PRIME-FIELD DIGIT-SUM DICHOTOMY AND IMPROVED THREE-REGISTER POWERING EXPONENTS IN PASSIVE CATALYTIC REGISTER PROGRAMS

ALEXANDER S. DONAHUE

Abstract. We prove new results extending Vinciguerra's operator analysis of passive-output catalytic register programs (arXiv:2609.18692, Sep. 2026); no question posed in that paper is resolved.

(1) *Positive-characteristic two-register dichotomy (Theorem 1.1).* Over any prime field $\mathbb{F}_p$, under formal-polynomial cleanliness with $\mathbb{F}_p$ constants, for $t \geq 1$ and $f(x) = \sum_{d \geq 0} \alpha_d x^d \in \mathbb{F}_p[x]$,

$$f \in \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_p}(2,t) \iff \forall\, d \geq 2 \text{ with } \alpha_d \neq 0,\ \ s_p(d) \leq \min(p-1,\, t-1),$$

where $s_p(d)$ is the base-p digit sum of d. Thus the char.-0 lower bounds of [9, Thms. 3.2, 3.3, 3.5] do not extend to char. p ($D^{\mathrm{pass}}_{2,\mathbb{F}_p}(2) = +\infty$ via x^{p^m}), whereas formal polynomials with $s_p(d) \geq p$ (e.g., x^3 over $\mathbb{F}_2$, though $x^3 = x$ pointwise on $\mathbb{F}_2$) have no $\mathbb{F}_p$-constant program for any $t \geq 1$; for $p > \deg f$, this gives a large-characteristic analogue of Theorem 3.5.

(2) *Bilinear two-access Frobenius programs (Theorem 1.2).* Over any char.-2 commutative ring, an explicit bilinear passive program ($R_i \leftarrow R_i + R_j R_\ell$, $j, \ell \neq i$) cleanly computes x^{2^k} in $k+1$ registers, $4k-2$ bilinear ops, and *only two* input accesses ($\forall k \geq 1$).

(3) *Improved three-register powering exponents (Theorem 1.3).* Explicit Pascal-row shear loops give $x^7 \in \mathrm{RP}^{\mathrm{pass}}_K(3,5)$ (char $\neq 2,3$), $x^9 \in \mathrm{RP}^{\mathrm{pass}}_K(3,6)$ (char $\notin \{2,3,5,17\}$), and x^{11}, x^{13}, x^{15} in $7, 8, 9$ accesses over $\mathbb{Q}$ and good primes, lowering the 4-register powering exponent (Cor. 4.3.1) from $\log_5 4 \approx 0.86135$ to $\log_{13} 8 \approx 0.81071$ (smallest among our constructed programs and unique integer minimizer of $\log_{2t-3} t$ for $t \geq 4$, not proved optimal), and tightening paper-level streaming (Cor. 5.1.1) and matrix powering (Thm. 5.2, $O(N^{\log_7 5}) = O(N^{0.82709})$ via `prog5_x7` at $2m^2 + \binom{m^2+2}{2} - 1$ scalars).

(4) *Balanced zero-workspace bilinear product (Theorem 1.4).* Over any commutative ring, the classical four-step recursion (cf. [2]) computes $x_0 \cdots x_{d-1}$ ($d = 2^k$) in $\log_2 d + 2$ registers and $(7d^2 - 4)/3$ ops.

All 20 theorems, corollaries, and ring-soundness bridges (incl. $\mathbb{F}_4$ separation, distinct-operand bilinear collapse, 4-register powering $\forall N \geq 1$ under stated invertibility hypotheses, coordinate-wise R^n powering, and register containment for Theorem 1.4) plus $0 \neq 1$ non-vacuity witnesses are formalized in 14 zero-Mathlib Lean 4.35.0 `module` files using only standard axioms (`propext`, `Classical.choice`, `Quot.sound`) and pass 5 distinct kernels (6 runs: `lean`, `leanchecker` [`--fresh`, `-paranoid`], `con-leche`, `con-ron`, `nanoda_bin`), paired with an 8/8 LRAT-verified CaDiCaL 2.1.2 Boolean XAG audit [3, 8].

1. Introduction and Main Results

Catalytic computation, introduced by Buhrman, Cleve, Koucký, Loff, and Speelman [4], augments a space-bounded machine with a large *catalytic* tape initialized to an arbitrary unknown state τ that must be restored bit-for-bit upon halting. Following the breakthroughs of Cook and

Date: November 26, 2026 (v1.0.1).

2020 *Mathematics Subject Classification.* 68Q06, 68Q15, 11T06, 12E20.

Key words and phrases. Catalytic computing, register programs, finite fields, Lucas's theorem, powering algorithms, streaming algorithms, matrix powering, Lean 4 formal verification.

ORCID: 0009-0006-8168-9648; DOIs: 10.5281/zenodo.24517198 (v1.0.1: 24517199); Code, Blueprint, & SWHID (`f960eeda`). Apache-2.0 / CC-BY-4.0. Not peer reviewed; certified by 5 Lean 4.35 kernels (6 runs: `lean`, `leanchecker` [`--fresh`/`-paranoid`], `con-leche`, `con-ron`, `nanoda_bin`).

Mertz [5] on tree evaluation and Williams [10] on time–space simulation, algebraic *register programs* [6, 2, 1] have emerged as the primary engine for designing clean catalytic algorithms with sharp resource trade-offs.

In September 2026, Vinciguerra [9] introduced a Lie-theoretic operator representation for *passive-output* register programs over fields of characteristic 0, writing $\mathrm{RP}_K^{\mathrm{pass}}(r,t)$ for the univariate polynomials cleanly computable using r registers and at most t input accesses without ever reading the output register, and setting $D_{r,K}^{\mathrm{pass}}(t) = \sup\{\deg f : f \in \mathrm{RP}_K^{\mathrm{pass}}(r,t)\}$. Vinciguerra established three foundational results over characteristic 0:

- $D_{r,K}^{\mathrm{pass}}(1) = D_{r,K}^{\mathrm{pass}}(2) = 1$ and $D_{r,K}^{\mathrm{pass}}(3) = 2$ for every $r \geq 2$ ([9, Theorems 3.2 and 3.3]);
- $D_{2,K}^{\mathrm{pass}}(t) = t - 1$ for every $t \geq 2$ ([9, Theorem 3.5]);
- $x^{2r-1} \in \mathrm{RP}_K^{\mathrm{pass}}(r,4)$ for every $r \geq 2$ and $\mathrm{char}(K) = 0$ or $> 2r-1$ ([9, Theorem 2.3]), which for $r = 3$ yields $x^5 \in \mathrm{RP}_K^{\mathrm{pass}}(3,4)$ and recursive powering exponent $\log_5 4 \approx 0.861353$.

Vinciguerra's operator proofs rely essentially on the formal exponential $\exp(xD) = \sum_{n\geq 0} \frac{x^n}{n!} D^n$ and the Campbell–Baker–Hausdorff formula, which suggests two questions that we study here (distinct from Section 6 of [9], which asks whether $D_{r,K}^{\mathrm{pass}}(4) > 2r-1$ and whether additional accesses drastically increase $D_{r,K}^{\mathrm{pass}}$, both of which stay open):

(Q1) *Positive-characteristic behavior:* What is the computational power of $\mathrm{RP}_{\mathbb{F}_p}^{\mathrm{pass}}(r,t)$ over prime fields $\mathbb{F}_p$ (under formal-polynomial cleanliness with $\mathbb{F}_p$ constants), where $n!$ ceases to be invertible at $n = p$ and $a^p = a$ acts on instruction shifts?

(Q2) *Beyond four input accesses ($t \geq 5$):* Can $r = 3$ programs with $t \geq 5$ input accesses achieve a strictly smaller recursive powering exponent $\log_N t$ than the 4-access base program $x^5 \in \mathrm{RP}_K^{\mathrm{pass}}(3,4)$ ($\log_5 4 \approx 0.861353$)?

In this paper we answer (Q1) completely for $r = 2$ in the formal-polynomial model (and give bilinear programs for arbitrary r in char. 2), establish lower bounds $D_{3,\mathbb{Q}}^{\mathrm{pass}}(t) \geq 2t - 3$ ($t = 5, \ldots, 9$) for (Q2) lowering the 4-register powering exponent from 0.861353 to 0.810714 (not proved optimal), formalize a balanced zero-workspace bilinear product (cf. [2]), and contrast catalytic cancellation with Boolean XAG complexity [3, 8].

1.1. Summary of Main Theorems.

Let $s_p(d) = \sum_{i\geq 0} d_i$ denote the sum of the base-p digits of a nonnegative integer $d = \sum_{i\geq 0} d_i p^i$ ($0 \leq d_i \leq p-1$).

Theorem 1.1 (Base-p Digit-Sum Dichotomy for $\mathrm{RP}_{\mathbb{F}_p}^{\mathrm{pass}}(2,t)$; Lean `T1`)**.** *Let p be any prime, let $t \geq 1$, and let $f(x) = \sum_{d\geq 0} \alpha_d x^d \in \mathbb{F}_p[x]$. Under formal-polynomial cleanliness with program constants in $\mathbb{F}_p$ (Definition 2.1),*

$$f \in \mathrm{RP}_{\mathbb{F}_p}^{\mathrm{pass}}(2,t) \quad \Longleftrightarrow \quad \forall\, d \geq 2 \text{ with } \alpha_d \neq 0, \quad s_p(d) \leq \min(p-1,\, t-1).$$

Theorem 1.1 yields four corollaries (Lean `T1a`–`T1d`):

- *Characteristic-0 lower bounds of* [9, Thms. 3.2, 3.3, 3.5] *do not extend to char. p:* For every prime p and every $m \geq 0$, $x^{p^m} \in \mathrm{RP}_{\mathbb{F}_p}^{\mathrm{pass}}(2,2)$, so $D_{2,\mathbb{F}_p}^{\mathrm{pass}}(2) = +\infty$.
- *Exact $\mathbb{F}_2$ saturation (formal polynomials):* Over $\mathbb{F}_2$, $\mathrm{RP}_{\mathbb{F}_2}^{\mathrm{pass}}(2,t) = \mathrm{RP}_{\mathbb{F}_2}^{\mathrm{pass}}(2,2)$ for all $t \geq 2$; in particular, no $\mathbb{F}_2$-constant program yields the formal polynomial x^3 for any $t \geq 0$ (though $x^3 = x$ pointwise on $\mathbb{F}_2$).
- *Universal impossibility threshold:* Whenever $d \geq 2$ satisfies $s_p(d) \geq p$, no $\mathbb{F}_p$-constant program yields the formal polynomial x^d for *any* $t \geq 1$.
- *Large-characteristic analogue of Theorem 3.5:* Whenever $2 \leq d < p$, $s_p(d) = d$, so $x^d \in \mathrm{RP}_{\mathbb{F}_p}^{\mathrm{pass}}(2,t) \iff t \geq d+1$ (nothing is proved over char.-0 fields here).

