MATHDISCOVERIES
Mathematical Preprint

Working Draft

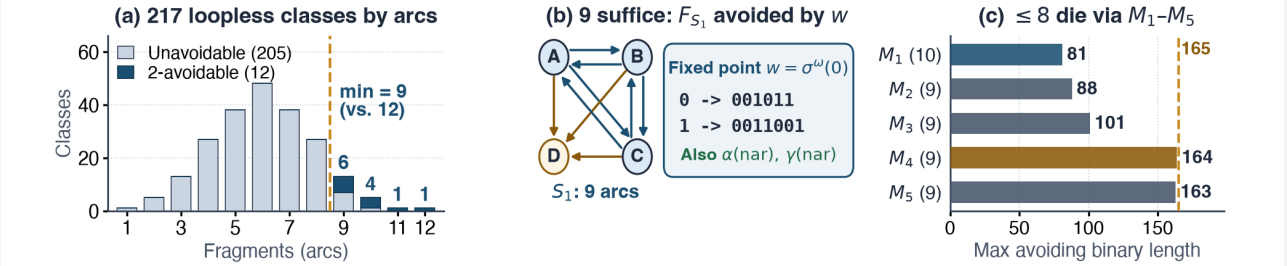
The Fewest Fragments of a 2-Avoidable Four-Variable xyx-Formula

Tyler B. Hughes[†]

[†]Independent Researcher, United States

Supporting Information: `recompute.py`, audited Lean 4 + Mathlib library (`lean/XYXAvoid/`), and trust-boundary 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: An xyx-formula of a loopless digraph G has one fragment XYX per arc $X \to Y$, and is 2-avoidable if some infinite binary word avoids every non-empty substitution realizing all fragments as factors. Ochem and Rosenfeld (2021) proved the complete 12-arc digraph on 4 vertices is 2-avoidable (Theorem 23) and asked in Section 7 (Discussion) for a 2-avoidable 4-variable xyx-formula with fewer fragments. Under the loopless reading (excluding cubes XXX), 9 fragments suffice and 8 or fewer do not. On the upper side, Cassaigne-type morphism descent proves that the fixed point w of $\sigma : 0 \mapsto 001011,\ 1 \mapsto 0011001$ (and Narayana images $\alpha(\mathrm{nar}), \gamma(\mathrm{nar})$) avoids the 9-fragment formula $F_{S_1} = ABA.BAB.ACA.CAC.BCB.CBC.ADA.BDB.CDC$. On the lower side, every loopless 4-vertex digraph lies either above 7 minimal classes (≥ 9 arcs) or below 5 maximal unavoidable digraphs $M_1, \ldots, M_5$ whose exhaustive trees terminate at lengths $81, 88, 101, 164, 163$; hence every ≤ 8-fragment formula occurs in every binary word of length 165. Our Lean 4 + Mathlib library (`lean/XYXAvoid/`) proves the 217-class census, dichotomy, ≤ 8-arc reduction, descent soundness, and lower-bound bridge over the 5 tree certificates.

Verification Scope (G11): WARN: Human referee: pending
Main results: 4 of 7 in Lean Kernel (57%) · 1 Lean + Cert · 2 Cert Only
Abstract claims: 1 of 4 in Lean Kernel (25%) · 1 Lean + Cert · 2 Cert Only

Full Trust Boundary: Section 5.

MSC 2020: 68R15, 05C20, 05A05 **Keywords:** avoidability of formulas, combinatorics on words, xyx-formulas, binary morphisms, digraph homomorphisms, Lean 4 + Mathlib formalization

1. INTRODUCTION AND MAIN RESULTS

A *formula* in combinatorics on words is a finite list of pattern fragments over an alphabet of variables. A word over a finite alphabet Σ contains an *occurrence* of a formula if a single choice of non-empty words in Σ^+ for the variables places every fragment somewhere in the word as a contiguous factor. Ochem and Rosenfeld [1] studied *xyx-formulas*, whose fragments all have the palindromic shape XYX, and represented each formula by a directed graph G with one arc $X \to Y$ for every fragment XYX (their Section 5). A formula is 2-*avoidable* if some infinite binary word avoids it.

