MATHDISCOVERIES
Mathematical Preprint

Working Draft

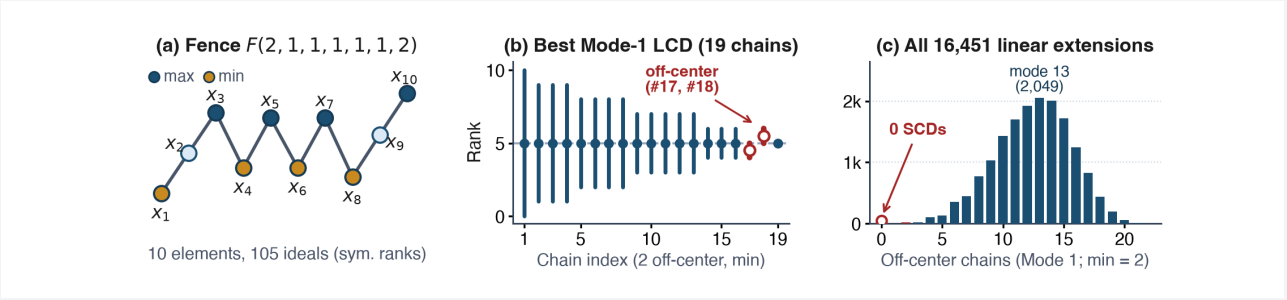
Fence Posets and Lexicographic Chain Decompositions: A 10-Element Counterexample and Nested $F(a,2,2,d)$ Chains

Tyler B. Hughes*,†

†Independent Researcher, United States

Supporting Information: `recompute.py`, Lean 4 library (`lean/FenceLexCounterexample/`), and trust ledgers

Disclosure: AI-assisted working draft directed by Tyler B. Hughes. Not for submission as is. No human mathematician has yet refereed the proof.



ABSTRACT: A fence poset $F(\alpha)$ turns a composition $\alpha = (\alpha_1, \ldots, \alpha_s)$ into a zigzag of $n = 1 + \sum_i \alpha_i$ elements whose ideal lattice $L(\alpha)$ has symmetric, bottom-, or top-interlacing rank sizes (Oğuz–Ravichandran, 2023). Following McConville, Sagan, and Smyth (2021), Elizalde and Sagan asked in Conjecture 4.6 (Question 4.6 in *Ann. Comb.*) whether every $F(\alpha)$ has a linear extension whose greedy lexicographic chain decomposition (LCD) is an SCD, BCD, or TCD matching this interlacing. We refute this at the 10-element fence $F(2,1,1,1,1,1,2)$ (105 ideals, symmetric ranks $1,4,8,13,17,19,17,13,8,4,1$): none of its 16,451 linear extensions yields an SCD, BCD, TCD, or NCD under either the printed sequence reading (Mode 1) or the 0/1-word reading (Mode 2). An explicit non-extension labeling yields a 19-chain Mode-1 SCD and NCD, leaving Conjecture 6.2 intact. Finally, for every $a \geq 1$ and $d \geq 1$, a repaired peel hook-grid proof shows $|L(a,2,2,d)| = 5ad + 7a + 7d + 10$ and constructs a nested BCD, settling Conjecture 1.5 for $F(a,2,2,d)$. All four results are formalized in Lean 4 + Mathlib (`lean/FenceLexCounterexample/`, 197 declarations, zero `sorry`, standard axioms only).

Verification Scope (G11): WARN: Human referee: pending
Main results: 5 of 6 in Lean Kernel (83%) · 1 Cert Only · 1 open (not counted)
Abstract claims: 4 of 4 in Lean Kernel (100%)

Full Trust Boundary: Section 5.

MSC 2020: 06A07, 06D05, 05A17, 68V15 **Keywords:** fence posets, distributive lattices, symmetric chain decompositions, lexicographic chain decompositions, Lean 4 formalization

1. INTRODUCTION AND MAIN RESULTS

A *fence poset* turns a composition $\alpha = (\alpha_1, \ldots, \alpha_s)$ of positive integers into a zigzag partial order: starting from x_1, walk up α_1 cover steps, down α_2 steps, up α_3 steps, and so forth across $n = 1 + \sum_{i=1}^{s} \alpha_i$ elements. Its lower order ideals (down-sets), ordered by inclusion, form a finite distributive lattice $L(\alpha)$ whose rank-generating polynomial enumerates ideals by cardinality. These polynomials arise in q-deformed rational numbers and continued fractions [4] [to check], where Morier-Genoud and Ovsienko conjectured that every fence rank sequence is unimodal.

McConville, Sagan, and Smyth [3] refined rank unimodality into a structural program: they conjectured (Conjecture 1.4 of [3]) that every rank sequence of $L(\alpha)$ is symmetric, bottom-interlacing, or top-interlacing according to a simple recursive rule on α, and proposed explaining this pattern bijectively by partitioning $L(\alpha)$ into saturated chains centered at or within half a rank of the middle (Conjecture 1.5 of [3]). Oğuz and Ravichandran [5] subsequently proved the interlacing property algebraically (restated as Theorem 1.2 of [1]), while the existence of centered chain decompositions (Conjecture 1.5, restated as Conjecture 4.5 of [1]) remains open in general.

To construct such decompositions algorithmically, McConville, Sagan, and Smyth [3] introduced a greedy *lexicographic chain decomposition* (LCD) driven by an arbitrary bijection (labeling) of $F(\alpha)$ and posed Conjecture 6.2: every fence admits some labeling whose LCD is nested (an NCD), verified in [3] for all fences with at most 7 elements. Elizalde and Sagan [1] specialized the labeling to a *linear extension* of $F(\alpha)$ and stated the following conjecture (Section 4.3 of arXiv:2201.03044v1; printed as Question 4.6 in the journal version [1], while the ensuing remark continues to refer to "this conjecture"):

> *Conjecture/Question 4.6* ([1]). For every composition β, is there a linear extension of $F(\beta)$ whose cor-

responding LCD is an SCD, BCD, or TCD of $L(\beta)$ consistent with Theorem 1.2?

They remarked that "so many linear extensions give a CD of the desired type that it is hard to find a common feature which runs through some subset of them."

Our main results answer this question in the negative at $n = 10$ under both natural readings of lexicographic order, clarify the survival of Conjecture 6.2 and the separation between readings, and settle Conjecture 1.5 unconditionally for the infinite two-parameter family $F(a, 2, 2, d)$.

Theorem 1 (Mode-1 Refutation of Conjecture/Question 4.6). *Let $F_{\text{head}} = F(2,1,1,1,1,1,2)$ be the 10-element rising-first fence $x_1 < x_2 < x_3 > x_4 < x_5 > x_6 < x_7 > x_8 < x_9 < x_{10}$. For every linear extension L of F_{head}, the Mode-1 (printed increasing-sequence) lexicographic chain decomposition $\mathrm{LCD}_1(F_{\text{head}}, L)$ fails to be an SCD, fails to be a BCD, fails to be a TCD, and fails to be an NCD.*

By Theorem 1.2 of [1] (proved in [5]), the ideal lattice $L(2,1,1,1,1,1,2)$ has symmetric rank sizes: 105 ideals with rank sequence $1, 4, 8, 13, 17, 19, 17, 13, 8, 4, 1$, and 16,451 linear extensions (certified in Lean 4 by `card_orderIdeal_headFence`, `rankCount_headFence`, and `card_linearExtension_headFence`). Thus Theorem 1 refutes Conjecture 4.6 of [1] (and answers Question 4.6 in the journal version in the negative). Because an alternative convention compares ideals by their 0/1 characteristic words with $0 < 1$ (Mode 2), we verify that the counterexample does not depend on which reading is adopted.

Theorem 2 (Mode-2 Refutation at $F(2,1,1,1,1,1,2)$). *For every linear extension L of $F_{\text{head}} = F(2,1,1,1,1,1,2)$, the Mode-2 (genuine 0/1-word, $0 < 1$) lexicographic chain decomposition $\mathrm{LCD}_2(F_{\text{head}}, L)$ fails to be an SCD, fails to be a BCD, fails to be a TCD, and fails to be an NCD.*

The obstruction is strictly tied to the linear-extension requirement of [1]: dropping order-preservation restores a centered lexicographic decomposition immediately, so F_{head} does not refute Conjecture 6.2 of [3].

Proposition 3 (Conjecture 6.2 Survival Witness). *For the explicit non-linear-extension labeling*

$$L_{\text{wit}} = (x_1, x_2, x_3, x_5, x_7, x_8, x_4, x_9, x_{10}, x_6)$$

of F_{head} (violating $x_4 < x_3$), $\mathrm{LCD}_1(F_{\text{head}}, L_{\text{wit}})$ is both an SCD and an NCD consisting of 19 chains.

Investigating four-part fences that also defeat Mode-1 linear extensions (starting at $F(3,2,2,3)$) led to an explicit structural decomposition that bypasses the lexicographic recipe altogether. After repairing a coverage gap at $d \geq a+2$ in an initial peel scheme, we obtain an unconditional nested BCD for every four-part fence $F(a,2,2,d)$.

Theorem 4 (All-(a,d) Repaired Peel Nested-BCD Theorem). *For all integers $a \geq 1$ and $d \geq 1$, the distributive ideal lattice $L(a,2,2,d)$ of the four-part fence $F(a,2,2,d)$ admits a chain decomposition that is both bottom-centered (BCD) and nested (NCD). In particular, Conjecture 1.5 of [3] holds unconditionally for every $F(a,2,2,d)$ $(a,d \geq 1)$, including every symmetric-end fence $F(k,2,2,k)$ $(k \geq 1)$.*

Figure 1 summarizes the discovery pipeline, proof methods, and verification tiers separating pure-kernel Lean 4 proofs, exact computation, and open problems.