One might suspect that $D^{\mathrm{pass}}_{2,\mathbb{F}_p}(2) = +\infty$ is merely an artifact of Definition 2.1 permitting arbitrary-degree univariate basic instructions $O \leftarrow O + R^{p^m}$. Our second theorem rules this out decisively.

Theorem 1.2 (Strictly Bilinear 2-Access Frobenius Collapse; Lean `T2`)**.** *For every $k \geq 1$, there is an explicit passive-output register program on $k+1$ registers whose basic instructions are strictly bilinear updates $R_i \leftarrow R_i + R_j R_\ell$ $(j, \ell \neq i$, output R_0 never read) and whose input accesses are $R_i \leftarrow R_i + x$, which over* every *commutative ring R of characteristic 2 cleanly computes $R_0 \leftarrow R_0 + x^{2^k}$ using $4k-2$ bilinear instructions and **exactly two** input accesses.*

Turning to characteristic 0 (and good prime characteristics) with $r = 3$ registers, we go beyond the 4-access construction $x^5 \in \mathrm{RP}^{\mathrm{pass}}_K(3,4)$ of [9, Theorem 2.3] using explicit elementary shear loops whose work-register updates are pure integer Pascal-row shears $v \leftarrow v \pm qu^d$.

Theorem 1.3 (Three-Register Powering Improvement; Lean `T3a`–`T3g`)**.** *Let K be any commutative ring (in particular any field) in which the scaling integer D_t is invertible. Explicit literal passive-output 3-register programs (Definition 2.1) establish:*

1. $x^7 \in \mathrm{RP}^{\mathrm{pass}}_K(3,5)$ *whenever* $D_5 = 576 = 2^6 \cdot 3^2$ *is invertible (i.e.,* $\mathrm{char}(K) \neq 2, 3$*), with* $\log_7 5 \approx 0.827087$*;*
2. $x^9 \in \mathrm{RP}^{\mathrm{pass}}_K(3,6)$ *whenever* $D_6 = 382{,}500{,}000 = 2^5 \cdot 3^2 \cdot 5^7 \cdot 17$ *is invertible (*$\mathrm{char}(K) \notin \{2, 3, 5, 17\}$*), with* $\log_9 6 \approx 0.815465$*;*
3. $x^{11} \in \mathrm{RP}^{\mathrm{pass}}_K(3,7)$ *whenever* $D_7 = 68{,}724{,}637{,}549{,}388{,}656{,}012{,}800$ *is invertible in K, achieving* $\log_{11} 7 \approx 0.811508$*;*
4. $x^{13} \in \mathrm{RP}^{\mathrm{pass}}_K(3,8)$ *whenever* $D_8 = 2{,}656{,}207{,}148{,}788{,}801{,}777{,}680$ *is invertible in K, achieving* $\log_{13} 8 \approx \mathbf{0.810714}$*;*
5. $x^{15} \in \mathrm{RP}^{\mathrm{pass}}_K(3,9)$ *whenever D_9 (Table 2) is invertible in K, achieving* $\log_{15} 9 \approx 0.811368 > \log_{13} 8$.

Moreover, exact integer exponentiation certifies the strict chain of rational separations

$$\log_{13} 8 < \frac{30}{37} < \log_{11} 7 < \frac{13}{16} < \log_9 6 < \frac{9}{11} < \log_7 5 < \frac{5}{6} < \log_5 4,$$

and unconditionally $\log_{13} 8 < \frac{30}{37} < \log_{2t-3} t$ *for* every *integer $t \geq 4$ with $t \neq 8$ (Lean `T3f`), proving the integer fact that $t = 8$ is the unique minimizer of $\varepsilon(t) = \log_{2t-3} t$ over integers $t \geq 4$ (optimality among all 3-register programs is open; no upper bound is proved).*

Composing Theorem 1.3(4) inside Vinciguerra's powering and streaming framework ([9, Corollaries 4.3.1 and 5.1.1]) tightens the scalar powering exponent in 4 passive registers to $O(N^{0.81072})$ accesses (smaller than $20N^{\log_5 4}$ for $N \gtrsim 5.2 \times 10^4$) and paper-level catalytic streaming frequency moments $F_D(\sigma)$ at the $r = 3$ space tier to $O(D^{0.81072})$ passes, while lifting our quadratic-shear $(d = 2)$ 5-access program `prog5_x7` via [9, §4.4, Theorem 5.2] improves non-commutative matrix powering A^N to $O(N^{0.82709})$ recursive calls at the exact $r = 3$ scalar register count $2m^2 + \binom{m^2+2}{2} - 1$ (with `prog8_x13` achieving $O(N^{0.81072})$ calls at shear-degree-6 scalar register count $3m^2 + \binom{m^2+5}{6}$) — see Section 5.

Finally, on the multivariate bilinear frontier, we formalize a balanced zero-workspace product construction (the classical four-step recursion; cf. [2]) improving the naive linear-chain bound $3 \cdot 4^{d-1} - 2$ to quadratic cost with logarithmic catalysts:

Theorem 1.4 (Balanced Zero-Workspace Bilinear Product; Lean `T4`)**.** *For every $k \geq 0$ and $d = 2^k$, over every commutative ring R, the d-fold multilinear product $x_0 x_1 \cdots x_{d-1}$ is cleanly computed with zero clean workspace, $k + 2 = \log_2 d + 2$ registers (for $k \geq 1$), strictly pairwise-distinct bilinear operand indices ($R_{\mathrm{dst}} \leftarrow R_{\mathrm{dst}} \pm R_a R_b$ with dst, a, b distinct), and exact primitive operation count $(7d^2 - 4)/3$.*

1.2. **Epistemic Verification Tiers.** To enforce transparent verification boundaries, every claim in this manuscript belongs to an explicit tier summarized in Table 1.

TABLE 1. Epistemic verification tiers separating machine-checked Lean proofs, exact Python/SAT computations, paper-level asymptotic reductions, and open problems.

Tier	Scope	Certification Mechanism
Tier-1	Theorems 1.1–1.4, Cor. 5.2 (`T3h`), algebraic Cor. 5.3 (`T3i`), `T1a`–`T1e`, `T1s`, `T1es`, `T2b`, `T3a`–`T3g`	Zero-Mathlib Lean 4.35.0; 5 kernels / 6 runs (Table 5)
Tier-2	Full $\mathbb{F}_4$ ($d \le 16$) spectrum, $r = 4$ loops, Boyar–Find SAT (8/8 LRAT)	Exact $\mathbb{F}_p/\mathbb{Q}/\mathbb{Z}$ arithmetic & CaDiCaL 2.1.2
Tier-3	Bit-space accounting of Cor. 5.3 & Cor. 5.4 (matrix powering)	Paper proof via [9, Cor. 5.1.1, §4.4, Thm. 5.2]
Tier-4	General $D^{\text{pass}}_{3,\mathbb{Q}}(t) = 2t-3$, $r \ge 4$, Boyar–Find	Explicitly open (Section 8.1)

2. Preliminaries: Passive Register Programs

We adopt the algebraic register-program model of Alekseev et al. [1] and Vinciguerra [9, Definition 2.2].

Definition 2.1 (Passive-output register program)**.** Let R be a commutative ring with 1. An R-register program on $r \ge 2$ registers $(R_0, R_1, \dots, R_{r-1})$ (writing $O := R_0$ for the designated output register) over inputs $x_1, \dots, x_n$ is a finite sequence of instructions of two types:

$$R_i \leftarrow R_i + p(R_0, \dots, R_{i-1}, R_{i+1}, \dots, R_{r-1}) \qquad \text{(basic instruction, 0 accesses)},$$
$$R_i \leftarrow R_i + \lambda x_j, \quad \lambda \in R \qquad \text{(input access, 1 access)},$$

where p is an input-independent polynomial over R not depending on the target register R_i. The program is *passive* (with output register O) if O never appears on the right-hand side of any instruction. It *cleanly computes* a polynomial $f(x) \in R[x]$ if, viewing $(O, R_1, \dots, R_{r-1}, x)$ as formal indeterminates in the polynomial ring $R[O, R_1, \dots, R_{r-1}, x]$, executing the instructions terminates in the exact polynomial identity

$$(O_{\text{final}}, R_{1,\text{final}}, \dots, R_{r-1,\text{final}}) = (O + f(x),\ R_1, \dots, R_{r-1}).$$

We write $f \in \mathrm{RP}^{\text{pass}}_R(r,t)$ if f is cleanly computed by a passive r-register program using at most t input accesses, and set $D^{\text{pass}}_{r,R}(t) = \sup\{\deg f : f \in \mathrm{RP}^{\text{pass}}_R(r,t)\}$.

Remark 2.2 (Formal-polynomial vs. functional cleanliness over finite fields)**.** We use the instruction set of [9, Definition 2.2] with *formal-polynomial cleanliness* in $\mathbb{F}_p[O, R_1, \dots, R_{r-1}, x]$ ($\mathbb{F}_p$ constants, valid over every commutative $\mathbb{F}_p$-algebra). Definition 2.2 of [9] is functional (pointwise clean execution), which differs over $\mathbb{F}_p$ since $x^p = x$: $O \leftarrow O + x$ maps $(o, u) \mapsto (o + x^3, u)$ on $\mathbb{F}_2$, whereas Theorem 1.1 (`T1b`) excludes formal x^3. Lean `T1s` certifies formal cleanliness $\Rightarrow$ literal ring execution; the converse is a paper-level argument.