In Theorem 23 of [1], Ochem and Rosenfeld proved that the complete loopless digraph on 4 vertices—having all 12 arcs, hence 12 fragments—is 2-avoidable (specifically, avoided by the fixed point of $0 \mapsto 001,\ 1 \mapsto 011$). In Section 7 (Discussion) of both the journal and preprint versions of [1], they listed among items attempted "without success" the problem:

> *"Improve Theorem 23 by showing that some xyx-formula on 4 variables and fewer fragments is 2-avoidable."*

The loopless reading. Taken strictly to the letter without excluding self-loops XXX, the question is vacuous: the 4-fragment formulas $AAA.ABA.ACA.ADA$ and $AAA.BBB.CCC.DDD$ are avoided by every cube-free binary word (such as the Thue–Morse word). However, Ochem and Rosenfeld [1] introduce Theorem 23 as providing "an xyx-formula with index 2 that is not divisible by AAA." Accordingly, we read the question as asking about *loopless xyx-formulas*—those having no fragment XXX, corresponding to loopless directed graphs. Under this reading, we establish that the exact minimum number of fragments is 9.

Theorem 1 (Eight or Fewer Fragments Are Unavoidable — Lean + Cert). *Assuming the maximal-digraph unavoidability hypothesis* `h_max5` *(that none of the* 5 *maximal digraphs* $M_1, \dots, M_5$ *of Table 1 is* 2*-avoidable, discharged by Proposition 7), for every* 12*-bit loopless digraph* $m \in \mathrm{Mask}_4$ *on* 4 *vertices with at most* 8 *arcs, the associated xyx-formula* F_m *is not* 2*-avoidable. Equivalently, every loopless xyx-formula on at most* 4 *variables with at most* 8 *fragments occurs in every binary word of length* 165*.*

Theorem 2 (Nine Fragments Suffice — Cert Only). *Let* $w = \sigma^\omega(0)$ *be the infinite binary fixed point of the morphism*

$$\sigma : 0 \mapsto 001011, \qquad 1 \mapsto 0011001.$$

Then w *avoids the* 9*-fragment loopless xyx-formula*

$$F_{S_1} = ABA.BAB.ACA.CAC.BCB.CBC.ADA.BDB.CDC,$$

whose digraph S_1 *consists of a bidirected triangle on* A, B, C *together with the three arcs* $A \to D$, $B \to D$, *and* $C \to D$*. Moreover, the two binary images* $\alpha(\mathrm{nar})$ *and* $\gamma(\mathrm{nar})$ *of Narayana's word also avoid* F_{S_1}*. Combined with Theorem 1,* 9 *is the exact minimum number of fragments of a* 2*-avoidable loopless xyx-formula on* 4 *variables.*

Our proof architecture separates the finite combinatorial census and algebraic soundness theorems proved unconditionally inside the Lean 4 + Mathlib kernel (`lean/XYXAvoid/`) from the external deterministic computational certificates (the 5 exhaustive binary search trees for $M_1, \dots, M_5$ and the Cassaigne-type morphism descent closures for F_{S_1}), summarized in Figure 1. Specifically, the unconditional Lean 4 + Mathlib core establishes the following four lemmas.

Lemma 3 (Exact Four-Variable Class Census — Lean Kernel). *The set* `nonemptyClasses4` *of non-empty* S_4*-isomorphism classes of loopless digraphs on* 4 *vertices has exact cardinality* 217 *(and* 218 *when the empty digraph is included).*

Lemma 4 (Four-Variable Antichain Dichotomy — Lean Kernel). *Every* 12*-bit loopless digraph* $m \in \mathrm{Mask}_4$ *on* 4 *vertices either contains an isomorphic subgraph copy of at least one of the* 7 *minimal digraphs* $\{S_1, G_2, G_3, G_4, G_5, G_6, S_{10}\}$ *(each having* 9 *or* 10 *arcs) or embeds as a subgraph into at least one of the* 5 *maximal digraphs* $\{M_1, M_2, M_3, M_4, M_5\}$ *(exclusively).*

Lemma 5 (Reduction of All At-Most-Eight-Arc Classes — Lean Kernel). *Every non-empty canonical isomorphism class* $c \in$ `nonemptyClasses4` *with at most* 8 *arcs belongs to* `unavoidableClasses4` *(the* 205*-class complement of the upward closure of the* 7 *minimal classes), and hence embeds into at least one of* $M_1, \dots, M_5$ *by Lemma 4.*