The snippet below checks the 105 ideals and symmetric rank sizes of $F(2,1,1,1,1,1,2)$ directly in standard Python (replayed verbatim in `recompute.py`).

Instant Witness Check: $F(2,1,1,1,1,1,2)$ Ideals & Ranks

```
below = [[],[0],[1,3],[],[3,5],[],[5,7],[],[7],[8]]
ideals = [m for m in range(1 << 10) if all(
    not (m >> e & 1) or all(m >> b & 1 for b in below[e])
    for e in range(10))]
ranks = [sum(m.bit_count() == r for m in ideals)
         for r in range(11)]
assert len(ideals) == 105
assert ranks == [1,4,8,13,17,19,17,13,8,4,1]
```

The rest of the paper is organized as follows. Section 2 fixes exact definitions matching our Lean 4 library verbatim, proves the Mode-1/Mode-2 equal-rank reversal lemma and the symmetric peak collapse lemma, and works out $F(1,1,1)$ by hand. Section 3 proves Theorems 1, 2, and 4, Proposition 3, and the Mode-1/Mode-2 separation at $F(3,2,2,3)$ (Proposition 7). Section 4 presents the computational census through $n = 11$, off-center failure diagnostics (Proposition 8), and Conjecture 9. Section 5 presents the machine-generated Lean 4 Reader's Trust Boundary, Section 6 reviews prior work and open questions, and Section 7 documents reproducibility and AI disclosure.

2. DEFINITIONS, NOTATION, AND GENERAL LEMMAS

We follow Elizalde and Sagan [1] and McConville, Sagan, and Smyth [3], matching the modules `Basic.lean` and `LexChainDecomp.lean` of our Lean 4 library verbatim.

Fence posets and lattices. A *composition* $\alpha = (\alpha_1, \ldots, \alpha_s)$ is a nonempty list of positive integers. The rising-first *fence poset* $F(\alpha)$ has $n = 1 + \sum_{i=1}^{s} \alpha_i$ elements $x_1, \ldots, x_n$ (0-indexed in Lean via `Fin (α.sum + 1)`). For each $0 \leq k < n-1$, the adjacent cover is $x_{k+1} < x_{k+2}$ when k falls in an odd-indexed run α_{2j-1} and $x_{k+1} > x_{k+2}$ when k falls in an even-indexed run α_{2j}; the order $\leq$ is the reflexive-transitive closure of these covers (`fenceLe_iff_reflTransGen`).

A *lower order ideal* (down-set) of a finite poset P is a subset $I \subseteq P$ such that $y \in I$ and $x \leq y$ imply $x \in I$. Ordered by inclusion, the ideals form a finite distributive lattice $L(P)$ (`OrderIdeal P`), graded by cardinality $\mathrm{rk}(I) = |I| \in \{0, \ldots, |P|\}$. Write $r_k(P) = |\{I \in L(P) : \mathrm{rk}(I) = k\}|$ for the k-th rank size. A sequence $(b_0, \ldots, b_n)$ is *symmetric* if $b_k = b_{n-k}$ for all $0 \leq k \leq n$, *top-interlacing* if $b_0 \leq b_n \leq b_1 \leq b_{n-1} \leq \cdots$, and *bottom-interlacing* if $b_n \leq b_0 \leq b_{n-1} \leq b_1 \leq \cdots$. By Theorem 1.2 of [1] (proved in [5]), the rank sequence of $L(\alpha)$ is symmetric if $s = 1$; bottom-interlacing if s is even or $(s \geq 3$

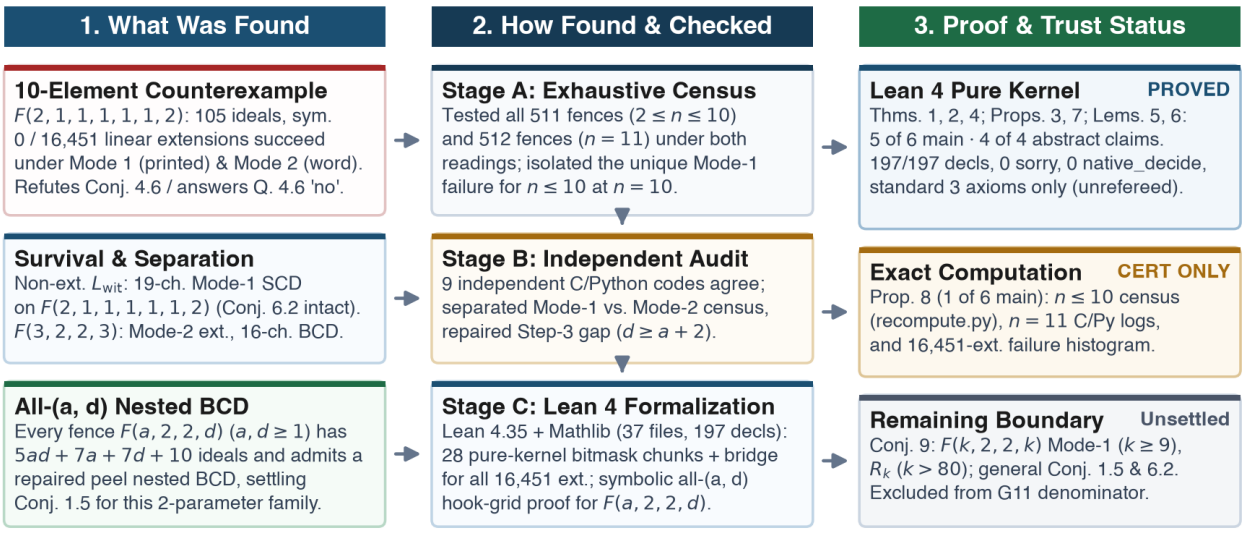


Figure 1. Discovery and proof architecture for `fence-lattice-chain-decompositions`. Column 1 summarizes the counterexample, survival/separation witnesses, and all-(a,d) nested-BCD theorem. Column 2 traces the census ($n \le 11$), independent C/Python audit, and Lean 4 + Mathlib formalization (197 audited declarations). Column 3 separates Lean 4 pure-kernel proofs (5 of 6 main results; 4 of 4 abstract claims), exact computation (1 of 6 main results), and unsettled conjectures (excluded from the proved denominator).

is odd and $\alpha_1 > \alpha_s$); top-interlacing if $s \ge 3$ is odd and $\alpha_1 < \alpha_s$; and when $s \ge 3$ is odd with $\alpha_1 = \alpha_s$, it is symmetric, bottom-interlacing, or top-interlacing according as the inner composition $\alpha' = (\alpha_2, \dots, \alpha_{s-1})$ is symmetric, top-interlacing, or bottom-interlacing.

Chain decompositions. An ideal J *covers* I in $L(P)$ ($I \lessdot J$, `Covers P I J`) if $I \subseteq J$ and $|J| = |I| + 1$. A *saturated chain* $C = (I_0 \lessdot I_1 \lessdot \cdots \lessdot I_\ell)$ is a nonempty list of ideals with consecutive covers; write $u(C) = \mathrm{rk}(I_0)$ and $v(C) = \mathrm{rk}(I_\ell)$ for its bottom and top ranks. A *chain decomposition* (CD) of $L(P)$ is a partition of $L(P)$ into pairwise disjoint saturated chains (`ChainDecomp P`). Let $n = |P|$. A CD $\mathcal{C}$ is:

- *symmetric* (SCD, `IsSCD`) if $u(C) + v(C) = n$ for all $C \in \mathcal{C}$;
- *bottom-centered* (BCD, `IsBCD`) if $u(C) + v(C) = n$ or $u(C) + v(C) + 1 = n$ for all $C \in \mathcal{C}$;
- *top-centered* (TCD, `IsTCD`) if $u(C) + v(C) = n$ or $u(C) + v(C) = n + 1$ for all $C \in \mathcal{C}$;
- *nested* (NCD, `IsNCD`) if for every $C_1, C_2 \in \mathcal{C}$, either $[u(C_1), v(C_1)] \subseteq [u(C_2), v(C_2)]$ or $[u(C_2), v(C_2)] \subseteq [u(C_1), v(C_1)]$.

Clearly every SCD is a BCD and a TCD, and every BCD or TCD is an NCD (since integer intervals whose endpoint sums differ by at most 1 must nest).

Linear extensions and both lexicographic readings. A *labeling* of P ($|P| = n$) is a bijection $L : P \to \{0, \dots, n-1\}$ (equivalently, the ordering $e_0, \dots, e_{n-1}$ with $e_p = L^{-1}(p)$). It is a *linear extension* (`IsLinearExtension P L`) if $x \le_P y$ implies $L(x) \le L(y)$.

Given any labeling L, two natural total orders arise on subsets $s \subseteq P$:

- **Mode 1 (printed increasing-sequence reading, `lexIncLT`):** write s as the strictly increasing list of its labels $\{L(x) : x \in s\}$ and compare lists lexicographically. This matches [1] Section 4.3 verbatim (for their $F(2,4,1)$ example with extension $x_7, x_8, x_6, x_5, x_4, x_1, x_2, x_3$, the ideal $\{x_1, x_6, x_7, x_8\}$ writes as x_7, x_8, x_6, x_1).
- **Mode 2 (genuine 0/1-word reading, `lexWordLT`):** associate to s the length-n binary word $w_L(s) \in \{0,1\}^n$ with $w_L(s)_p = 1$ if $L^{-1}(p) \in s$ and 0 otherwise, and compare words lexicographically with $0 < 1$.

In either mode $\mu \in \{1, 2\}$, the greedy *lexicographic chain decomposition* $\mathrm{LCD}_\mu(P, L)$ (`lcd`) repeatedly chooses the μ-lexicographically smallest unused ideal I_0 of minimum remaining rank, extends greedily by taking at each step the μ-lexicographically smallest unused upper cover I_{j+1} of I_j until no unused upper cover exists, removes the resulting chain, and repeats until all ideals are covered.