3. Theorem 1.1: Base-p Digit-Sum Dichotomy for Two Registers

Over characteristic 0, Vinciguerra [9, Theorem 3.5] proved $D^{\text{pass}}_{2,K}(t) = t - 1$. In characteristic $p > 0$, two competing phenomena interact (visualized across prime fields in Figure 1): (i) the Frobenius identity $a^p = a$ on $\mathbb{F}_p$ collapses high powers a^d to $a^{s_p(d)}$, allowing degrees such as $d = p^m$ to be isolated in only two accesses; (ii) conversely, because the register shift a ranges only over the p-element field $\mathbb{F}_p$, Lucas's theorem couples a^d to a strictly smaller binomial sub-exponent $j < d$ whenever $s_p(d) \ge p$, making such degrees permanently unreachable in two passive registers regardless of t.

Proof of Theorem 1.1. Let P be any passive two-register program over $\mathbb{F}_p$ on (O, R) with at most t input accesses. Since O is never read and a basic instruction on R cannot read R, every instruction in P has one of four forms: $O \leftarrow O + q(R)$ $(q \in \mathbb{F}_p[z])$, $R \leftarrow R + c$ $(c \in \mathbb{F}_p)$, $O \leftarrow O + \lambda x$, or $R \leftarrow R + \lambda x$. At every step, the work register holds $R + b + ax$ for scalars $b, a \in \mathbb{F}_p$ initially $(0, 0)$.

Step 1: Cycle-closure normal form and homogeneous decoupling. Let $t_R \le t$ be the number of input accesses to R. Because P is clean, the sequence of x-coefficients $a_0 = 0, a_1, \dots, a_{t_R} = 0$ visited by R forms a closed walk returning to 0. Hence every $a \in \mathbb{F}_p$ held by R during any instruction belongs to the list $A = [a_0]$ (if $t_R = 0$) or $A = [a_0, \dots, a_{t_R-1}]$ (if $t_R \ge 1$), which has length $|A| = \max(1, t_R) \le t$ (since $t \ge 1$). For any output update $O \leftarrow O + q(R)$ executed at state (b, a), the trinomial expansion of $(R + b + ax)^k = \sum_{d=0}^{k} \binom{k}{d} b^{k-d} (R + ax)^d$ shows that the coefficient of $R^{d-j}x^j$ $(0 \le j \le d)$ is

$$\binom{d}{j} a^j \sum_{k \ge d} ([z^k]q) \binom{k}{d} b^{k-d}.$$

Because input accesses $O \leftarrow O + \lambda x$ only modify the $(d_R, d_x) = (0, 1)$ term, summing over all output instructions in P yields, for every $d \ge 2$ and every $0 \le j \le d$, a list of weighted shifts $\{(a_\ell, h_\ell)\}$ with all $a_\ell \in A$ such that

$$\binom{d}{j} \sum_{\ell} h_\ell a_\ell^j \equiv \begin{cases} \alpha_d \pmod p & \text{if } j = d, \\ 0 \pmod p & \text{if } 0 \le j < d. \end{cases} \tag{1}$$

Step 2: Lucas's theorem and Fermat digit-sum collapse. By Lucas's theorem [7], $\binom{d}{j} \equiv \prod_i \binom{d_i}{j_i} \pmod p$ where $d = \sum_i d_i p^i$ and $j = \sum_i j_i p^i$. In particular, $\binom{d}{j} \not\equiv 0 \pmod p$ if and only if $j_i \le d_i$ for every digit position i. By Fermat's Little Theorem ($a^p \equiv a \pmod p$ for all $a \in \mathbb{F}_p$, including $a = 0$ since $0^0 = 1$ and $0^n = 0$ for $n \ge 1$), induction on n gives

$$a^n \equiv a^{s_p(n)} \pmod p \qquad \text{for all } a \in \mathbb{F}_p,\ n \ge 0. \tag{2}$$

Now fix any $d \ge 2$ with $\alpha_d \not\equiv 0 \pmod p$, and set $s := s_p(d)$. For every integer r with $0 \le r < s$, greedily choosing digits $0 \le j_i \le d_i$ summing to r produces $0 \le j < d$ with $s_p(j) = r$ and $\binom{d}{j} \not\equiv 0 \pmod p$. Since p is prime, (1) and (2) force

$$\begin{aligned} &\sum_{\ell} h_\ell a_\ell^r \equiv \sum_{\ell} h_\ell a_\ell^j \equiv 0 \pmod p \qquad (\forall\, 0 \le r < s), \\ &\sum_{\ell} h_\ell a_\ell^s \equiv \sum_{\ell} h_\ell a_\ell^d \equiv \alpha_d \not\equiv 0 \pmod p. \end{aligned} \tag{3}$$

Step 3: Necessity ($\Rightarrow$). First, suppose for contradiction that $s \ge p$. Set $r := s - (p - 1)$; then $1 \le r < s$, so $\sum_\ell h_\ell a_\ell^r \equiv 0 \pmod p$ by (3). Yet $s = (r - 1) + p$, so for every $a \in \mathbb{F}_p$, $a^s = a^{r-1}a^p \equiv a^{r-1}a = a^r \pmod p$, giving $\alpha_d \equiv \sum_\ell h_\ell a_\ell^s \equiv \sum_\ell h_\ell a_\ell^r \equiv 0 \pmod p$, contradiction. Thus $s \le p - 1$.

Second, suppose for contradiction that $s \ge t$. Since $|A| \le t \le s$, form the monic degree-s polynomial

$$Q(z) := z^{s-|A|} \prod_{a \in A} (z - a) = z^s + \sum_{r=0}^{s-1} q_r z^r \in \mathbb{Z}[z].$$

Because $a_\ell \in A$ for every ℓ, we have $Q(a_\ell) = 0$ identically, whence

$$0 = \sum_{\ell} h_\ell Q(a_\ell) \equiv \sum_{\ell} h_\ell a_\ell^s + \sum_{r=0}^{s-1} q_r \sum_{\ell} h_\ell a_\ell^r \equiv \alpha_d + 0 = \alpha_d \not\equiv 0 \pmod p,$$

contradiction. Therefore $s_p(d) \le \min(p - 1, t - 1)$.

Step 4: Sufficiency ($\Leftarrow$) via forward finite differences. If $t = 1$, then $\min(p-1, 0) = 0$ forces $\alpha_d \equiv 0 \pmod p$ for all $d \geq 2$ (as $s_p(d) \geq 1$ for $d \geq 1$), so $O \leftarrow O + \alpha_0$; $O \leftarrow O + \alpha_1 x$ cleanly computes f in one access. Assume now $t \geq 2$ and set $M := \min(p-1, t-1) \geq 1$. For any $1 \leq s \leq M < p$, the s-th iterated forward finite difference $(\Delta^s g)(0) = \sum_{a=0}^{s} (-1)^{s-a} \binom{s}{a} g(a)$ satisfies $(\Delta^s (y \mapsto y^r))(0) = 0$ for $0 \leq r < s$ and $= s!$ for $r = s$. Since $s < p$, $p \nmid s!$, so $s!$ admits an inverse $u_s \in \mathbb{F}_p^*$. Defining $w_{s,a} := u_s (-1)^{s-a} \binom{s}{a} \bmod p$ for $0 \leq a \leq s$ (and 0 for $a > s$) yields $\sum_{a=0}^{s} w_{s,a} a^r \equiv \delta_{r,s} \pmod p$ for all $0 \leq r \leq s$.

Consider the $(M+1)$-access program visiting shifts $a = 0, 1, \ldots, M$ via M unit steps $R \leftarrow R + x$ and one closing step $R \leftarrow R + (p - M)x$ (total accesses $M + 1 \leq t$, restoring R cleanly). At shift $a \in \{0, \ldots, M\}$, execute $O \leftarrow O + P_a(R)$ where $[z^0]P_a = \alpha_0 \mathbf{1}_{a=0}$ and for $d \geq 1$ with $\alpha_d \neq 0$ (noting $s_p(1) = 1 \leq M$ as well), $[z^d]P_a = \alpha_d w_{s_p(d),a} \bmod p$. For every (d_R, d_x) with $d = d_R + d_x \geq 1$ and $\alpha_d \neq 0$, the accumulated coefficient of $R^{d_R} x^{d_x}$ is $\alpha_d \binom{d}{d_x} \sum_{a=0}^{s_p(d)} w_{s_p(d),a} a^{d_x} \pmod p$. If $\binom{d}{d_x} \equiv 0 \pmod p$, this vanishes (and $d_x \neq d$). If $\binom{d}{d_x} \not\equiv 0 \pmod p$, Lucas's theorem ensures $s_p(d_x) \leq s_p(d)$ with strict inequality whenever $d_x < d$ (i.e., $d_R > 0$). By (2), the inner sum reduces to $\sum_{a=0}^{s_p(d)} w_{s_p(d),a} a^{s_p(d_x)} \equiv \mathbf{1}_{d_R=0} \pmod p$, yielding α_d when $d_R = 0$ and 0 when $d_R > 0$. □