Lemma 6 (Two-Move Lexicographic Descent Soundness — Lean Kernel). *Let* G *be a loopless digraph on a finite vertex set* V*, let* $W : \mathbb{N} \to \alpha$ *be an infinite word, and let* $\mathrm{freeVars} : \mathcal{S} \to \mathbb{N}$, $\mathrm{Sat} : \mathcal{S} \to \mathbb{N} \to \mathrm{Prop}$, $C \subseteq \mathcal{S}$*, and* $s_0 \in C$ *be such that every occurrence* $h : V \to \alpha^+$ *of* F_G *in* W *satisfies* $\mathrm{Sat}(s_0, \sum_{v \in V} |h(v)|)$*. If every* $s \in C$ *and* $\mu \in \mathbb{N}$ *with* $\mathrm{Sat}(s, \mu)$ *admits either*

1. *a freeze transition to* $s' \in C$ *with* $\mathrm{freeVars}(s') < \mathrm{freeVars}(s)$ *and* $\mathrm{Sat}(s', \mu')$ *for some* $\mu' \in \mathbb{N}$*, or*
2. *a desubstitution transition to* $s' \in C$ *with* $\mathrm{freeVars}(s') = \mathrm{freeVars}(s)$ *and* $\mathrm{Sat}(s', \mu')$ *for some* $\mu' < \mu$*,*

then W *avoids* F_G*.*

The maximal-digraph unavoidability hypothesis `h_max5` of Theorem 1 is discharged by five exhaustive binary search trees.

Proposition 7 (Five Maximal Unavoidable Trees — Cert Only). *Unpruned exhaustive binary-tree search terminates for each of the* 5 *maximal unavoidable digraphs* M_1, M_2, M_3, M_4, M_5 *(Table 1) at maximum avoiding binary-word lengths* 81, 88, 101, 164, *and* 163 *(*62,118, 122,990, 727,866, 645,746, *and* 636,758 *words tested, respectively). Consequently, every binary word of length* 165 *contains an occurrence of all* 5 *formulas, and none of* $M_1, \dots, M_5$ *is* 2*-avoidable.*

Concept in Action: 14-Letter Occurrence of F_{G_5} vs. F_{S_1}

```
w = "01101001101101"  # alpha(nar)[:14]
h = {"A": "1", "B": "0", "C": "10", "D": "01"}
G5 = ["ABA","ACA","ADA","BAB","BCB","CAC","CDC","DAD","DBD"]
S1 = ["ABA","BAB","ACA","CAC","BCB","CBC","ADA","BDB","CDC"]
assert all(h[f[0]]+h[f[1]]+h[f[2]] in w for f in G5)
assert [f for f in S1
        if h[f[0]]+h[f[1]]+h[f[2]] not in w] == ["CBC","BDB"]
```

This short check (executed verbatim by `recompute.py`) illustrates how sensitive 9-fragment formulas are to arc orientation: in the 14-letter binary word `01101001101101`, the single assignment $A = 1$, $B = 0$, $C = 10$, $D = 01$ realizes all 9 fragments of F_{G_5} (for example $CDC = 10|01|10 = 100110$ at letters 5 to 10), whereas the same assignment fails F_{S_1} on $CBC = 10010$ and $BDB = 0010$—and indeed an exact unbounded-block-length search confirms that F_{S_1} has zero occurrences anywhere in `01101001101101` (and zero occurrences in the leading 1,000 letters of w, $\alpha(\mathrm{nar})$, and $\gamma(\mathrm{nar})$).

Roadmap. Section 2 establishes exact definitions matching `lean/XYXAvoid/Basic.lean`, walks through the 14-letter occurrence example (Figure 2), and defines $w = \sigma^\omega(0)$ alongside the companion Narayana images. Section 3 proves the 217-class orbit census, the antichain dichotomy, the $\leq$ 8-arc reduction, the lower-bound bridge (Theorem 1), bifix coding injectivity, and two-move descent soundness (Lemma 6). Section 4 documents the five maximal exhaustive trees (Proposition 7), the independent Cassaigne-type descent certificates proving Theorem 2, our exact 1,000-letter prefix scans, and the honest methodological limitation. Section 5 presents the machine-generated Lean 4 + Mathlib Reader's Trust Boundary. Sections 6 and 7 cover prior work, open questions, data availability, and AI disclosure.