Lemma 5 (Mode-1 and Mode-2 Equal-Rank Reversal). *For any bijection $L : \mathrm{Fin}\, n \simeq \mathrm{Fin}\, n$ and any two subsets $s, t \subseteq \mathrm{Fin}\, n$ with $|s| = |t|$,*

$$s <^L_{\mathrm{inc}} t \iff t <^L_{\mathrm{word}} s.$$

Proof. Assign to each subset s the integer key $K_L(s) = \sum_{x \in s} 2^{n-1-L(x)}$, the base-2 value of $w_L(s)$. Since all words have length n, $s <^L_{\mathrm{word}} t$ holds if and only if $K_L(s) < K_L(t)$. For two distinct subsets s, t of equal cardinality $|s| = |t|$, neither increasing label sequence is a prefix of the other, so their first differing position occurs at the smallest label $p \in \{0, \dots, n-1\}$ where $L^{-1}(p) \in s \triangle t$. The subset containing $L^{-1}(p)$ has the smaller label at that first difference (hence is smaller under $<^L_{\mathrm{inc}}$), while its 0/1 word has $1 > 0$ at bit p (hence has strictly larger binary key K_L). Therefore $s <^L_{\mathrm{inc}} t$ if and only if $K_L(s) > K_L(t)$, which is equivalent to $t <^L_{\mathrm{word}} s$. □

Because every selection in LCD compares ideals of identical rank, Lemma 5 shows that Mode 1 picks the maximum binary key at every step and Mode 2 picks the minimum binary key. In particular, extending a chain from I to $I \cup \{x\}$ chooses the unused cover with smallest $L(x)$ in Mode 1 and largest $L(x)$ in Mode 2.

Lemma 6 (Symmetric Peak Collapse). *Let P be a finite poset with $|P| = 2m$ whose ideal lattice $L(P)$ has symmetric rank sizes $r_k = r_{2m-k}$ and a strict unique maximum at the middle rank m ($r_k < r_m$ for all $k \neq m$). Then a chain decomposition $\mathcal{C}$ of $L(P)$ is an NCD if and only if it is an SCD. Moreover, for any finite poset with symmetric rank sizes, $\mathcal{C}$ is a BCD if and only if it is an SCD, and a TCD if and only if it is an SCD.*

Proof. For the BCD/TCD collapse on any symmetric-rank poset ($|P| = n$), define the centered rank sum of $L(P)$ as $\Sigma = \sum_{I \in L(P)} (2\,\mathrm{rk}(I) - n) \in \mathbb{Z}$. Symmetry $r_k = r_{n-k}$ forces $\Sigma = 0$. On the other hand, each saturated chain $C \in \mathcal{C}$ of length $\ell = v(C) - u(C) + 1 \geq 1$ contributes $\ell \cdot (u(C) + v(C) - n)$ to Σ. In a BCD every summand is ≤ 0, so $\Sigma = 0$ forces $u(C) + v(C) = n$ for every $C \in \mathcal{C}$ (an SCD); TCD is identical with ≥ 0.

For the NCD collapse at $|P| = 2m$ with unique strict peak r_m, every chain $C \in \mathcal{C}$ must intersect rank m: otherwise let $C_0 \in \mathcal{C}$ be a chain of minimum length $v - u$, which by nesting lies inside every chain's rank interval $[u(C), v(C)]$; if $m \notin [u(C_0), v(C_0)]$, then every chain intersecting rank m also intersects rank $u(C_0)$, giving $r_{u(C_0)} \geq r_m$, contradicting $r_{u(C_0)} < r_m$. Thus every chain spans rank m, so $|\mathcal{C}| = r_m$, and for each $0 \leq k \leq 2m$, r_k equals the number of chains containing rank k. By induction on $j = 0, \ldots, m-1$, suppose every chain with $u(C) < j$ or $v(C) > 2m - j$ is symmetric; nesting implies that either all chains reaching rank j also reach $2m - j$ or vice versa, and equality $r_j = r_{2m-j}$ forces both sets to coincide. Hence $u(C) + v(C) = 2m$ for all $C \in \mathcal{C}$. □

Worked example: $F(1,1,1)$. For $\alpha = (1,1,1)$, $F(1,1,1)$ is $x_1 < x_2 > x_3 < x_4$ (4 elements). Its 8 ideals are $\emptyset$ (rank 0), $\{x_1\}, \{x_3\}$ (rank 1), $\{x_1, x_3\}, \{x_3, x_4\}$ (rank 2), $\{x_1, x_2, x_3\}, \{x_1, x_3, x_4\}$ (rank 3), and $\{x_1, x_2, x_3, x_4\}$ (rank 4), with symmetric rank sizes $1, 2, 2, 2, 1$. Of its 5 linear extensions, 2 yield a Mode-1 SCD (x_1, x_3, x_2, x_4 and x_3, x_4, x_1, x_2) and 3 fail, whereas 4 yield a Mode-2 SCD (Figure 2(a)–(b)).

3. PROOFS OF THE MAIN RESULTS

Pure-Kernel Refutation at $F(2,1,1,1,1,1,2)$.

Proof of Theorems 1 and 2, and Proposition 3. Consider the 10-element fence $F_{\mathrm{head}} = F(2,1,1,1,1,1,2)$ (0-indexed $0, \ldots, 9$ in Lean). Encoding each subset $s \subseteq \{0, \ldots, 9\}$ by its 10-bit mask $m(s) = \sum_{e \in s} 2^e < 1{,}024$, a mask encodes a lower order ideal of F_{head} if and only if it satisfies the 9 cover implications (`isLower_headFence_iff_mask`). Direct enumeration inside Lean's kernel establishes $|L(F_{\mathrm{head}})| = 105$ (`card_orderIdeal_headFence`) with symmetric unimodal rank sequence

$$(r_0, \ldots, r_{10}) = (1, 4, 8, 13, 17, 19, 17, 13, 8, 4, 1)$$

(`rankCount_headFence`, `hasSymmetricRanks_headFence`), with a unique strict peak $r_5 = 19$ at rank $m = 5$ (`headFence_unique_peak`). By Lemma 6, for every chain decomposition $\mathcal{C}$ of $L(F_{\mathrm{head}})$, the four properties SCD, BCD, TCD, and NCD are pairwise equivalent (`headFence_equiv_cd_classes`). Furthermore, certifying the dynamic-programming recurrence $N(I) = \sum_{e \in \max(I)} N(I \setminus \{e\})$ over all 1,024 bitmasks proves that F_{head} has exactly 16,451 linear extensions (`card_linearExtension_headFence`).

To prove Theorems 1 and 2 without trusting external code or compiler execution, we partition all 16,451 linear extensions of F_{head} by their first two elements (e_0, e_1). Because the minimal elements of F_{head} are $\{0, 3, 5, 7\}$, the valid length-2 prefixes form exactly 14 pairs:

$$(0,1),\ (0,3),\ (0,5),\ (0,7),\ (3,0),\ (3,5),\ (3,7),$$
$$(5,0),\ (5,3),\ (5,7),\ (7,0),\ (7,3),\ (7,5),\ (7,8).$$

In the zero-Mathlib kernel module `HeadKernel.lean`, recursive bitmask evaluators `checkAllM1` and `checkAllM2` traverse every valid linear extension extending a prefix, simulate the greedy LCD step-by-step using the equal-rank binary key ordering of Lemma 5 (pre-sorted rank buckets and shrinking unused-cover scan), and return `true` on a branch only when an off-center chain ($u + v \neq 10$) is encountered before exhausting a fixed fuel budget (16 chains; running out of fuel or buckets returns `false`, so false positives are impossible).

All 28 prefix chunks (14 for Mode 1 and 14 for Mode 2, in `HeadChunks/M1_01..14.lean` and `M2_01..14.lean`) are proved inside Lean 4's kernel via `decide +kernel` and combined by exhaustive case analysis over all 10 elements into `headAllM1` and `headAllM2` (`HeadChunks.lean`). Soundness theorems in `HeadBridge.lean` (`refutation_mode1_of_checkAllM1`, `refutation_mode2_of_checkAllM2`) prove by induction on depth that every linear extension $L : \mathrm{Fin}\,10 \simeq \mathrm{Fin}\,10$ enters an accepted branch and that every bitmask step matches `OrderIdeal`, `pickLexMin`, `extendChain`, and `lcd`. Together with `headFence_equiv_cd_classes`, this yields $\neg\mathrm{IsSCD} \wedge \neg\mathrm{IsBCD} \wedge \neg\mathrm{IsTCD} \wedge \neg\mathrm{IsNCD}$ for every linear extension L under both Mode 1 (`headFence_refutation_mode1`) and Mode 2 (`headFence_refutation_mode2`).

To make the obstruction concrete, consider the optimal Mode-1 linear extension $L_{\mathrm{best}} = (x_1, x_2, x_4, x_3, x_8, x_6, x_7, x_5, x_9, x_{10})$, one of the 22 extensions achieving the minimum of 2 off-center chains (Hero Figure, Panel (b)). Its Mode-1 LCD builds 19 chains, of which chains 1–16 are centered on rank 5. Chain 17 begins at the least unused rank-4 ideal $\{x_6, x_8, x_9, x_{10}\}$ and climbs to $\{x_1, x_6, x_8, x_9, x_{10}\}$ at rank 5, where it must terminate because all three rank-6 upper covers ($\{x_1, x_2, x_6, x_8, x_9, x_{10}\}$, $\{x_1, x_4, x_6, x_8, x_9, x_{10}\}$, and $\{x_1, x_6, x_7, x_8, x_9, x_{10}\}$) were already consumed by chains 12, 15, and 16. Consequently chain 17 has span $[4, 5]$ (sum 9), stranding $\{x_4, x_5, x_6, x_8, x_9\} \lessdot \{x_4, x_5, x_6, x_8, x_9, x_{10}\}$ as chain 18 with span $[5, 6]$ (sum 11).