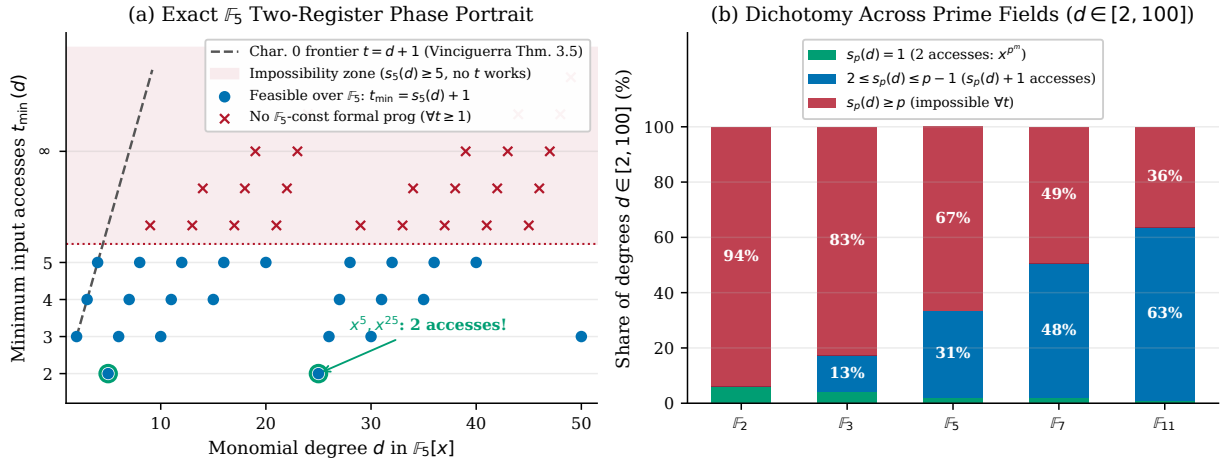


FIGURE 1. Visual phase diagram of Theorem 1.1. (a) Exact minimum input accesses $t_{\min}(d) = s_5(d) + 1$ over $\mathbb{F}_5$ for $d \in [2, 50]$, contrasting Vinciguerra's characteristic-0 line $t = d + 1$ against the 2-access Frobenius powers (x^5, x^{25}) and the permanent impossibility region $s_5(d) \geq 5$. (b) Share of degrees $d \in [2, 100]$ in each regime across $\mathbb{F}_2, \mathbb{F}_3, \mathbb{F}_5, \mathbb{F}_7, \mathbb{F}_{11}$ (exact by Theorem 1.1; $\mathbb{F}_2$–$\mathbb{F}_7$ also independently audited by RREF in `exp1`).

Remark 3.1 (Sharpness over extension fields $\mathbb{F}_{p^e}$ ($e \geq 2$); Lean `T1e`)**.** Theorem 1.1 relies specifically on $a^p = a$ for prime fields $\mathbb{F}_p$. Over a non-prime finite field $\mathbb{F}_q$ ($q = p^e$, $e \geq 2$), the multiplicative group has order $q - 1$ rather than $p - 1$, so a^j reduces modulo $q - 1$ on $\mathbb{F}_q^*$ (while keeping $0^0 = 1$ distinct from $0^j = 0$ for $j \geq 1$). Over $\mathbb{F}_4 = \mathbb{F}_2[\omega]/(\omega^2 + \omega + 1)$, our Tier-1 Lean 4.35 theorem `T1e` (`ExtensionFieldSeparation.lean`) formally proves the exact sharp separation: (i) $x^5 \in \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_4}(2, 3)$ via $\omega^2 u^5 + \omega(u + x)^5 + (u + \omega x)^5 = x^5$ and $x^5 \notin \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_4}(2, 2)$ (exact $t_{\min} = 3$); (ii) $x^3 \in \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_4}(2, 4)$ via $u^3 + (u+x)^3 + (u+\omega x)^3 + (u+\omega^2 x)^3 = x^3$ and $x^3 \notin \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_4}(2, 3)$ (exact $t_{\min} = 4$, both strictly breaking $\mathbb{F}_2$ saturation); and (iii) $x^7 \notin \mathrm{RP}^{\mathrm{pass}}_{\mathbb{F}_4}(2, t)$ for *all* $t \geq 0$, because $a^7 = a$ for every $a \in \mathbb{F}_4$ forces $[x^7]\Delta O = [R^6 x]\Delta O = 0$. Exact finite-field linear algebra for

$d \in [1, 16]$ (`exp1`, Tier-2) further confirms $t_{\min} = 2$ for $d \in \{1, 2, 4, 8, 16\}$, $t_{\min} = 3$ for $d \in \{5, 10\}$, $t_{\min} = 4$ for $d \in \{3, 6, 9, 12\}$, and impossibility for all t at $d \in \{7, 11, 13, 14, 15\}$.

Remark 3.2 (Literal commutative-ring soundness bridge; Lean `T1s`, `T1es`)**.** To eliminate any trust-base gap between the symbolic trinomial-coefficient semantics (`Pass2Prog.exec`, `Pass2GF4Prog.exec`) and literal ring evaluation, `Pass2Soundness.lean` proves the binomial theorem from first principles in zero-Mathlib Lean 4.35 (`CRing.add_pow`) and certifies that direct ring execution (using only ring addition, multiplication, and powers, with zero binomial formulas in its definition) agrees identically with symbolic execution on *every* program over every characteristic-p commutative ring R (`T1s`) and every commutative $\mathbb{F}_4$-algebra (R, φ) (`T1es`).

4. Theorem 1.2: Strictly Bilinear Two-Access Collapse over Characteristic 2

To prove that the failure of characteristic-0 access lower bounds in characteristic 2 is not caused by high-degree basic instructions $O \leftarrow O + R^{2^k}$, we restrict basic instructions strictly to bilinear products $R_i \leftarrow R_i + R_j R_\ell$ ($j, \ell \neq i$, passive output R_0 never read) and input accesses to $R_i \leftarrow R_i + x$.

Proof of Theorem 1.2. Let R be any commutative ring of characteristic 2 ($\forall a \in R,\ a + a = 0$), so Freshman's Dream $(a + b)^2 = a^2 + b^2$ holds identically. Fix $k \geq 1$ and registers $(R_0, R_1, \ldots, R_k)$ initialized to arbitrary catalytic values $(c_0, c_1, \ldots, c_k) \in R^{k+1}$. Define the forward and backward squaring cascades ($k - 1$ bilinear ops each, zero input accesses):

$$\mathsf{Fwd}_k := \left(R_{j+1} \leftarrow R_{j+1} + R_j^2\right)_{j=1}^{k-1}, \qquad \mathsf{Bwd}_k := \left(R_{j+1} \leftarrow R_{j+1} + R_j^2\right)_{j=k-1}^{1}.$$

Execute the two-phase passive-output program `frobBilinProg`:

$$\text{Phase 1:}\quad (R_1 \leftarrow R_1 + x) \ \to\ \mathsf{Fwd}_k \ \to\ (R_0 \leftarrow R_0 + R_k^2) \ \to\ \mathsf{Bwd}_k,$$
$$\text{Phase 2:}\quad (R_1 \leftarrow R_1 + x) \ \to\ \mathsf{Fwd}_k \ \to\ (R_0 \leftarrow R_0 + R_k^2) \ \to\ \mathsf{Bwd}_k.$$

Let $\Gamma_j(c) := \sum_{i=1}^{j} c_i^{2^{j-i}}$ be the value deposited in R_j by Fwd_k on initial state c. In Phase 1, after the first input access $R_1 \leftarrow c_1 + x$, induction using $(a + b)^2 = a^2 + b^2$ shows that Fwd_k sets $R_k = \Gamma_k(c) + x^{2^{k-1}}$. Therefore the first output update adds

$$d_1 = \left(\Gamma_k(c) + x^{2^{k-1}}\right)^2 = \Gamma_k(c)^2 + x^{2^k}$$

to R_0. Because every instruction $R_{j+1} \leftarrow R_{j+1} + R_j^2$ is self-inverse in characteristic 2 and disjoint from R_0, Bwd_k restores $(R_2, \ldots, R_k)$ to $(c_2, \ldots, c_k)$, and the second input access $R_1 \leftarrow (c_1 + x) + x = c_1$ restores R_1 cleanly. In Phase 2, Fwd_k runs on the unperturbed catalysts and sets $R_k = \Gamma_k(c)$, so the second output update adds $d_2 = \Gamma_k(c)^2$ to R_0, yielding $R_0 = c_0 + d_1 + d_2 = c_0 + x^{2^k}$, while Bwd_k restores $(R_2, \ldots, R_k)$ identically. The program performs exactly 2 input accesses and $4(k - 1) + 2 = 4k - 2$ bilinear instructions, and never reads R_0. □

Remark 4.1 (Strictly distinct bilinear operands; Lean `T2b`)**.** If self-squares $R_j \cdot R_j$ are additionally disallowed, adding a single swap catalyst R_{k+1} ($k + 2$ registers total) lets each squaring step $R_i \leftarrow R_i + R_j^2$ be replaced by passive linear copies plus strictly pairwise-distinct bilinear updates ($R_{k+1} \leftarrow R_{k+1} + R_j$; $R_i \leftarrow R_i + R_j R_{k+1}$; $R_{k+1} \leftarrow R_{k+1} + R_j$; $R_i \leftarrow R_i + R_j R_{k+1}$), preserving exact cleanliness and keeping the input-access count fixed at 2. This translated construction is formally machine-checked in Lean 4.35 as Theorem `T2b` (`t2b_distinct_operand_frobenius` in `BilinearFrobenius.lean`).

5. Theorem 1.3: Improved Three-Register Powering, Streaming, and Matrix Powering

We now turn to characteristic 0 (and good prime characteristics) on $r = 3$ registers (O, u, v), where Vinciguerra [9, Theorem 2.3] achieved $x^5 \in \mathrm{RP}_K^{\mathrm{pass}}(3, 4)$ and recursive exponent $\log_5 4 \approx 0.861353$.

5.1. **Elementary Pascal-Row Shear Loops.** Fix $t \geq 4$, distinct integer shifts $s_0, s_1, \ldots, s_{t-1} \in \mathbb{Z}$ with $s_0 = 0$, and an auxiliary shear degree $2 \leq d \leq t-2$. Let $a_0, \ldots, a_{t-1} \in \mathbb{Z}$ be a cleared-denominator Lagrange annihilation vector satisfying

$$\sum_{m=0}^{t-1} a_m (u + s_m x)^d = 0 \qquad \text{identically in } \mathbb{Z}[u, x].$$

For uniform consecutive shifts $s_m = m$ $(0 \leq m \leq t-1)$, a_m is simply the signed binomial row $a_m = (-1)^{t-1-m}\binom{t-1}{m}$. Define prefix sums $Q_0 = 0$ and $Q_{k+1} = Q_k - a_k$ for $0 \leq k \leq t-1$, so that $Q_t = -\sum_{m=0}^{t-1} a_m = 0$ and $Q_k - Q_{k+1} = a_k$. For uniform shifts, $Q_k = (-1)^{t-k}\binom{t-2}{k-1}$ for $1 \leq k \leq t-1$ is literally Pascal's triangle row $t-2$!