2. DEFINITIONS, DIGRAPHS, AND WORKED EXAMPLE

All definitions below match `lean/XYXAvoid/Basic.lean` and `lean/XYXAvoid/Morphisms.lean` verbatim.

Factors, *xyx*-formulas, and 2-avoidability. For a finite alphabet Σ, a finite word $u \in \Sigma^*$ is a *contiguous factor* of $w \in \Sigma^*$ (`IsFactor u w`) if $w = pus$ for some $p, s \in \Sigma^*$. An infinite word $W : \mathbb{N} \to \Sigma$ has length-n prefix $\mathrm{prefixOf}(W, n) = W(0)W(1)\cdots W(n-1)$, and u is a factor of W (`IsFactorInf u W`) if u is a factor of $\mathrm{prefixOf}(W, n)$ for some $n \in \mathbb{N}$.

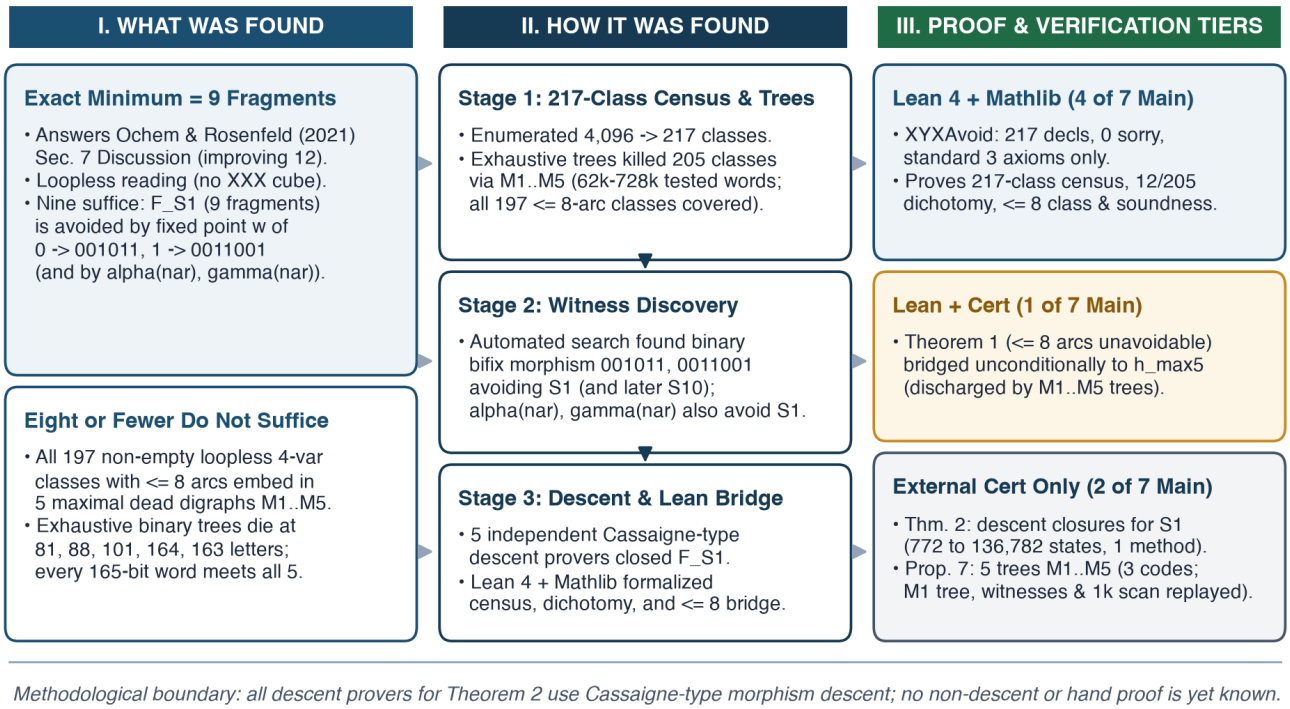


Figure 1. Discovery and proof architecture for the fewest-fragments theorem. Column I summarizes the exact minimum fragment count (9 suffice via F_{S_1} avoided by $w = \sigma^\omega(0)$, $\alpha(\mathrm{nar})$, and $\gamma(\mathrm{nar})$; all 197 non-empty classes with at most 8 arcs embed in $M_1, \ldots, M_5$ and die by length 165). Column II traces the three-stage computational discovery path. Column III separates the 4 lemmas proved unconditionally inside the Lean 4 + Mathlib kernel, Theorem 1 bridged to the external tree certificates, and the 2 external certificate results (Theorem 2 and Proposition 7).