Finally, for Proposition 3, take the bijection L_{wit} from Proposition 3 (`headWitnessLabeling`). Since $x_4 < x_3$ in F_{head} while x_3 precedes x_4 in L_{wit}, L_{wit} is not a linear extension (`headWitness_not_linearExtension`). Running the certified full-lattice evaluator `runFullLCD_sound` with fuel $105 = |L(F_{\mathrm{head}})|$ proves inside Lean's kernel that $\mathrm{LCD}_1(F_{\mathrm{head}}, L_{\mathrm{wit}})$ consists of 19 chains, all satisfying $u + v = 10$, hence is both an SCD and an NCD (`headWitness_mode1_scd_ncd`). □

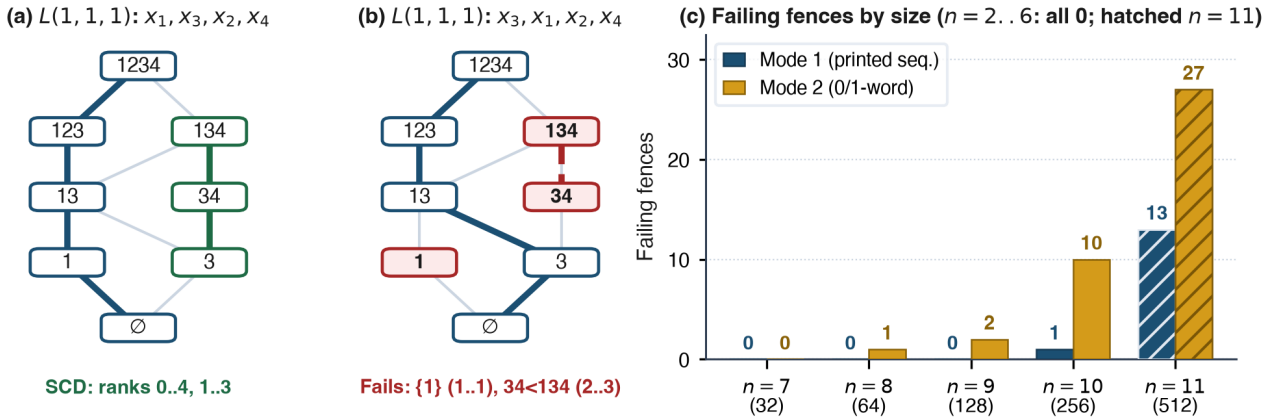


Figure 2. Worked example $L(1,1,1)$ and exhaustive failure census through $n = 11$. Panels (a)–(b): Hasse diagrams of $L(1,1,1)$ showing Mode-1 LCD chains for x_1, x_3, x_2, x_4 (SCD: ranks 0..4 and 1..3) versus x_3, x_1, x_2, x_4 (fails: stranded singleton $\{x_1\}$ at rank 1 and off-center chain $\{x_3, x_4\} \lessdot \{x_1, x_3, x_4\}$ at ranks 2..3). Panel (c): Number of failing fences by element count n (total compositions in parentheses; $n = 2 \ldots 6$ have 0 failures in both modes; $n \le 10$ verified in `recompute.py`; hatched $n = 11$ counts from independent C/Python census logs).

Mode-1 vs. Mode-2 Separation at $F(3,2,2,3)$. Although $F(2,1,1,1,1,1,2)$ fails under both readings, the two readings diverge sharply in general. The smallest four-part Mode-1 failure is $F(3,2,2,3)$ ($n = 11$, 97 ideals, bottom-interlacing rank sizes 1, 3, 7, 12, 15, 16, 15, 12, 8, 5, 2, 1), which cleanly separates Mode 1 from Mode 2.

Proposition 7 (Mode-1 vs. Mode-2 Separation at $F(3,2,2,3)$). *For the* 11*-element fence* $F(3,2,2,3)$ (97 *ideals*), *the explicit linear extension*

$$L_{\text{sep}} = (x_1, x_2, x_3, x_6, x_{11}, x_5, x_4, x_{10}, x_7, x_9, x_8)$$

produces a Mode-2 decomposition $\mathrm{LCD}_2(F(3,2,2,3), L_{\text{sep}})$ *that is both a BCD and an NCD consisting of* 16 *chains.*

Proof. Direct kernel checking of the 10 cover inequalities confirms that L_{sep} (`f3223Mode2Labeling`) is a linear extension of $F(3,2,2,3)$ (`f3223Mode2_isLinearExtension`), and checking all 2,048 bitmasks proves $|L(F(3,2,2,3))| = 97$ (`card_orderIdeal_f3223`). Applying `runFullLCD_sound` with fuel 97 proves inside Lean's kernel that $\mathrm{LCD}_2(F(3,2,2,3), L_{\text{sep}})$ has 16 chains, each with $u + v \in \{10, 11\}$, hence is a BCD and an NCD (`f3223Mode2_bcd_ncd`). □

By contrast, exhaustive enumeration over all 12,000 linear extensions of $F(3,2,2,3)$ in `recompute.py` finds 0 Mode-1 BCDs and 598 Mode-2 BCDs. In Lean we also certify (`f3223R3_not_linearExtension`, `f3223R3_mode1_bcd_ncd`) that the non-linear-extension labeling $R_3 = (x_1, x_2, x_3, x_4, x_6, x_8, x_{10}, x_{11}, x_5, x_9, x_7)$ produces a 16-chain Mode-1 nested BCD.

All-(a,d) Repaired Peel Nested-BCD Theorem. We now prove Theorem 4 symbolically for every $a \ge 1$ and $d \ge 1$, following `PeelBCDAllAD.lean` in `lean/FenceLexCounterexample/` (Figure 3).

Proof of Theorem 4. Let $a, d \ge 1$ and $n = a + d + 5 = |F(a,2,2,d)|$. Write the elements of $F(a,2,2,d)$ from left to right as

$$p_0 < p_1 < \cdots < p_a > M_1 > V < M_2 < Q > e_1 > \cdots > e_d.$$

Here $p_0, \ldots, p_{a-1}$ form the left rising prefix below the peak p_a, $\{M_1, V, M_2, Q\}$ is the 4-element middle core satisfying $M_1 > V < M_2 < Q$, and $e_1 > \cdots > e_d$ is the right falling tail with minimal element e_d.

Step 1 (Peel coordinate bijection and ideal count). Let $F' = \{M_1, V, M_2, Q, e_1, \ldots, e_d\}$ be the right sub-fence and set $W = L(F')$. The middle core $\{M_1, V, M_2, Q\}$ has exactly 7 lower ideals:

$$\emptyset,\ \{V\},\ \{V, M_1\},\ \{V, M_2\},$$
$$\{V, M_1, M_2\},\ \{V, M_2, Q\},\ \{V, M_1, M_2, Q\}.$$

Every ideal $w \in W$ is uniquely specified by a pair (m, c) where $m \in \{0, \ldots, d\}$ counts the bottom m tail elements $\{e_{d-m+1}, \ldots, e_d\}$ and c is one of the 7 core ideals, subject only to the constraint that $Q \in c$ forces $m = d$ (since $Q > e_1 > \cdots > e_d$). Thus 5 core ideals (those without Q) pair with all $d + 1$ tail values and 2 core ideals (those containing Q) pair with $m = d$, giving $|W| = 5(d+1) + 2 = 5d + 7$.

Let $W_1 = \{w \in W : M_1 \in w\}$ ($|W_1| = 2(d+1)+1 = 2d+3$). An ideal $I \in L(F(a,2,2,d))$ restricts on F' to some $w \in W$ and on the left chain $\{p_0, \ldots, p_a\}$ to a prefix $\{p_0, \ldots, p_{a'-1}\}$ of size a', where p_a may enter I if and only if $M_1 \in w$ (since $p_a > p_{a-1}$ and $p_a > M_1$). Therefore $I \mapsto (a', w)$ is a bijection (`PeelCoord.peelEquiv`) between $L(F(a,2,2,d))$ and

$$\mathrm{PeelCoord}(a,d) = \{(a', w) : w \in W,\ 0 \le a' \le a + \mathbf{1}_{w \in W_1}\},$$

with $\mathrm{rk}(a', w) = a' + \mathrm{rk}(w)$. Summing fiber sizes yields the exact ideal count (`card_orderIdeal_F_a22d`)

$$\begin{aligned} |L(F(a,2,2,d))| &= (a+1)(|W| - |W_1|) + (a+2)|W_1| \\ &= (a+1)(3d+4) + (a+2)(2d+3) \\ &= 5ad + 7a + 7d + 10. \end{aligned}$$

Step 2 (Repaired hook-grid decomposition). Let $c = (c_0 \lessdot c_1 \lessdot \cdots \lessdot c_q)$ be a saturated chain in W (so $c_t \in W_1$ is monotone in t). Set the fiber height $h(t) = a + \mathbf{1}_{c_t \in W_1}$ and the *base height* $H = h(0) = a + \mathbf{1}_{c_0 \in W_1}$. For each hook index $0 \le i \le \min(H, q)$, define the saturated hook chain $H_i(c)$