Observe that the following literal Definition 2.1 loop on (O, u, v) requires *no* exponential conjugation lemmas and uses *exactly* t *input accesses* ($s_t := s_0 = 0$):

$$\text{For } k = 0, \ldots, t-1: \quad \begin{cases} O \leftarrow O + F_k(u, v) & \text{(passive output, 0 accesses)}, \\ v \leftarrow v - Q_{k+1} u^d & \text{(shear, 0 accesses; omitted if } Q_{k+1} = 0), \\ u \leftarrow u + (s_{k+1} - s_k)x & \textbf{(1 input access)}, \\ v \leftarrow v + Q_{k+1} u^d & \text{(shear, 0 accesses; omitted if } Q_{k+1} = 0). \end{cases} \tag{4}$$

Lemma 5.1 (Exact Loop Closure over $\mathbb{Z}$)**.** *Starting from (O_0, u_0, v_0), immediately before step k $(0 \leq k \leq t-1)$ the work registers hold*

$$u_k = u_0 + s_k x, \qquad v_k = v_0 + Q_k u_k^d + \sum_{j=0}^{k-1} a_j u_j^d.$$

After step $t-1$, the work registers return identically to (u_0, v_0) in $\mathbb{Z}[x, u_0, v_0]$.

Proof. At $k = 0$, $s_0 = 0$ and $Q_0 = 0$, giving (u_0, v_0). Assuming the invariant at step k, subtracting $Q_{k+1} u_k^d$ updates v to $v_0 + (Q_k - Q_{k+1})u_k^d + \sum_{j=0}^{k-1} a_j u_j^d = v_0 + \sum_{j=0}^{k} a_j u_j^d$. Shifting u by $(s_{k+1} - s_k)x$ sets $u = u_{k+1}$, and adding $Q_{k+1} u_{k+1}^d$ restores the invariant for $k+1$. At $k = t$, $s_t = 0$ gives $u_t = u_0$, while $Q_t = 0$ gives $v_t = v_0 + \sum_{j=0}^{t-1} a_j (u_0 + s_j x)^d = v_0$. □

Choosing each output polynomial $F_k(u, v) = \sum_{j=0}^{\lfloor N/d \rfloor} c_{k,j} u^{N-dj} v^j$ to be weighted-homogeneous of degree N (with $\mathrm{wt}(u) = 1$, $\mathrm{wt}(v) = d$) yields a finite linear system over $\mathbb{Q}$ for the equation $\sum_{k=0}^{t-1} F_k(u_k, v_k) = x^N$. Clearing denominators produces pure integer coefficients $c_{k,j} \in \mathbb{Z}$ and a scaling integer $D_t \in \mathbb{N}^+$ such that $\sum_{k=0}^{t-1} F_k(u_k, v_k) = D_t x^N$ in $\mathbb{Z}[x, u_0, v_0]$. Over any commutative ring K in which D_t is invertible, scaling every F_k by D_t^{-1} yields an exact monic program cleanly computing $O \leftarrow O + x^N$ in t input accesses.

5.2. **Explicit Certified Programs for $t = 5, 6, 7, 8$ (and $t = 9$).** Table 2 summarizes our synthesized and verified 3-register programs built via loop (4) and Lemma 5.1 across $t = 4, \ldots, 9$. Crucially, at $t = 5$, choosing quadratic shear degree $d = 2$ (rather than $d = t - 2 = 3$, which saturates at $N = 5$) unlocks $x^7 \in \mathrm{RP}_K^{\mathrm{pass}}(3, 5)$ with scaling integer $D_5 = 576 = 2^6 \cdot 3^2$, so its bad primes are *only* $\{2, 3\}$ — strictly improving Vinciguerra's $t = 4$ exponent $0.861353 \to 0.827087$ under the *exact same* characteristic restriction ($\mathrm{char}(K) \neq 2, 3$, valid over $\mathbb{F}_5$ and $\mathbb{F}_7$!). For $t = 6, 7, 8, 9$, taking $d = t - 2$ gives sparse 3-term output polynomials per step (18, 21, 24, 27 nonzero terms total),

reaching x^9, x^{11}, x^{13}, and x^{15}, with *all five* programs ($t = 5, 6, 7, 8, 9$) formally certified by pure kernel reduction in Lean 4.35.0 (`ThreeRegisterBreakthrough.lean`, Theorems `T3a`–`T3d` and `T3g`). Moreover, Theorem `T3f` (`t3f_global_integer_optimality`) constructively proves in Lean 4.35 via the single rational separator $30/37$ that $8^{37} < 13^{30}$ and $(2t-3)^{30} < t^{37}$ for *every* integer $t \geq 4$ with $t \neq 8$ (checking $t < 20$ explicitly and bounding $(2t-3)^{30} \leq 2^{30}t^{30} < 20^7 t^{30} \leq t^{37}$ for all $t \geq 20$), certifying that $t = 8$ is the unique minimizer of $\log_{2t-3} t$ over integers $t \geq 4$ (an integer-arithmetic fact: $\log_{13} 8$ is the smallest exponent among our constructed programs, whereas optimality among all 3-register programs is open—if $D^{\mathrm{pass}}_{3,\mathbb{Q}}(4) \geq 7$, $\log_7 4 \approx 0.7124$ would be smaller). Explicit integer coefficients for our two smallest-denominator programs ($t = 5 \to x^7$ and $t = 6 \to x^9$) are listed in Tables 3 and 4; the full 21-term ($t = 7$), 24-term ($t = 8$), and 27-term ($t = 9$) integer certificates are compiled directly inside `lean/Catalytic/ThreeRegisterBreakthrough.lean`, and the resulting degree frontier and exponent curve $\varepsilon(t) = \log_{2t-3} t$ are plotted in Figure 2.

TABLE 2. Three-register passive powering frontier $D^{\mathrm{pass}}_{3,\mathbb{Q}}(t) \geq 2t - 3$ ($t = 4, \ldots, 9$). Bold row marks the unique integer minimizer of $\varepsilon(t) = \log_{2t-3} t$.

t	d	Shifts	N	$\log_N t$	Terms	Bad primes of D_t	Certification
4	2	$[0, \ldots, 3]$	5	0.861353	12	$\{2, 3\}$	Baseline [9]
5	2	$[0, \ldots, 4]$	7	0.827087	18	$\{2, 3\}$	Lean 4.35 (`T3a`)
6	4	$[0, \ldots, 5]$	9	0.815465	18	$\{2, 3, 5, 17\}$	Lean 4.35 (`T3b`)
7	5	$[0, \ldots, 5, 7]$	11	0.811508	21	$\{2, 3, 5, 7, 538423, 2512452073\}$	Lean 4.35 (`T3c`)
8	**6**	$[\mathbf{0}, \ldots, \mathbf{7}]$	**13**	**0.810714**	**24**	$\{2, 3, 5, 7, 137, 1602207194491\}$	**Lean 4.35 (`T3d`)**
9	7	$[0, \ldots, 7, 9]$	15	0.811368	27	$\{2, 3, 5, 7, 23, \ldots\}$	Lean 4.35 (`T3g`)

TABLE 3. Explicit 5-access, 3-register passive program `prog5_x7` computing $O \leftarrow O + 576\,x^7$ cleanly over every commutative ring ($d = 2$, shifts $[0, 1, 2, 3, 4]$, bad primes only $\{2, 3\}$).

Step k	Q_{k+1}	α_k	u^7	u^5v	u^3v^2	uv^3
0	-1	$+1$	$-1{,}567$	$-5{,}292$	$-7{,}245$	$-3{,}570$
1	$+3$	$+1$	$-73{,}006$	$+92{,}358$	$-29{,}610$	$+9{,}730$
2	-3	$+1$	0	$-174{,}132$	0	$-12{,}320$
3	$+1$	$+1$	$+73{,}006$	$+92{,}358$	$+29{,}610$	$+9{,}730$
4	0	-4	$+1{,}567$	$-5{,}292$	$+7{,}245$	$-3{,}570$

TABLE 4. Explicit antisymmetric 6-access, 3-register passive program `prog6_x9` computing $O \leftarrow O + 382{,}500{,}000\,x^9$ ($d = 4$, shifts $[0, 1, 2, 3, 4, 5]$, bad primes $\{2, 3, 5, 17\}$). Note $F_{5-k} = -F_k$.

Step k	Q_{k+1}	α_k	u^9	u^5v	uv^2
0	$+1$	$+1$	$+4{,}154{,}985$	$-9{,}394{,}056$	$+5{,}454{,}540$
1	-4	$+1$	$-72{,}618{,}275$	$-30{,}013{,}200$	$-12{,}729{,}150$
2	$+6$	$+1$	$+193{,}081{,}500$	$-8{,}505{,}000$	$+10{,}914{,}750$
3	-4	$+1$	$-193{,}081{,}500$	$+8{,}505{,}000$	$-10{,}914{,}750$
4	$+1$	$+1$	$+72{,}618{,}275$	$+30{,}013{,}200$	$+12{,}729{,}150$
5	0	-5	$-4{,}154{,}985$	$+9{,}394{,}056$	$-5{,}454{,}540$

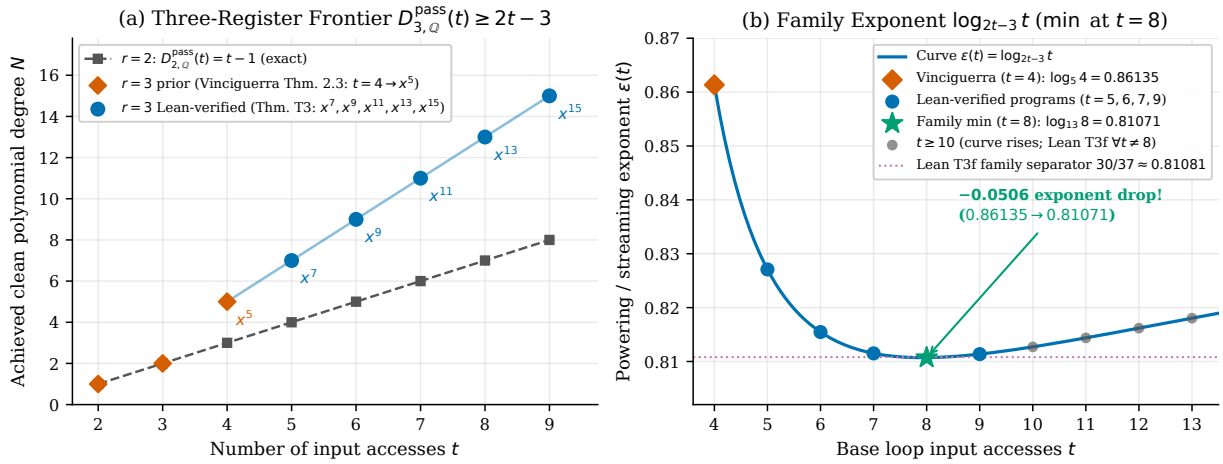


FIGURE 2. Theorem 1.3 three-register powering improvement. (a) Achieved clean polynomial degrees $D^{\text{pass}}_{3,\mathbb{Q}}(t) \geq 2t-3$ versus the two-register ceiling $t-1$ and Vinciguerra's $t=4 \to x^5$ baseline. (b) Composition exponent curve $\varepsilon(t) = \log_{2t-3} t$, decreasing strictly from $\log_5 4 = 0.86135$ to its unique integer minimizer $\log_{13} 8 = 0.81071$ at $t=8$ before rising for $t \geq 9$.

