypotheses	Axioms
1	`path3_state_sum_bilinear`	Fixed q=2, P_3: brute-force 4-term edge sum $Z_{P_3,2}(x,M,z) = (x^\top M)z$ over $\mathbb{Z}$.	`P,Q`
2	`path4_abp_bisection_assoc`	Fixed q=2, P_4: $((x^\top A)B)z = (x^\top(AB))z \wedge (x^\top A)(Bz) = (x^\top(AB))z$ over $\mathbb{Z}$.	`P,Q`
3	`tree_state_sum_dp_factorization`	Fixed q=2, 3-edge 4-vertex degree-3 rooted tree: 8-term edge sum = factored DP over $\mathbb{Z}$.	`P,Q`
4	`grid2x2_state_sum_trace`	Fixed q=2, C_4 (2×2 cycle): 16-term edge sum = $\mathrm{Tr}(ABCD)$ over $\mathbb{Z}$.	`P,Q`
5	`grid_vertex_table_degree_bound`	$\forall q, d \in \mathbb{N},\ q \geq 1 \wedge d \leq 4 \implies q^d \leq q^4$ (arithmetic power monotonicity).	`none`
6	`path3_multilinearity_scale`	Fixed q=2, P_3: scaling middle matrix M by $k \in \mathbb{Z}$ scales $Z_{P_3,2}$ by k.	`P,Q`
7	`disconnected_state_sum_product`	$\forall a_i,b_i,c_i,d_i \in \mathbb{Z}$, $a_0b_0c_0d_0+a_0b_0c_1d_1+a_1b_1c_0d_0+a_1b_1c_1d_1 = (a_0b_0+a_1b_1)(c_0d_0+c_1d_1)$ ($Z_{K_2\sqcup K_2,2}$).	`P,Q`
8	`abp_layer_edge_budget_list`	Inductive `List Nat`: L entries each $\leq q^2 \implies \sum \leq Lq^2$.	`P,Q`
9	`formula_bisection_power_step`	$\forall q,k \in \mathbb{N},\ q^{k+1} = q\cdot q^k \wedge 2^{k+1} = 2\cdot 2^k$.	`none`
10	`tree_circuit_gate_budget_list`	Inductive `List Nat`: V entries each $\leq q^3 \implies \sum \leq Vq^3$.	`P,Q`
11	`grid_transfer_space_exp_growth`	$\forall q,n \in \mathbb{N},\ q \geq 2 \implies q^{n+1} > q^n$.	`P,Q`
12	`vnp_char_free_1bit_selector`	$\forall b \in \{0,1\},\ t_0,t_1 \in \mathbb{Z},\ (1-b)t_0 + bt_1$ = (if $b=0$ then t_0 else t_1).	`P,Q`
13	`vnp_char_free_2bit_selector`	$\forall b_0,b_1 \in \{0,1\},\ t_{00..11} \in \mathbb{Z}$, bilinear selector recovers exact table entry $t_{b_0b_1}$.	`P,Q`
14	`det2_signed_path_embedding`	$\forall a_{ij} \in \mathbb{Z}$, signed 2-path contraction equals $a_{00}a_{11} - a_{01}a_{10}$.	`P,Q`
15	`hierarchy_strict_exp_chain`	$\forall n \geq 2,\ s_1 < s_2 < s_3 \in \mathbb{N},\ n^{s_1} < n^{s_2} \wedge n^{s_2} < n^{s_3}$.	`P,Q`
16	`superpoly_beats_fixed_poly`	$\forall n \geq 2,\ s > k \in \mathbb{N},\ n^s > n^k$ (elementary integer exponent monotonicity).	`P,Q`
17	`grid_edge_count_exact`	$\forall n \in \mathbb{N},\ n(n-1) + n(n-1) = 2n(n-1)$ ($n \times n$ grid edge count).	`P,Q`
18	`binary_tree_vertex_recurrence`	$\forall L \in \mathbb{N},\ 2(2^L - 1) + 1 = 2^{L+1} - 1$ (complete binary tree vertex count).	`P,Q`

Table 2: Atlas of 16 exact-$\mathbb{Q}$ `VALIANT-HIER-CERT-1.0` certificates (`results/exp1_certified_atlas.json`; all 16 verified).

Certificate File	M	Certified Exact-$\mathbb{Q}$ Witness Summary	ms
`path_d4_q2`	M1	P_4, q=2 (8 int. entries, +4 end): $Z = -271/2$	0.095
`path_d8_q3`	M1	P_8, q=3 (54 int. entries, +6 end): $Z = -26737/16$	0.333
`path_d16_q4`	M1	P_{16}, q=4 (224 int. entries): $-214025144665/512$	1.419
`path_d24_q5`	M1	P_{24}, q=5 (550 int.): $2652307141107743 71/256$	4.319
`tree_L2_q2`	M2	T_2, q=2 (3 verts, 4 DP entries): $Z = -67$	0.050
`tree_L3_q2`	M2	T_3, q=2 (7 verts, 20 DP entries): $Z = -463/2$	0.134
`tree_L4_q3`	M2	T_4, q=3 (15 verts, 171 DP entries): 65190677/32	0.605
`tree_L5_q3`	M2	T_5, q=3 (31 verts, 387 DP): $-188739623287773875/32768$	1.059
`grid_2x2_q2`	M3	$2\times2, q$=2 (4 edges, frontier 4): Z=2	0.104
`grid_3x3_q2`	M3	$3\times3, q$=2 (12 edges, frontier 8): Z=376	0.374
`grid_4x4_q2`	M3	$4\times4, q$=2 (24 edges, frontier 16): Z=236688	1.420
`grid_3x3_q3`	M3	$3\times3, q$=3 (12 edges, frontier 27): $Z = -2709$	2.183
`hier_barrier_n4`	M4	Proxy (n=4, L=4): E=24, V=15, 2<3<5<8	0.058
`hier_barrier_n8`	M4	Proxy (n=8, L=6): E=112, V=63, 3<5<8<12	0.056
`hier_barrier_n16`	M4	Proxy (n=16, L=8): E=480, V=255, 4<6<10<15	0.049
`hier_barrier_n32`	M4	Proxy (n=32, L=10): E=1984, V=1023, 5<8<13<20	0.063