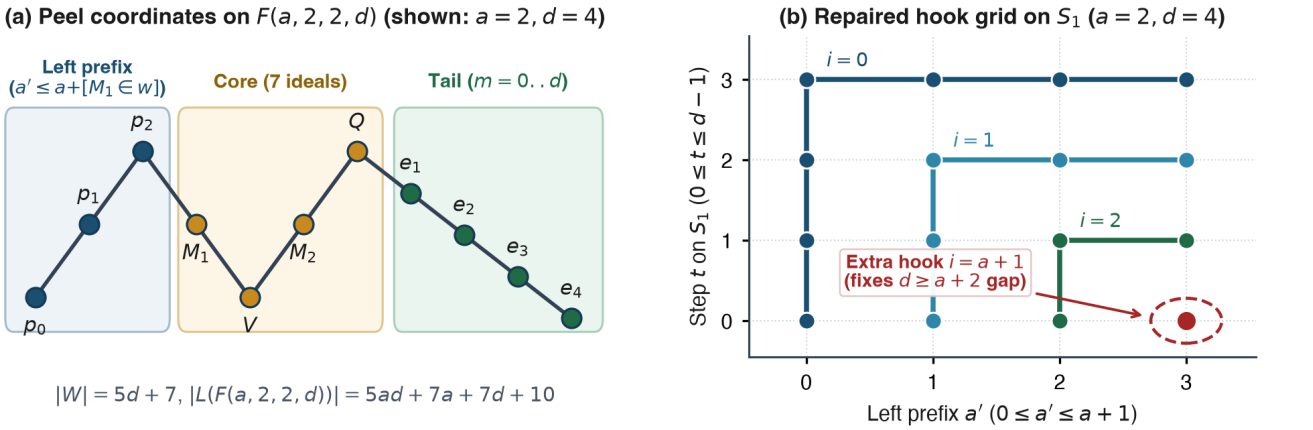


Figure 3. Repaired peel hook-grid decomposition for $F(a,2,2,d)$. Panel (a): Decomposition of $F(2,2,2,4)$ into the left rising prefix ($p_0 < p_1 < p_2$), the 4-element middle core $\{M_1, V, M_2, Q\}$ (7 ideals), and the right falling tail ($e_1 > e_2 > e_3 > e_4$), yielding $|L(F(a,2,2,d))| = 5ad+7a+7d+10$. Panel (b): Hook grid (a',t) over chain S_1 at $(a,d) = (2,4)$ ($q = d-1 = 3$, height $a+1 = 3$). Unrepaired hooks $i = 0,1,2$ miss $(3,0) = (a+1, (0,\{V,M_1\}))$; setting base height $H = a + [c_0 \in W_1]$ supplies the extra hook $i = a+1 = 3$ repairing every $d \ge a+2$ case.

in PeelCoord(a,d) by climbing vertically along $a' = i$ from $t = 0$ to $t = q-i$, then horizontally along $t = q-i$ from $a' = i+1$ to $a' = h(q-i)$:

$$H_i(c): \quad (i,c_0) \lessdot (i,c_1) \lessdot \cdots \lessdot (i,c_{q-i}) \lessdot (i+1,c_{q-i}) \lessdot \cdots \lessdot (h(q-i),c_{q-i}).$$

Since $\mathrm{rk}(c_t) = \mathrm{rk}(c_0) + t$, the bottom and top ranks of $H_i(c)$ are $u = i + \mathrm{rk}(c_0)$ and $v = h(q-i) + \mathrm{rk}(c_0) + q - i$, so its rank sum is

$$u + v = h(q-i) + \mathrm{rk}(c_0) + \mathrm{rk}(c_q). \tag{1}$$

By construction, $(a',c_t) \in H_i(c)$ if and only if $\min(a', q-t) = i$ (`mem_hookCoords_iff`). The resulting hooks $H_0(c)$, …, $H_{\min(H,q)}(c)$ are pairwise disjoint (`allHooks_pairwise_disjoint`) and partition every coordinate (a',c_t) ($0 \le t \le q$, $0 \le a' \le h(t)$) whenever c satisfies the *cover condition* (`HookCoverCond`):

$$h(t) > H \implies q - t \le H \qquad (0 \le t \le q).$$

Indeed, for any (a',c_t) with $a' \le h(t)$, if $a' \le H$ then $\min(a', q-t) \le H$; if $a' > H$ then $h(t) > H$, so $q - t \le H$ and again $i = \min(a', q-t) \le \min(H,q)$.

Crucially, setting $H = a + \mathbf{1}_{c_0 \in W_1}$ rather than a repairs a coverage gap present when c lies wholly inside W_1 (where $h(t) = a+1$ for all t): on chain S_1 below ($q = d-1$), whenever $d \ge a+2$, the coordinate $(a+1, (0,\{V,M_1\}))$ has hook index $\min(a+1, d-1) = a+1 > a$ and is missed by every unrepaired hook $i \le a$ (`unrepaired_step3_misses_point`; Figure 3(b)). Including $i = a+1$ when $c_0 \in W_1$ restores complete coverage with zero extra casework.

Step 3 (W-chain partitions for $a \ge 2, d \ge 1$). Assume $a \ge 2$. We partition W into saturated chains whose parameters are summarized in Table 1:

- **Chain X ($d \ge 1$, length $d+5$, $q = d+4$):** $(0,\emptyset)$, …, $(d,\emptyset)$, $(d,\{V\})$, $(d,\{V,M_1\})$, $(d,\{V,M_1,M_2\})$, and $(d,\{V,M_1,M_2,Q\})$. Here $c_0 \notin W_1$ ($H = a$), and the top 3 elements ($t = d+2, d+3, d+4$) lie in W_1 with $q - t \le 2 \le a$, so `HookCoverCond` holds and (1) gives $u + v = h(q-i) + (d+4) \in \{n-1, n\}$.
- **Chain Y ($d \ge 1$, length $d+3$, $q = d+2$):** $(0,\{V\})$, …, $(d-1,\{V\})$, $(d-1,\{V,M_2\})$, $(d,\{V,M_2\})$, $(d,\{V,M_2,Q\})$. No element lies in W_1, so $h(t) \equiv a = H$ and $u + v = a + 1 + (d+3) = n - 1$.
- **Case $d = 1$, Chain S (length 2, $q = 1$):** $(0,\{V,M_1\})$, $(0,\{V,M_1,M_2\})$. Both lie in W_1, so $H = h(t) = a+1$, `Hook CoverCond` holds trivially, and $u + v = (a+1) + 2 + 3 = a + 6 = n$.
- **Case $d \ge 2$, Chain Z (length $d+1$, $q = d$):** $(0,\{V,M_2\})$, …, $(d-2,\{V,M_2\})$, $(d-2,\{V,M_1,M_2\})$, $(d-1,\{V,M_1,M_2\})$. Here $H = a$ and the top 2 elements ($t \in \{d-1, d\}$) lie in W_1 with $q - t \le 1 < a$, giving $u + v = h(q-i) + (d+4) \in \{n-1, n\}$.
- **Case $d \ge 2$, Chain S_1 (length d, $q = d-1$):** $(t,\{V,M_1\})$ for $0 \le t \le d-1$. All elements lie in W_1, so $H = h(t) = a+1$, and $u + v = (a+1) + 2 + (d+1) = a + d + 4 = n - 1$.
- **Case $d \ge 3$, Chain S_2 (length $d-2$, $q = d-3$):** $(t,\{V,M_1,M_2\})$ for $0 \le t \le d-3$. All elements lie in W_1, so $H = h(t) = a+1$, and $u + v = (a+1) + 3 + d = a + d + 4 = n - 1$.

One checks directly by case analysis on the 7 core states that $\{X,Y,S\}$ partitions W for $d = 1$ ($6 + 4 + 2 = 12 = 5(1) + 7$), $\{X,Y,Z,S_1\}$ partitions W for $d = 2$ ($7+5+3+2 = 17 = 5(2)+7$), and $\{X,Y,Z,S_1,S_2\}$ partitions W for $d \ge 3$ ($(d+5) + (d+3) + (d+1) + d + (d-2) = 5d+7$). Collecting all hook chains across the partition produces a BCD (and therefore an NCD) of $L(F(a,2,2,d))$ for all $a \ge 2, d \ge 1$ (`exists_bcd_F_a22d_ge2`, `exists_bcd_F_a221`).

Table 1. W-chain partition parameters for $F(a,2,2,d)$ ($a \ge 2$, $n = a+d+5$).

Chain	Domain	q	$\mathrm{rk}(c_0)$	$\mathrm{rk}(c_q)$	H	Hook $u+v$
X	$d \ge 1$	$d+4$	0	$d+4$	a	$n-1, n$
Y	$d \ge 1$	$d+2$	1	$d+3$	a	$n-1$
S	$d = 1$	1	2	3	$a+1$	n
Z	$d \ge 2$	d	2	$d+2$	a	$n-1, n$
S_1	$d \ge 2$	$d-1$	2	$d+1$	$a+1$	$n-1$
S_2	$d \ge 3$	$d-3$	3	d	$a+1$	$n-1$

Step 4 (Boundary cases $a = 1$, $d \geq 2$ and $a = d = 1$). For any even-length composition α, reversing vertex indices $i \mapsto (n-1)-i$ flips both adjacency order and edge directions simultaneously, yielding an order-preserving poset isomorphism $F(\alpha) \cong F(\alpha^{\mathrm{rev}})$ (`fencePosetReverseIso`). Hence for $a = 1$ and $d \geq 2$, $F(1,2,2,d) \cong F(d,2,2,1)$, and transporting the nested BCD from Step 3 settles $(1,2,2,d)$. Finally, for $a = d = 1$ ($n = 7, 29$ ideals), the 6 explicit saturated chains in `PeelCoord`$(1,1)$ (with rank spans $[0,7]$, $[1,6]$, $[1,5]$, $[2,5]$, $[2,5]$, $[3,4]$, all summing to 6 or 7) partition $L(F(1,2,2,1))$ into a nested BCD (`exists_bcd_F_1221`). Combining all cases proves Theorem 4 (`exists_bcd_and_ncd_F_a22d`). □

4. COMPUTATIONAL CENSUS, DIAGNOSTICS, AND OPEN CONJECTURE

To place $F(2,1,1,1,1,1,2)$ in context, we summarize the exhaustive computational census across all fences up to $n = 11$ elements and the off-center failure distribution across all 16,451 linear extensions.