5.3. **Downstream Improvements to Powering, Streaming, and Matrix Powering.** Substituting $x^{13} \in \mathrm{RP}^{\text{pass}}_K(3,8)$ (or $x^7 \in \mathrm{RP}^{\text{pass}}_K(3,5)$) into the Composition Lemma of [1] (cited as Lemma 8 in [9]; Lemma 2.1 in `arXiv:2504.17412v1`) and [9, Cor. 4.3.1, 5.1.1, §4.4, Thm. 5.2] yields Tier-1 four-register powering (Cor. 5.2, `T3h`), coordinatewise R^n powering (`T3i`) backing paper-level streaming (Cor. 5.3), and Tier-3 matrix powering (Cor. 5.4; $\varepsilon_* := \log_{13} 8 \approx 0.810714 < \log_7 5 \approx 0.827087 < 0.861353 = \log_5 4$):

Corollary 5.2 (Four-Register Powering; exponent improvement on [9, Corollary 4.3.1]; Lean `T3h`)**.** *Let K be any commutative ring and $N \geq 1$.*

(1) *Setting $\ell := \lfloor \log_{13} N \rfloor$, whenever D_8, $12!$, and $(\ell+1)!$ are invertible in K (in particular every field of characteristic 0, and every prime field $\mathbb{F}_p$ with $p > \max(12, \ell+1)$ and $p \nmid D_8$), there exists an explicit passive 4-register program cleanly computing x^N using strictly fewer than $\frac{104}{3} \cdot 8^\ell \leq \frac{8}{3} \cdot 13 \cdot N^{\log_{13} 8} = O(N^{0.81072})$ input accesses.*

(2) *Setting $\ell' := \lfloor \log_7 N \rfloor$, whenever $6!$ and $(\ell'+1)!$ are invertible in K (in particular characteristic 0 and every prime field $\mathbb{F}_p$ with $p > \max(6, \ell'+1)$; note $6! = 720$ invertible automatically implies $576 = 2^6 \cdot 3^2$ is invertible via $720^2 = 576 \cdot 900$), composing `prog5_x7` achieves strictly fewer than $\frac{70}{3} \cdot 5^{\ell'} \leq \frac{10}{3} \cdot 7 \cdot N^{\log_7 5} = O(N^{0.82709})$ accesses in 4 passive registers.*

The improvement over [9, Cor. 4.3.1] *($20N^{\log_5 4}$) is in the exponent only: the x^{13}-based bound is smaller for $N \gtrsim 5.2 \times 10^4$ (the x^7-based bound for $N \gtrsim 90$), and with 5 registers* [9, Cor. 4.3.1] *already achieves $\log_7 4 \approx 0.7124$.*

Proof. Work on four registers (O, U, R_1, R_2) where O is the passive output register of the main program and (U, R_1, R_2) act as work registers. Expand $N = \sum_{i=0}^{\ell} b_i \delta^i$ ($0 \leq b_i < \delta$, $(\delta, t) \in \{(13,8), (7,5)\}$). Crucially, because the inner subprograms C_i computing $U \leftarrow U + x^{b_i \delta^i}$ never touch O and do *not* require U to remain passive inside C_i, we apply clean non-passive composition (Composition Lemma of [1], register bound $\max\{r,s\} = 3$, whereas passive composition [9, Lemma 2.2] would incur $\max\{r, s+1\}$): starting from a 2-register forward-difference digit program for x^{b_i} using at most $b_i + 1 \leq \delta$ accesses (requiring $(\delta-1)!$ invertible), each recursive substitution of

the scalable base block (`prog8_x13` or `prog5_x7`) reuses the other two work registers catalytically and restores both identically, yielding clean subprograms C_i on (U, R_1, R_2) with $q_i \le \delta \cdot t^i$ accesses. Accumulating $\prod_{i=0}^{\ell} x^{b_i \delta^i} = x^N$ via the $(\ell+1)$-fold Cook–Mertz multivariate finite-difference recurrence $T_i = 2T_{i-1} + 2q_i$ on (O, U, R_1, R_2) (requiring $(\ell+1)!$ invertible, with O strictly passive) satisfies $(t-2)T_\ell + 2\delta \cdot 2^{\ell+1} \le 2\delta t \cdot t^\ell$, giving

$$T_\ell < \frac{2\delta t}{t-2}\, t^\ell \le \frac{2\delta t}{t-2}\, N^{\log_\delta t}$$

accesses in 4 passive registers ($\frac{104}{3} \cdot 8^\ell$ for $(13, 8)$ and $\frac{70}{3} \cdot 5^{\ell'}$ for $(7, 5)$). The entire construction and exact integer bound $3 \cdot \text{accesses} < 104 \cdot 8^\ell$ (resp. $3 \cdot \text{accesses} < 70 \cdot 5^{\ell'}$) are machine-checked over arbitrary commutative rings in `FourRegisterPowering.lean` (`T3h`). □

Corollary 5.3 (Catalytic Streaming Frequency Moments; paper-level via [9, Corollary 5.1.1])**.** *For every $D \ge 1$, the D-th frequency moment $F_D(\sigma) = \sum_{i=1}^{n} f_i^D$ of a stream $\sigma \in \{1, \dots, n\}^m$ can be computed exactly by a catalytic streaming algorithm using $O(D \log(m+1) + \log n + \log D)$ bits of regular workspace, $O(nD \log(m+1))$ bits of catalytic memory (matching the $r = 3$ space tier of* [9, Corollary 5.1.1]*), and at most $\frac{8}{3} \cdot 13 \cdot D^{\log_{13} 8} = O(D^{0.81072})$ passes over the stream (reducing the pass exponent from 0.86135 on the catalytic pass/space trade-off compared against the baselines of Kaplan et al. and Becker et al. in* [9, §5.1]*).*

Proof. Since the frequency vector $\mathbf{f} = (f_1, \dots, f_n)$ lives in the commutative $\mathbb{F}_p$-algebra $\mathbb{F}_p^n$ and D_8 has only six fixed prime factors (Table 2), choosing a prime $p > \max(12, m^D, \lfloor \log_{13} D \rfloor + 1)$ not dividing D_8 satisfies $\log p = O(D \log(m+1))$ and directly instantiates Corollary 5.2(1) over $\mathbb{F}_p^n$ (the coordinatewise powering bound and sum $\sum_{i=1}^{n} f_i^D$ are formally checked in `FourRegisterPowering.lean` as `T3i`; the stream/update model and bit-space accounting follow the paper-level reduction of [9, Theorem 5.1 and Corollary 5.1.1]). □

Corollary 5.4 (Non-Commutative Matrix Powering; sharpening [9, Theorem 5.2])**.** *Let $A \in M_m(\mathbb{F}_p)$ and $N \ge 1$.*

1. Zero-register-overhead $r = 3$ improvement via `prog5_x7` ($d = 2$): *For every prime $p > \max(7, \lfloor \log_7 N \rfloor + 1)$, lifting `prog5_x7` via the homogeneous linear form $\ell_A(z) = \sum_{i=1}^{m^2} a_i z_i$ and linear map μ_7 of* [9, §4.4, Theorem 5.2] *cleanly computes A^N using at most $\frac{10}{3} \cdot 7 \cdot N^{\log_7 5} = O(N^{0.82709})$ recursive matrix calls and the* exact *$r = 3$ scalar register count $2m^2 + \binom{m^2+2}{2} - 1 = 3m^2 + \binom{m^2+1}{2}$ of* [9, Theorem 5.2] *(strictly improving Vinciguerra's $O(N^{0.86136})$ with zero extra scalar registers).*
2. Degree-6 shear lift via `prog8_x13` ($d = 6$): *For every prime $p > \max(13, \lfloor \log_{13} N \rfloor + 1)$ with $p \nmid D_8$, lifting `prog8_x13` via μ_{13} cleanly computes A^N in at most $\frac{8}{3} \cdot 13 \cdot N^{\log_{13} 8} = O(N^{0.81072})$ recursive matrix calls using $3m^2 + \binom{m^2+5}{6} = \Theta(m^{12})$ scalar registers (reflecting the homogeneous degree-6 shear register v).*

Proof. In Vinciguerra's homogeneous lift [9, §4.4], setting $x \mapsto \ell_A(z)$ (linear in $b = m^2$ commuting variables $z_1, \dots, z_b$) maps u to m^2 scalar registers and v ($v \leftarrow v \pm Q_k u^d$) to $\binom{m^2+d-1}{d}$ scalars holding a degree-d homogeneous polynomial. Since each $F_k(u, v)$ is weighted-homogeneous of degree $N_0 \in \{7, 13\}$ ($\mathrm{wt}(u) = 1, \mathrm{wt}(v) = d$), applying μ_{N_0} (well-defined for $p > N_0$) yields A^{N_0} cleanly, and recursive base-N_0 composition uses $3m^2 + \binom{m^2+d-1}{d}$ scalars total (including passive output O). At $d = 2$ (`prog5_x7`), $3m^2 + \binom{m^2+1}{2} = 2m^2 + \binom{m^2+2}{2} - 1$ matches [9, Thm. 5.2] at $r = 3$; at $d = 6$ (`prog8_x13`), the $\Theta(m^{12})$ scalar cost exceeds Vinciguerra's $r = 4$ lift ($\Theta(m^6)$ scalars, $O(N^{\log_7 4})$ calls), making `prog5_x7` the strict winner at $\Theta(m^4)$ scalars. □