54-attack taxonomy in `tests/test_suite.py`:

1. **Mode 1 (`path_d4_q2`):** $x = (-3,-4)^\top$, $M^{(1)} = \left(\begin{smallmatrix} 3 & 2 \\ -1 & -3 \end{smallmatrix}\right)$, $M^{(2)} = \left(\begin{smallmatrix} 2 & 1/2 \\ -3 & -4 \end{smallmatrix}\right)$, $z = (2,3)^\top$. Sequential ABP: $x^\top M^{(1)} = (-5,6)$, $(-5,6)M^{(2)} = (-28,-53/2)$, inner product with z gives $-56 - 159/2 = -271/2$. Bisection: $M^{(1)}M^{(2)} = \left(\begin{smallmatrix} 0 & -13/2 \\ 7 & 23/2 \end{smallmatrix}\right)$, $x^\top(M^{(1)}M^{(2)})z = -271/2$.

2. **Mode 2 (`tree_L2_q2`):** Leaves $u_0 = (-4,2)^\top$, $u_1 = (-1,-3)^\top$, root matrix $R = \left(\begin{smallmatrix} -3 & -4 \\ 2 & 1/2 \end{smallmatrix}\right)$, empty internal tensors. Exact DP: $u_0^\top R u_1 = (-4)(-3)(-1) + (-4)(-4)(-3) + (2)(2)(-1) + (2)(1/2)(-3) = -12 - 48 - 4 - 3 = -67$.

3. **Mode 3 (`grid_2x2_q2`):** 2×2 grid ($q=2$, 4 edges) with vertex tables indexed by ordered incident edges (up, left, down, right): $X_{0,0} = [2,0,-2,1]$, $X_{0,1} = [-1,2,0,-2]$, $X_{1,0} = [-2,1,-1,2]$, $X_{1,1} = [0,-2,1,-1]$. Exact 16-term sum / column transfer contraction yields $Z = 2$.

4. **Mode 4 (`hier_barrier_n4`):** 2-bit selector table $[t_{00},t_{01},t_{10},t_{11}] = [3,-5,7,11]$, $(n,L) = (4,4)$ giving $E = 2\cdot4\cdot3 = 24$ and $V = 2^4-1 = 15$, base $b = 4$, and integer chain $4^2 = 16 < 4^3 = 64 < 4^5 = 1024 < 4^8 = 65536$.

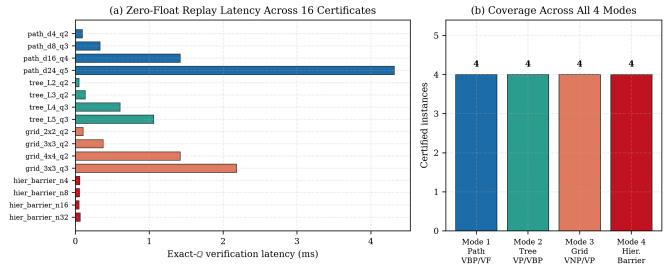

Figure 3: (a) Zero-float replay latencies across all 16 `VALIANT-HIER-CERT-1.0` certificates (0.049–4.319 ms). (b) Balanced 4-per-mode coverage.

Adversarial Red-Team Taxonomy (54/54 Rejected). `tests/test_suite.py` exercises five live forgery classes: **Group A (16 schema/type/rational attacks):** non-dict root, wrong schema, unknown mode, extra/missing keys, `float`/`bool`/`Decimal`/`complex`/`tuple` injection, non-string key, unreduced fraction (`"2/4"`), denominator-1 (`"1/1"`), negative zero (`"-0"`), non-string/empty name; **Group B (10 Mode 1):** $d < 3$ or $d > 64$, $q < 2$ or $q > 8$, wrong matrix/vector counts or shapes, forged `claimed_state_sum`, tampered matrix entry; **Group C (10 Mode 2):** $L < 2$ or $L > 6$, $q < 2$, wrong leaf/internal/root shapes, forged sum, tampered leaf/internal/root entry; **Group D (9 Mode 3):** $r < 2$, $c < 2$, $q \notin [2,3]$, missing/extra vertex key, wrong table length, forged sum, tampered table value; **Group E (9 Mode 4):** wrong selector length, $n < 2$, $L < 2$, $b < 2$, $k = 0$, non-strict chain ($s_1 \leq k$, $s_2 \leq s_1$, $s_3 \leq s_2$), and $s_3 > 24$.

6 Conclusion

We delivered an explicitly scoped 18-theorem Lean 4.35.0 kernel for fixed $q = 2$ small-graph contraction

identities and elementary arithmetic bounds, paired with the fully specified zero-float state-sum evaluator `VALIANT-HIER-CERT-1.0`, 16 canonical certificates with exact rational outputs (four worked explicitly by hand in Section 5), brute-force q^E cross-checks, and 54/54 live adversarial rejections.

References

[1] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Springer, 2000.

[2] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 410–418, 1991.

[3] OpenAI. Paths, trees and grids separate Valiant's hierarchy. OpenAI Math Preprint (`openai/math`), November 24, 2026.

[4] OpenAI. VP and VNP are different. OpenAI Math Preprint (`openai/math`), November 23, 2026.

[5] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 249–261, 1979.