Proposition 8 (Exhaustive Census Through $n = 11$ and Failure Diagnostics — Cert Only). *Exhaustive enumeration across all* 511 *compositions with* $2 \leq n \leq 10$ *elements* (`recompute.py`) *and independent C/Python census logs across all* 512 *compositions at* $n = 11$ *establish:*

1. *Under Mode 1 (printed reading),* $F(2,1,1,1,1,1,2)$ *is the unique failing fence for* $n \leq 10$*, and exactly* 13 *fences fail at* $n = 11$ *(including* $F(3,2,2,3)$ *and* $F(3,1,1,1,1,3)$*).*
2. *Under Mode 2 (genuine* 0/1*-word reading),* 13 *fences fail for* $n \leq 10$ *(*1 *at* $n = 8$*, namely* $F(1,2,1,2,1)$*;* 2 *at* $n = 9$*;* 10 *at* $n = 10$*) and* 27 *fences fail at* $n = 11$*.*
3. *Across all* 16,451 *linear extensions of* $F(2,1,1,1,1,1,2)$ *under Mode 1, the number of off-center chains ranges from* 2 *(*22 *extensions) to* 22 *(*1 *extension), peaking at* 13 *(*2,049 *extensions), while chain counts span* 19 *(*1,098 *extensions) to* 26 *(*6 *extensions); in* 7,484 *extensions the first off-center chain runs from rank* 1 *to rank* 8*.*

Table 2 records the exact failure counts by element count n (plotted in Figure 2(c)). Across the nine independently written C and Python verification programs (eight independent C and Python implementations across multiple verification rounds plus the standalone script `recompute.py`), every program reports 0 valid linear extensions out of 16,451 for $F(2,1,1,1,1,1,2)$ under Mode 1, and every program testing Mode 2 reports 0 out of 16,451 as well. Independent cross-checks also verified 0 valid linear extensions under colex, reversed colex, and the order-dual fence.

Table 2. Exhaustive census of failing fences by element count n under Mode 1 and Mode 2 ($n \leq 10$ replayed in `recompute.py`; $n = 11$ from independent C/Python census logs).

n	Comps.	Mode 1	Mode 2	Notable failures
2 … 7	63	0	0	None (matches [3])
8	64	0	1	Mode 2: $(1,2,1,2,1)$ (45 ideals)
9	128	0	2	Mode 2: $(1,2,1,3,1)$, $(1,3,1,2,1)$
10	256	1	10	Mode 1: $(2,1,1,1,1,1,2)$ (105 id.)
11	512	13	27	Mode 1: $(3,2,2,3)$, $(3,1,1,1,1,3)$, …

For the symmetric four-part family $F(k,2,2,k)$, exhaustive search over all linear extensions found 0 Mode-1 BCDs for $k = 3,4,5,6,7,8$ (checking 12,000, 84,150, 532,532, 3,139,136, 17,566,848, and 94,477,500 extensions), whereas an explicit non-linear-extension labeling R_k (which defers the element immediately past each of the three turning points) produced a Mode-1 nested BCD for every tested $2 \leq k \leq 80$ (for $k = 3$, this is certified in Lean by `f3223R3_mode1_bcd_ncd`). While Theorem 4 settles Conjecture 1.5 for all $F(k,2,2,k)$, the LCD behavior for arbitrary k remains open.

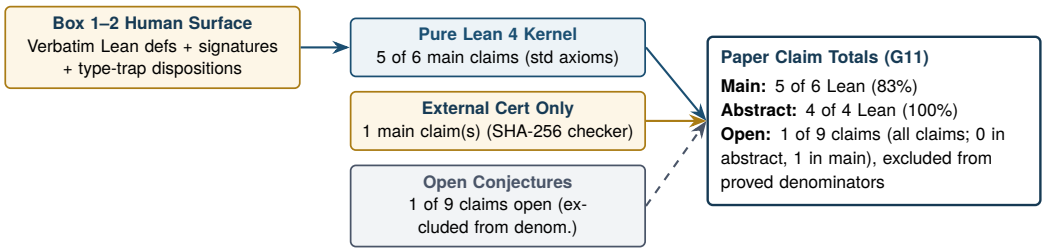
Conjecture 9 (Infinite Mode-1 Failure and Labeling Family for $F(k,2,2,k)$ — Open). *Under Mode 1 (printed reading): (i) for every* $k \geq 3$*, no linear extension of* $F(k,2,2,k)$ *yields a BCD; and (ii) for every* $k \geq 2$*, the explicit non-linear-extension labeling* R_k *yields a nested BCD of* $L(k,2,2,k)$*.*

5. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel & Toolchain:** leanprover/lean4:v4.35.0; committed axiom receipt anchored to git commit 2f09f46076a2 (independent gate-box kernel replay pending); standard axioms only (propext, Classical.choice, Quot.sound).
2. **Core Definitions & Fidelity (Boxes 1–2):** Verify that the verbatim Lean definitions in Box 1 match Section 2 and that each Lean signature in Box 2 matches its stated theorem.
3. **External Certificates & Written Proofs:** 1 SHA-256-pinned external artifact(s) plus any claims marked Written or Unsettled in Table 4 below.

Reader's Trust-Flow Architecture (Distinct Evidence Paths to Claim Totals)



Anti-Shortcut Audit Strip — Source-Level Approx (Gate Replay Pending)

WARN: Committed receipt (gate replay pending) | PASS: sorry/admit: 0 (per committed receipt)
PASS: native_decide: 0 (per committed receipt) | PASS: Custom axioms & meta: 0 (per committed receipt)
PASS: Coverage: 197/197 public theorems (source-level approx) | WARN: Fidelity: 7/7 exact (draft cross-family)
WARN: Witnesses: pending gate replay | WARN: Human referee: pending

Box 1 — Verbatim Lean 4 Custom Definitions (Human Audit Surface)

FenceLexCounterexample.ChainDecomp (FenceLexCounterexample/Basic.lean:853-856) — Section 2 (disjoint saturated-chain partition of L(P))

```
structure ChainDecomp {V : Type*} [Fintype V] [DecidableEq V] (P : FinPoset V) where
  chains : List (SaturatedChain P)
  pairwise_disjoint : chains.Pairwise (fun c₁ c₂ => ∀ I : OrderIdeal P, I ∈ c₁.elems
    ↪ → I ∉ c₂.elems)
  covers_all : ∀ I : OrderIdeal P, ∃ c ∈ chains, I ∈ c.elems
```

FenceLexCounterexample.Covers (FenceLexCounterexample/Basic.lean:622-623) — Section 2 (covering relation I <* J in L(P))

```
def Covers {V : Type*} (P : FinPoset V) (I J : OrderIdeal P) : Prop :=
  I.carrier ⊆ J.carrier ∧ J.rank = I.rank + 1
```

FenceLexCounterexample.IsBCD (FenceLexCounterexample/Basic.lean:868-871) — Section 2 (bottom-centered CD: u + v in {n-1, n})

Context binders in scope: `variable {V : Type*} [Fintype V] [DecidableEq V] {P : FinPoset V}`

```
def IsBCD (cd : ChainDecomp P) : Prop :=
  ∀ c ∈ cd.chains,
    c.botRank + c.topRank = Fintype.card V ∨
    c.botRank + c.topRank + 1 = Fintype.card V
```

FenceLexCounterexample.IsLinearExtension (FenceLexCounterexample/Basic.lean:1464-1465) — Section 2 (order-preserving bijection L)

```
def IsLinearExtension {n : ℕ} (P : FinPoset (Fin n)) (L : Fin n ≃ Fin n) : Prop :=
  ∀ x y : Fin n, P.le x y → L x ≤ L y
```

FenceLexCounterexample.IsNCD (FenceLexCounterexample/Basic.lean:886-887) — Section 2 (nested chain decomposition)

Context binders in scope: `variable {V : Type*} [Fintype V] [DecidableEq V] {P : FinPoset V}`

```
def IsNCD (cd : ChainDecomp P) : Prop :=
  ∀ c₁ ∈ cd.chains, ∀ c₂ ∈ cd.chains, IsNestedPair c₁ c₂
```

FenceLexCounterexample.IsSCD (FenceLexCounterexample/Basic.lean:863-864) — Section 2 (symmetric CD: u + v = n)

Context binders in scope: `variable {V : Type*} [Fintype V] [DecidableEq V] {P : FinPoset V}`

```
def IsSCD (cd : ChainDecomp P) : Prop :=
  ∀ c ∈ cd.chains, c.botRank + c.topRank = Fintype.card V
```

FenceLexCounterexample.IsTCD (FenceLexCounterexample/Basic.lean:875-878) — Section 2 (top-centered CD: u + v in {n, n+1})

Context binders in scope: `variable {V : Type*} [Fintype V] [DecidableEq V] {P : FinPoset V}`

```
def IsTCD (cd : ChainDecomp P) : Prop :=
  ∀ c ∈ cd.chains,
    c.botRank + c.topRank = Fintype.card V ∨
    c.botRank + c.topRank = Fintype.card V + 1
```

FenceLexCounterexample.SaturatedChain (FenceLexCounterexample/Basic.lean:741-744) — Section 2 (saturated chain of ideals)

```
structure SaturatedChain {V : Type*} (P : FinPoset V) where
  elems : List (OrderIdeal P)
  nonempty : elems ≠ []
  isChain : IsAdjacentRel (Covers P) elems
```