Remark 5.5 (Why $r = 4$ triangular loops do not immediately beat $\log_7 4$)**.** Exact modular linear algebra over $\mathbb{F}_{10^9+7}$ (`exp2`, Tier-2) on the $r = 4$ triangular closed-loop ansatz ($d_1 = 2, d_2 = 4$) yields

$N = 9$ at $t = 5$ ($\log_9 5 \approx 0.73249$; $N = 10$ unsolvable) and $N = 11$ at $t = 6$ ($\log_{11} 6 \approx 0.74722$; $N = 12$ unsolvable), failing to beat Vinciguerra's $t = 4$ baseline $N = 7$ ($\log_7 4 \approx 0.71241$). Because degree grows as $2t - 1$ (slope 2) here, $\log_{2t-1} t$ rises for $t \geq 4$; non-triangular or multi-stage $r = 4$ loops reaching $N \geq 10$ ($t = 5$) or $N \geq 13$ ($t = 6$) remain open (Open Problem 8.2).

6. Theorem 1.4: Balanced Zero-Workspace Bilinear Product

In the multivariate bilinear catalytic setting (where every instruction is either $R_i \leftarrow R_i \pm x_\ell$ or $R_i \leftarrow R_i \pm R_a R_b$ with i, a, b pairwise distinct, and zero clean workspace is available), naive linear-chain composition of a 10-operation product gadget computes $x_0 \cdots x_{d-1}$ using d catalysts and $3 \cdot 4^{d-1} - 2$ operations. Replacing the linear chain by the classical four-step clean-product commutator in a balanced binary tree cuts operations from $O(4^d)$ to $O(d^2)$ and registers from d to $\log_2 d + 2$ (Figure 3). Ben-Or and Cleve [2] compute any polynomial-size formula over a ring in three registers in a neighboring instruction set and survey logarithmic-register constructions; no lower bound is proved for Theorem 1.4, and whether $O(1)$ registers suffice in our strictly distinct-operand bilinear model remains open.

Proof of Theorem 1.4. For $k = 0$ ($d = 1$), the single instruction $R_{\text{target}} \leftarrow R_{\text{target}} + s x_{\text{offset}}$ ($s \in \{+1, -1\}$) uses $T(0) = 1$ op. For $k + 1$ ($d = 2^{k+1}$), given distinct registers target, r_a, r_b and an auxiliary pool P of size k (total $k + 3$ registers), let $A = \prod_{i<2^k} x_{\text{offset}+i}$ and $B = \prod_{i<2^k} x_{\text{offset}+2^k+i}$. Execute the 8-stage balanced commutator:

(1) $R_{\text{target}} \leftarrow R_{\text{target}} + s R_{r_a} R_{r_b}$,
(2) $R_{r_a} \leftarrow R_{r_a} + A$ (using pool $\{r_b\} \cup P$),
(3) $R_{\text{target}} \leftarrow R_{\text{target}} - s R_{r_a} R_{r_b}$,
(4) $R_{r_b} \leftarrow R_{r_b} + B$ (using pool $\{r_a\} \cup P$),
(5) $R_{\text{target}} \leftarrow R_{\text{target}} + s R_{r_a} R_{r_b}$,
(6) $R_{r_a} \leftarrow R_{r_a} - A$ (using pool $\{r_b\} \cup P$),
(7) $R_{\text{target}} \leftarrow R_{\text{target}} - s R_{r_a} R_{r_b}$,
(8) $R_{r_b} \leftarrow R_{r_b} - B$ (using pool $\{r_a\} \cup P$).

By induction, every recursive call in stages (2), (4), (6), (8) restores all registers in its auxiliary pool identically (crucially preserving R_{r_b} during stages (2),(6) and R_{r_a} during stages (4),(8)). Over any commutative ring R, the net update to R_{target} telescopes as

$$s\big(c_a c_b - (c_a + A)c_b + (c_a + A)(c_b + B) - c_a(c_b + B)\big) = sAB,$$

while R_{r_a} returns to $c_a + A - A = c_a$ and R_{r_b} returns to $c_b + B - B = c_b$. The exact operation count satisfies $T(k + 1) = 4T(k) + 4$ with $T(0) = 1$, solving cleanly to $3T(k) + 4 = 7 \cdot 4^k = 7d^2$, i.e., $T(k) = (7d^2 - 4)/3$. □

7. Empirical Companion: Nonlinear Cancellation in Boolean XAGs vs. Catalytic Programs

Our catalytic improvements (Theorems 1.1–1.3) are driven by *high-degree nonlinear cancellation*: intermediate work-register monomials of degree exceeding the target degree cancel completely upon summing across shifts or Frobenius phases. It is therefore instructive to compare this phenomenon against the open question of Boyar and Find [3] and Morse's recent proof [8] that four-term binary polynomial multiplication has unrestricted Boolean multiplicative complexity $\text{MC}_\wedge(\text{Mul}_4) = 9$ (matching its quadratic-only bound, with no gain from higher-degree intermediate cancellation).

In `experiments/exp4_boolean_mc_companion.py` (Tier-2), we run an exact CaDiCaL 2.1.2 SAT audit on unrestricted multi-output XAG circuits (allowing arbitrary nonlinear feedback between AND gates), archive all 8/8 UNSAT DIMACS `.cnf` and ASCII `.lrat` certificates under `results/sat_certs/` (18,932 total verified LRAT steps replayed by `scripts/verify_lrat.py` with mutant-rejection tests), and cross-check the CNF encoder both exhaustively against an independent brute-force truth-table enumerator ($n = 2, 3$) and differentially against an independent direct-parity-cube CNF encoder ($n = 2, 3$ and $\mathbb{F}_4$ multiplication) in `tests/test_all.py`:

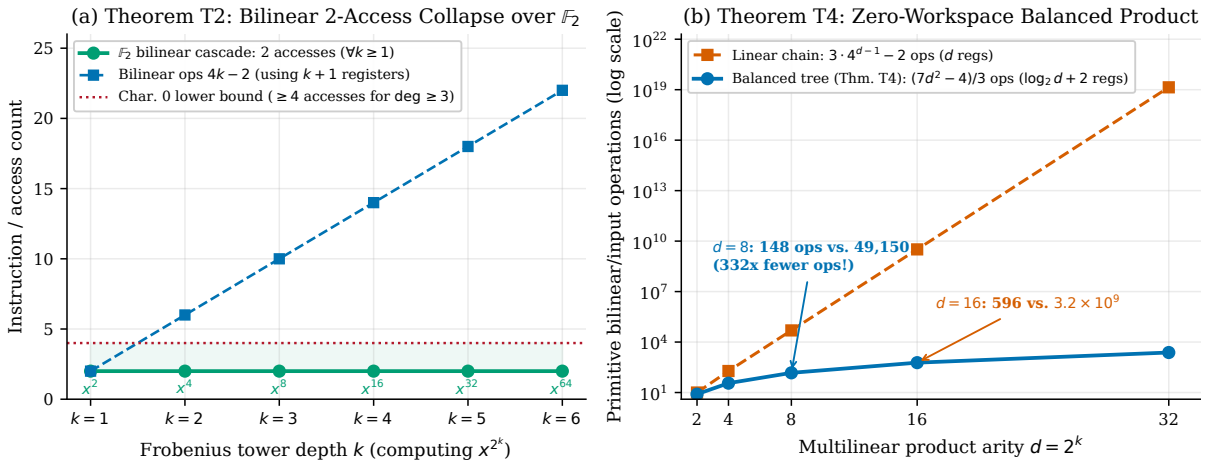


FIGURE 3. Theorems 1.2 and 1.4. (a) Strictly bilinear 2-access Frobenius collapse over $\mathbb{F}_2$: constant 2 input accesses and $4k-2$ bilinear ops for x^{2^k} $(k = 1, \ldots, 6)$, below the characteristic-0 lower bound of 4 accesses. (b) Log-scale comparison of our balanced zero-workspace product $(7d^2-4)/3$ ($\log_2 d + 2$ registers) against the linear chain $3 \cdot 4^{d-1} - 2$ (d registers).

- $\mathbb{F}_4$ multiplication $(n = 4, m = 2)$ is certified $k = 2$ UNSAT and $k = 3$ SAT ($\mathrm{MC}_\wedge = 3$, matching its quadratic Karatsuba bound);
- an orthogonal pair of bent quadratic forms on $n = 4$ is certified $k = 2$ UNSAT and $k = 3$ SAT ($\mathrm{MC}_\wedge = 3$);
- across 182 sampled 3-AND nonlinear-feedback circuits exhibiting genuine degree-3 cancellation down to degree 2, and 6 hard random $(n = 4, m = 3)$ quadratic maps (all certified $k = 2$ UNSAT), we observe **zero** counterexamples where nonlinear cancellation improves upon quadratic-only multiplicative complexity.

This sharp empirical contrast highlights a structural asymmetry: whereas Boolean XAG circuits exhibit strong cancellation rigidity in small regimes [3, 8], passive catalytic register programs exploit nonlinear cancellation aggressively — both via characteristic-p Frobenius collapse (Theorems 1.1–1.2) and via multi-call Pascal shear cancellation (Theorem 1.3).