FenceLexCounterexample.fencePoset (FenceLexCounterexample/Basic.lean:296-302) — Section 2 (rising-first fence poset F(alpha))

```
def fencePoset (α : List ℕ) : FinPoset (Fin (α.sum + 1)) where
  order := fencePartialOrder α
  decLe := instDecidableFenceLe α
  decLt u v :=
    letI := instDecidableFenceLe α u v
    letI := instDecidableFenceLe α v u
    inferInstanceAs (Decidable (fenceLe α u v ∧ ¬ fenceLe α v u))
```

FenceLexCounterexample.headFence (FenceLexCounterexample/HeadBridge.lean:1177-1177) — Section 1 & 3 (10-element fence F(2,1,1,1,1,1,2))

```
abbrev headFence : FinPoset (Fin 10) := fencePoset headAlpha
```

FenceLexCounterexample.lcd (FenceLexCounterexample/LexChainDecomp.lean:707-711) — Section 2 (greedy lexicographic chain decomposition)

Context binders in scope: `variable {n : ℕ} {P : FinPoset (Fin n)}`

```
noncomputable def lcd (mode : LexMode) (P : FinPoset (Fin n)) (L : Fin n ≃ Fin n) :
    ChainDecomp P where
  chains := lcdAux mode P L Finset.univ
  pairwise_disjoint := (lcdAux_spec mode P L Finset.univ).1
  covers_all I := ((lcdAux_spec mode P L Finset.univ).2 I).mp (Finset.mem_univ I)
```

FenceLexCounterexample.lexIncLT (FenceLexCounterexample/LexChainDecomp.lean:108-109) — Section 2 (Mode 1 increasing-sequence lex order)

```
def lexIncLT {n : ℕ} (L : Fin n ≃ Fin n) (s t : Finset (Fin n)) : Prop :=
  listLexLT (incLabelSeq L s) (incLabelSeq L t)
```

FenceLexCounterexample.lexWordLT (FenceLexCounterexample/LexChainDecomp.lean:112-113) — Section 2 (Mode 2 binary 0/1-word order, 0 < 1)

```
def lexWordLT {n : ℕ} (L : Fin n ≃ Fin n) (s t : Finset (Fin n)) : Prop :=
  listLexLT (zeroOneWord L s) (zeroOneWord L t)
```

Supporting Semantic Closure Index (non-Box-1 helpers): FenceLexCounterexample.OrderIdeal (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.Core (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.Core.all (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.Core.card (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.Core.hasM1 (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.Core.hasQ (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.FinPoset (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.FinPoset.le (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.FinPoset.lt (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.HasSymmetricRanks (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.IsAdjacentRel (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.IsNestedPair (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.LexMode (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.OrderIdeal.rank (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.PeelCoord (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.PeelCoord.rank (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.PosetIso (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.PosetIso.symm (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.SaturatedChain.botRank (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.SaturatedChain.topRank (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.WCoord (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.WCoord.inW1 (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.WCoord.rank (FenceLexCounterexample/PeelBCDAllAD.lean), FenceLexCounterexample.extendChain (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.f3223Alpha (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.f3223Fence (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.f3223Mode2Labeling (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.fenceEdgeDirs (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.fenceEdgeDirsAux (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.fenceEdgeUp (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.fenceLe (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.fenceLeBool (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.fencePartialOrder (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.headAlpha (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.headWitnessLabeling (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.incLabelSeq (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.instDecidableFenceLe (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.lcdAux (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.lexKey (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.listLexLT (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.minRank (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.mkEquivFromLists (FenceLexCounterexample/HeadBridge.lean), FenceLexCounterexample.modeKey (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.pickLexMin (FenceLexCounterexample/LexChainDecomp.lean), FenceLexCounterexample.rankCount (FenceLexCounterexample/Basic.lean), FenceLexCounterexample.zeroOneWord (FenceLexCounterexample/LexChainDecomp.lean).

Box 2 — Verbatim Main Theorem Signatures, Context Binders & Applicability

FenceLexCounterexample.headFence_refutation_mode1 (FenceLexCounterexample/HeadCounterexample.lean:24-27) src:0e02bb0e480c83

```
theorem headFence_refutation_mode1 (L : Fin 10 ≃ Fin 10) (hL : IsLinearExtension
  ↪ headFence L) :
    ¬ IsSCD (lcd .mode1 headFence L) ∧ ¬ IsBCD (lcd .mode1 headFence L) ∧
    ¬ IsTCD (lcd .mode1 headFence L) ∧ ¬ IsNCD (lcd .mode1 headFence L)
```

Assumptions: Any linear extension L : Fin 10 ~= Fin 10 of headFence (16,451 exist via card_linearExtension_headFence). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

FenceLexCounterexample.headFence_refutation_mode2 (FenceLexCounterexample/HeadCounterexample.lean:32-35) src:378ce7324f8e9d

```
theorem headFence_refutation_mode2 (L : Fin 10 ≃ Fin 10) (hL : IsLinearExtension
  ↪ headFence L) :
    ¬ IsSCD (lcd .mode2 headFence L) ∧ ¬ IsBCD (lcd .mode2 headFence L) ∧
    ¬ IsTCD (lcd .mode2 headFence L) ∧ ¬ IsNCD (lcd .mode2 headFence L)
```

Assumptions: Any linear extension L : Fin 10 ~= Fin 10 of headFence under Mode 2 (0/1-word order, 0 < 1). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

FenceLexCounterexample.exists_bcd_and_ncd_F_a22d (FenceLexCounterexample/PeelBCDAllAD.lean:2048-2066) src:0799736feb3bf5

```
theorem exists_bcd_and_ncd_F_a22d (a d : ℕ) (ha : 1 ≤ a) (hd : 1 ≤ d) :
    ∃ cd : ChainDecomp (fencePoset [a, 2, 2, d]), IsBCD cd ∧ IsNCD cd
```

Assumptions: Any natural numbers a >= 1 and d >= 1 (unconditional across the entire family F(a, 2, 2, d)). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

FenceLexCounterexample.headWitness_mode1_scd_ncd (FenceLexCounterexample/HeadBridge.lean:2519-2541) src:f11c6563962651

```
theorem headWitness_mode1_scd_ncd :
    IsSCD (lcd .mode1 headFence headWitnessLabeling) ∧
    IsNCD (lcd .mode1 headFence headWitnessLabeling) ∧
    (lcd .mode1 headFence headWitnessLabeling).chains.length = 19
```

Assumptions: Closed witness headWitnessLabeling (non-extension via headWitness_not_linearExtension). Applicability: Unconditional (closed statement; no outer premise witness needed).

FenceLexCounterexample.f3223Mode2_bcd_ncd (FenceLexCounterexample/HeadBridge.lean:2812-2834) src:30076822e5d9f7

```
theorem f3223Mode2_bcd_ncd :
    IsBCD (lcd .mode2 f3223Fence f3223Mode2Labeling) ∧
    IsNCD (lcd .mode2 f3223Fence f3223Mode2Labeling) ∧
    (lcd .mode2 f3223Fence f3223Mode2Labeling).chains.length = 16
```

Assumptions: Closed witness f3223Mode2Labeling (linear extension via f3223Mode2_isLinearExtension). Applicability: Unconditional (closed statement; no outer premise witness needed).

FenceLexCounterexample.lexIncLT_iff_lexWordLT_rev_of_card_eq (FenceLexCounterexample/LexChainDecomp.lean:434-437) src:9f91514da8c4c7

```
theorem lexIncLT_iff_lexWordLT_rev_of_card_eq {n : ℕ} (L : Fin n ≃ Fin n)
    {s t : Finset (Fin n)} (hcard : s.card = t.card) :
    lexIncLT L s t ↔ lexWordLT L t s
```

Assumptions: Any n : Nat, bijection L : Fin n ~= Fin n, and equal-size subsets s, t : Finset (Fin n). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

FenceLexCounterexample.isNCD_iff_isSCD_of_symmetric_peak (FenceLexCounterexample/Basic.lean:1229-1320) src:704c9eba727531

Context binders in scope: `variable {V : Type*} [Fintype V] [DecidableEq V] {P : FinPoset V}`

```
theorem isNCD_iff_isSCD_of_symmetric_peak (hsym : HasSymmetricRanks P) (m : ℕ)
    (h2m : 2 * m = Fintype.card V)
    (hpeak : ∀ r ≤ Fintype.card V, r ≠ m → rankCount P r < rankCount P m)
    (cd : ChainDecomp P) :
    IsNCD cd ↔ IsSCD cd
```

Assumptions: Any finite poset P of size 2m with symmetric ranks and unique peak at rank m (satisfied by headFence). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

Table 3. Lean 4 Type-Trap Audit Checklist — Source-Level Approximation (Gate Replay Pending).[a]

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`HasSymmetricRanks`	`nat_sub`	`- r`	intended_convention	Quantifier guards r <= Fintype.card V before evaluating Fintype.card V - r.
`fenceLeBool`	`nat_sub`	`- u.val`	intended_convention	Short-circuit conjunction decide (u.val <= v.val) && ... ensures v.val >= u.val before subtraction.
`fenceLeBool`	`nat_sub`	`- v.val`	intended_convention	Short-circuit conjunction decide (v.val <= u.val) && ... ensures u.val >= v.val before subtraction.
`lexKey`	`nat_sub`	`- 1`	intended_convention	Fin n forces n >= 1 on any nonempty sum (empty sum is 0).