8. Lean 4.35.0 Formalization, Statement Fidelity, and Open Problems

Every Tier-1 theorem (`T1`, `T1a`–`T1e`, `T1s`, `T1es`, `T2`, `T2b`, `T3a`–`T3i`, `T4`) is formalized from first principles in zero-Mathlib Lean 4.35.0 (`lean/Catalytic/*.lean`, `Challenge.lean`, `Solution.lean`) using Lean 4.35 `module` files and strict core linters (`autoImplicit=false`, `linter.unusedVariables=true`, `linter.deprecated=true`) across 14 core files (8,151 lines; longest `FourRegisterPowering.lean`, 1,364 lines) plus standalone Palomar files `PalomarChallenge.lean` (283 lines) and `PalomarSolution.lean`. To rule out vacuous hypotheses or degenerate zero-ring models, `NonVacuity.lean` exhibits concrete $0 \neq 1$ witnesses (`ModP 2/5/7/11/13`, $\mathrm{Fin}\,3 \to$ `ModP 13`, `GF4` with `GF4Hom.id`, and `Int`) for every hypothesis bundle of Theorems 1.1–1.4, Corollaries 5.2–5.3, and their soundness bridges (incl. $D_5, \ldots, D_9$ and factorial inverses). A repository scan confirms 0 occurrences of `sorry`, `admit`, `native_decide`, `implemented_by`, `unsafe`, custom `axiom`, or `@[extern]` across all proof and solution modules (Tables 5–6).

8.1. **Limitations, Threats to Validity, and Open Problems.** We highlight four threats to validity and four unresolved Tier-4 frontiers:

TABLE 5. Multi-kernel verification audit of the 22 exported Lean 4.35.0 declarations (5 distinct kernel implementations across 6 locally reproducible runs).

#	Kernel / Tool	Implementation	Verified Result
1	`lean v4.35.0 + VerifyAxioms`	C++ reference kernel	19/19 jobs, 0 warnings; axioms $\subseteq$ {propext, choice, Quot.sound}
2	`leanchecker`	C++ `.olean` replayer	Exit 0 on `Solution` & `PalomarSolution`
3	`leanchecker-paranoid`	C++ paranoid mode (#2)	Exit 0 on `Solution` & `PalomarSolution`
4	`con-leche --verified`	Verified Lean kernel	Accepted 3,544 (`Solution`) / 3,502 (`PalomarSolution`)
5	`con-ron --verified`	Verified Rust kernel	Accepted 3,544 (`Solution`) / 3,502 (`PalomarSolution`)
6	`nanoda_bin 0.4.19`	Independent Rust kernel	3,544 / 3,502 decls, 0 errors, strict 3-axiom check

TABLE 6. Statement-fidelity mapping between formal Lean 4.35 declarations in `Challenge.lean`, primary literature definitions, and concrete non-vacuity witnesses.

ID	Lean Declaration	Literature Anchor	Non-Vacuity ($0 \neq 1$)
T1	`InRPPass2`, `GF4`, `execRing`, `execAlg` (`T1`, `T1a`–`T1e`, `T1s`, `T1es`)	Vinciguerra [9], Def. 2.2, Thm. 3.5	`IsPrime 2,3,5,7`; `ModP 5` (char = 5); `GF4` (`GF4Hom.id`)
T2	`BilinInstr`, `DistinctBilinInstr` (`T2`, `T2b`)	Bilinear & distinct-operand Def. 2.2	`F2 = ModP 2`, $k = 2$, $x = 1$ cleanly evaluated
T3	`Pass3InstrR`, `Pass4Instr`, `sumFin` (`T3a`–`T3i`)	Vinciguerra [9], Def. 2.2, Cor. 4.3.1, 5.1.1; Comp. Lem. [1]	`ModP 5,7,11,13` ($D_5^{-1} \dots D_9^{-1}$, $12!^{-1}$, $6!^{-1}$); $\mathrm{Fin}\,3 \to$ `ModP 13` (`T3i`)
T4	`CatOp`, `usesOnly`, `balancedProdProg` (`T4`)	Zero-workspace distinct-index bilinear ops ($\leq k+2$ regs)	`Int` ($d = 4$, 36 ops, clean restoration)

- *Formal vs. functional model:* Definition 2.1 uses formal-polynomial cleanliness with $\mathbb{F}_p$ constants, strictly stronger over $\mathbb{F}_p$ than Vinciguerra's functional Def. 2.2 ($x^3 = x$ on $\mathbb{F}_2$); `T1s` certifies formal $\Rightarrow$ ring execution, while the converse is paper-level.
- *Bad-prime denominators (`T3b`–`T3d`, `T3g`):* `prog5_x7` avoids only $\{2, 3\}$, whereas $t = 6..9$ exclude additional prime divisors of D_t (Table 2); all hold unconditionally in char. 0.
- *SAT companion boundary:* All 8/8 Tier-2 CaDiCaL 2.1.2 UNSAT certificates (`results/sat_certs/`) are verified by `scripts/verify_lrat.py` alongside $n = 2, 3$ brute-force and second-encoder checks, though not yet replayed in `cake_lpr`; no Tier-1 theorem depends on SAT.
- *Zero-dependency linters & Windows `lean4lean`:* Keeping zero external Lake deps preserves a minimal 5-kernel trust base while precluding Batteries' `#lint`; built-in Lean 4.35 core linters pass with 0 warnings. Windows `lean4lean.exe` fails locally with an upstream `.olean` header mismatch and is omitted from Table 5.

Open Problem 8.1 (Closed-form 3-register frontier)**.** Does $D^{\mathrm{pass}}_{3,\mathbb{Q}}(t) = 2t - 3$ hold for all $t \geq 4$? Our constructions show $D^{\mathrm{pass}}_{3,\mathbb{Q}}(t) \geq 2t - 3$ for $t = 4..9$; uniform non-vanishing and upper bounds for $r \geq 3$ remain open.

Open Problem 8.2 (Breaking $\log_7 4$ at $r = 4$ registers)**.** Can a non-triangular or multi-stage 4-register program reach $D^{\mathrm{pass}}_{4,\mathbb{Q}}(5) \geq 10$ or $D^{\mathrm{pass}}_{4,\mathbb{Q}}(6) \geq 13$, beating $\log_7 4 \approx 0.71241$?

Open Problem 8.3 (Extension fields $\mathbb{F}_{p^e}$ $(e \geq 2)$)**.** Extend Theorem 1.1 to a closed-form criterion over arbitrary $\mathbb{F}_{p^e}$, generalizing the $\mathbb{F}_4$ spectrum in Remark 3.1.

Open Problem 8.4 (Open question of Boyar–Find and Mul_5)**.** Resolve whether nonlinear cancellation can beat quadratic-only multiplicative complexity for any quadratic Boolean map [3], and complete $n = 5$ [8].

AI-Assistance, CRediT, COI, Versioning, and Data/Code Availability. *AI-Contribution and CRediT:* Antigravity/Gemini assisted in exploratory probing, Lean 4.35 proof engineering, figures, and drafting under Alexander S. Donahue (CRediT: Conceptualization; Methodology; Formal analysis; Validation; Writing – review & editing; Supervision). *Not done by AI:* problem choice, theorem acceptance, statement-fidelity sign-off, and scope decisions. No AI is an author; Tier-1 soundness is certified by 5 Lean 4.35 kernels (6 runs, zero `sorryAx`/`_native`; Table 5).

COI, Export Control, & Versioning: Self-funded independent research (ORCID 0009-0006-8168-9648); no employer, grant, or COI; fundamental math (15 CFR 734.8). Release `v1.0.1` (Version DOI 10.5281/zenodo.24517199; Concept DOI 10.5281/zenodo.24517198): corrections are logged in `ERRATA.md`, and any falsified claim triggers a dated withdrawal notice on Zenodo/GitHub rather than silent deletion.

Data, Code, and Reproducibility Availability: All Lean 4.35.0 proofs (`lean/`, including Palomar contract files `PalomarChallenge.lean`, `palomar_comparator.json`, and `formalization.yaml`), Python exact-arithmetic and Hypothesis/mutant/SAT tests (`experiments/`, `tests/test_all.py`), deterministic figure scripts (`scripts/make_figures.py`), offline browser verifier (`docs/`), pinned lockfiles and SBOM (`requirements.lock.txt`, `sbom.cdx.json`), and cryptographic checksums (`MANIFEST.sha256`) are released openly under Apache-2.0 (code) and CC-BY-4.0 (prose and figures); no third-party PDFs [9, 8] are redistributed.

References

1. Yaroslav Alekseev, Yuval Filmus, Ian Mertz, Alexander Smal, and Antoine Vinciguerra, *Catalytic computing and register programs beyond log-depth*, 50th International Symposium on Mathematical Foundations of Computer Science (MFCS), LIPIcs, vol. 345, 2025, arXiv:2504.17412 [cs.CC], pp. 6:1–6:18.
2. Michael Ben-Or and Richard Cleve, *Computing algebraic formulas using a constant number of registers*, SIAM Journal on Computing **21** (1992), no. 1, 54–58.
3. Joan Boyar and Magnus Gausdal Find, *Multiplicative complexity of vector valued Boolean functions*, Theoretical Computer Science **720** (2018), 36–46.
4. Harry Buhrman, Richard Cleve, Michal Kouckỳ, Bruno Loff, and Florian Speelman, *Computing with a full memory: Catalytic space*, Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC), 2014, pp. 857–866.
5. James Cook and Ian Mertz, *Tree evaluation is in space $O(\log n \cdot \log \log n)$*, Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC), 2024, pp. 1268–1278.
6. Don Coppersmith and Edna Grossman, *Generators for certain alternating groups with applications to cryptography*, SIAM Journal on Applied Mathematics **29** (1975), no. 4, 624–627.
7. Édouard Lucas, *Théorie des fonctions numériques simplement périodiques*, American Journal of Mathematics **1** (1878), no. 2, 184–240.
8. Gregory Morse, *Unrestricted Boolean multiplicative complexity of four-term binary polynomial multiplication: Rational places, Hasse jets, and the failure of nonlinear feedback*, arXiv:2608.30238 [cs.CC], August 2026.
9. Antoine Vinciguerra, *An operator approach to register programs for catalytic computing*, arXiv:2609.18692 [cs.CC], September 2026.
10. R. Ryan Williams, *Simulating time with square-root space*, Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC), 2025, arXiv:2502.17779 [cs.CC], pp. 13–23.

Independent Researcher, Spokane, WA, USA
Email address: `0_0@0thernes.art`