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`lexKey`	`nat_sub`	`- (L x)`	intended_convention	Type Fin n guarantees (L x).val < n, so n - 1 - (L x).val never underflows.
`modeKey`	`nat_sub`	`- 1`	intended_convention	Since 2^n >= 1 for every natural n, 2^n - 1 never underflows.
`modeKey`	`nat_sub`	`- lexKey`	intended_convention	lexKey_lt_two_pow proves lexKey L s < 2^n, preventing underflow.
`pickLexMin`	`classical_choose`	`Classical.choose`	intended_convention	modeKey_injective prevents ties; pickLexMin returns the unique minimum.
`instDecidableLe`	`local_instance_or_notation`	`instDecidableLe`	intended_convention	Decidable instance; no custom axioms or overrides.
`instDecidableLt`	`local_instance_or_notation`	`instDecidableLt`	intended_convention	Decidable instance; no custom axioms or overrides.
`instDecidableFenceLe`	`local_instance_or_notation`	`instDecidableFenceLe`	intended_convention	Decidable instance; no custom axioms or overrides.
`instFintype`	`local_instance_or_notation`	`instFintype`	intended_convention	Fintype instance; no custom axioms or overrides.
`instDecidableCovers`	`local_instance_or_notation`	`instDecidableCovers`	intended_convention	Decidable instance; no custom axioms or overrides.
`instDecidableIsLinearExtension`	`local_instance_or_notation`	`instDecidableIsLinearExtension`	intended_convention	Decidable instance; no custom axioms or overrides.
`instDecidableListLexLT`	`local_instance_or_notation`	`instDecidableListLexLT`	intended_convention	Decidable instance; no custom axioms or overrides.
`instFintype`	`local_instance_or_notation`	`instFintype`	intended_convention	Fintype instance; no custom axioms or overrides.

[a]Covers the specified Lean 4 hazard set (`Nat` subtraction/division/mod, totalized real/rational division, `Real.log`/`rpow`, `Finset.range`, `Classical.choose`, local instances); no finite checklist certifies all semantics.

Table 4. Per-Claim Reader's Trust Boundary Ledger (G11 Denominators Derived Mechanically).[a]

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 1: Every linear extension of F(2,1,1,1,1,1,2) fails SCD, BCD, TCD, and NCD under Mode 1 (refuting Conj./Q. 4.6)	Lean Kernel	`FenceLexCounterexample.headFence_refutation_mode1`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (28 decide+kernel chunks + bridge); 105 ideals, symmetric ranks, 16,451 extensions (Head Bridge).
Thm. 2: Every linear extension of F(2,1,1,1,1,1,2) fails SCD, BCD, TCD, and NCD under Mode 2 (0/1-word order)	Lean Kernel	`FenceLexCounterexample.headFence_refutation_mode2`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (14 Mode-2 chunks + bridge); 0/1-word order (0 < 1).
Prop. 3: Non-extension labeling L_wit of F(2,1,1,1,1,1,2) gives a 19-chain Mode-1 SCD and NCD (Conj. 6.2 intact)	Lean Kernel	`FenceLexCounterexample.headWitness_mode1_scd_ncd`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (non-extension proved by head Witness_not_linearExtension).
Thm. 4: Every F(a,2,2,d) (a >= 1, d >= 1) admits a nested BCD (Conj. 1.5 for this family)	Lean Kernel	`FenceLexCounterexample.exists_bcd_and_ncd_F_a22d`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel unconditionally for all a >= 1, d >= 1 (exact count 5ad+7a+7d+10 in card_order Ideal_F_a22d).
Lem. 5: Mode 1 and Mode 2 are exact reverses on equal-cardinality subsets	Lean Kernel	`FenceLexCounterexample.lexIncLT_iff_lexWordLT_rev_of_card_eq`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel for all n, bijections L, and equal-size subsets s, t.
Lem. 6: On even-size symmetric-rank posets with unique peak, NCD, BCD, and TCD equal SCD	Lean Kernel	`FenceLexCounterexample.isNCD_iff_isSCD_of_symmetric_peak`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (BCD/TCD collapses via is BCD/isTCD_iff_isSCD_of_symmetric_ranks).
Prop. 7: Linear extension L_sep of F(3,2,2,3) gives a 16-chain Mode-2 nested BCD (separating Mode 1 and Mode 2)	Lean Kernel	`FenceLexCounterexample.f3223Mode2_bcd_ncd`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (f3223Mode2_isLinearExtension; 0/12,000 Mode-1 failure in recompute.py).

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Prop. 8: Exhaustive census through n = 11 (Mode 1: 1 at n<=10, 13 at n=11; Mode 2: 13 at n<=10, 27 at n=11) and 16,451-ext. diagnostics	Cert Only	recompute.py (0ffd3e92ca49) [regenerable; replay pending]	External cert only; fidelity: n/a	Exact computation in recompute.py (n <= 10, 511 comps. & diagnostics) + C/Python logs at n = 11 (512 comps.; beyond kernel budget).
Conj. 9: F(k,2,2,k) has no Mode-1 linear-extension BCD for k >= 3 and R_k yields a Mode-1 nested BCD for k >= 2	Unsettled	Open conjecture	Lean: none; fidelity: n/a	Open conjecture beyond computed ranges k <= 8 and k <= 80 (excluded from G11 denominators).

[a]Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: ec26e71bf64c | Lean mode: committed_receipt | Closure SHA-256: b82839186c8e | Git anchor: 2f09f46076a2 | Axioms SHA-256: b2a6864ebc28 | Fidelity SHA-256: f60d1e42166a.

6. PRIOR WORK AND OPEN QUESTIONS

Morier-Genoud and Ovsienko [4] [to check] introduced q-deformed rationals and posed rank unimodality for fence lattices; Gansner [2] [to check] studied ideal lattices and chain decompositions of up-down posets. McConville, Sagan, and Smyth [3] formulated Conjectures 1.4, 1.5, and 6.2, introduced the greedy LCD for arbitrary labelings, proved Conjecture 1.5 for $s \leq 3$ parts (Theorem 1.6 of [3]), for $(k, 1, \ldots, k, 1, \ell)$ $(1 \leq \ell \leq k)$, and for one sufficiently long segment, and verified Conjecture 6.2 for $|F| \leq 7$. Oğuz and Ravichandran [5] proved Conjecture 1.4 (Theorem 1.2 of [1]) algebraically and asked about SCDs of circular fences. Elizalde and Sagan [1] studied partial rank symmetry, restated Conjecture 1.5 as 4.5, and posed Conjecture/Question 4.6.

Three prior-work search logs passed `litsearch.py check` (exit 0; 14 + 5 + 13 = 32 answered arXiv queries, 9 OEIS queries, 28 GitHub queries, DataCite, Zenodo, zbMATH, OpenAlex citers of [1] (6) and [3] (40), 19 citing arXiv preprints read in full, 1 Google Scholar query, and 7 keyed CORE requests returning 36 records). Neither $F(2,1,1,1,1,1,2)$ nor an answer to Conjecture/Question 4.6 was found in the searches logged; Theorem 4 has not received a separate search beyond these logs.

Unsearched venues and `[to check]` items. MathSciNet, books in print, paywalled portals beyond author copies of [1] and [3], Google Scholar beyond one query (API monthly query cap), the BIRS 2020 poster (blocked by `robots.txt`), and full texts of citing HAL/UQAM theses were not searched. References [4] and [2] were consulted via secondary citations and abstracts only (`[to check]` in `references-checked.md`).

Open questions. (1) Does Conjecture 1.5 of [3] hold for every fence $F(\alpha)$? (2) Does Conjecture 6.2 of [3] (arbitrary labelings) hold in general, and does R_k work for all $F(k,2,2,k)$ (Conjecture 9)? (3) Is there a short conceptual proof that every linear extension of $F(2,1,1,1,1,1,2)$ leaves at least 2 chains off center?

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

Manuscript files, figure scripts, and exact verifiers reside in `papers/fence-lattice-chain-decompositions/draft1/`, and the Lean 4 + Mathlib library in `lean/FenceLexCounterexample/` (37 files, 197 declarations). Running `python3 recompute.py` replays all hand examples, all 16,451 extensions of $F(2,1,1,1,1,1,2)$ in both modes, $F(3,2,2,3)$ (12,000 extensions), the $n \leq 10$ census (511 compositions), and $1 \leq a, d \leq 8$ (64 cases, 21 Step-3 gap witnesses). Running `lake build FenceLexCounterexample` and `lake env lean FenceLexCounterexample/Verify.lean` replays all 197 theorems.

AI research agents built and directed by Tyler B. Hughes generated the computational search, independent cross-checks, repaired peel proof, Lean 4 formalization, figures, and manuscript text. No human mathematician has yet refereed the proof.

Drafted: November 2026 **Updated:** November 24, 2026

REFERENCES

(1) Elizalde, S.; Sagan, B. E. Partial rank symmetry of distributive lattices for fences. *Ann. Comb.* **2023** (online 2022), *27* (2), 433–454. DOI: 10.1007/s00026-022-00600-8. arXiv:2201.03044v1.

(2) Gansner, E. R. On the lattice of order ideals of an up-down poset. *Discrete Math.* **1982**, *39* (2), 113–122. DOI: 10.1016/0012-365x(82)90134-0.

(3) McConville, T.; Sagan, B. E.; Smyth, C. On a rank-unimodality conjecture of Morier-Genoud and Ovsienko. *Discrete Math.* **2021**, *344* (8), 112483. DOI: 10.1016/j.disc.2021.112483. arXiv: 2008.13232.

(4) Morier-Genoud, S.; Ovsienko, V. q-deformed rationals and q-continued fractions. *Forum Math. Sigma* **2020**, *8*, e13. DOI: 10.1017/fms.2020.9. arXiv:1812.00170.

(5) Oğuz, E. K.; Ravichandran, M. Rank polynomials of fence posets are unimodal. *Discrete Math.* **2023**, *346* (2), 113218. DOI: 10.1016/j.disc.2022.113218. arXiv:2112.00518.