A *loopless digraph* $G = (V, E)$ consists of a finite vertex set V and an irreflexive arc relation $E \subseteq V \times V$ with $(v, v) \notin E$ for all $v \in V$. Following Ochem and Rosenfeld [1] (Section 5), its associated *xyx-formula* F_G has one fragment XYX for each arc $(X, Y) \in E$. An *occurrence* of F_G in a finite word w (`OccursDigraph G w`) or infinite word W (`OccursDigraphInf G W`) is a map $h : V \to \Sigma^+$ assigning a *non-empty* word to each vertex (distinct variables are permitted to receive the same word) such that $h(X)h(Y)h(X)$ is a factor of w (respectively W) for every $(X, Y) \in E$. Equivalently, writing $P(w) = \{(x, y) \in (\Sigma^+)^2 : xyx \text{ is a factor of } w\}$, an occurrence of F_G in w is a digraph homomorphism $h : G \to P(w)$. The formula F_G is 2-*avoidable* (`Avoidable2 G`) if some infinite binary word $W : \mathbb{N} \to \{0, 1\}$ has no occurrence of F_G.

Structural properties. Three basic facts, proved in `Basic.lean`, govern the problem:

1. *Cube-freeness.* Any non-empty cube u^3 in w gives an occurrence of every F_G via the constant map $h(v) = u$ (`occursDigraph_of_cube`). Thus every binary word avoiding any F_G is cube-free.
2. *Bit-flip and reversal invariance.* Complementing bits $0 \leftrightarrow 1$ (proved in `Basic.lean`) or reversing a word preserves occurrences of every F_G, because $(xyx)^R = x^R y^R x^R$. By contrast, reversing all arcs of G is *not* a symmetry, since XYX is asymmetric between X and Y.
3. *Homomorphism monotonicity* (*Lemma 18 of* [1]). If $\varphi : H \to G$ is a digraph homomorphism (mapping every arc of H to an arc of G), then any occurrence h of F_G pulls back to the occurrence $h \circ \varphi$ of F_H (`occurs_of_hom`). Hence 2-avoidability passes upward to supergraphs (`avoidable2_of_emb`), while unavoidability passes downward to subgraphs.

As a hand-checkable warm-up, the single-fragment formula ABA (one arc $A \to B$) is avoided by 0011, but has only $2, 4, 4, 2$ avoiding binary words of lengths $1, 2, 3, 4$ (26 words tested in `recompute.py`) and occurs in every binary word of length 5 (for example 00110 realizes $A = 0, B = 11$).

Worked 14-letter occurrence. Figure 2 contrasts two 9-fragment formulas:

$$F_{G_5} = ABA.ACA.ADA.BAB.BCB.CAC.CDC.DAD.DBD,$$
$$F_{S_1} = ABA.BAB.ACA.CAC.BCB.CBC.ADA.BDB.CDC.$$

In the 14-letter binary word `01101001101101` (the leading 14 letters of $\alpha(\mathrm{nar})$), setting $A = 1$, $B = 0$, $C = 10$, $D = 01$ places all 9 fragments of F_{G_5} as contiguous factors: $DAD = 01101$ (letters 1–5), $ACA = 1101$ (2–5), $ABA = 101$ (3–5), $BAB = 010$ (4–6), $BCB = 0100$ (4–7), $DBD = 01001$ (4–8), $CDC = 100110$ (5–10), $ADA = 1011$ (9–12), and $CAC = 10110$ (9–13). Thus F_{G_5} already occurs in the leading 13 letters (verified in `recompute.py`). By contrast, an exact unbounded-block-length search confirms that F_{S_1} has zero occurrences in `01101001101101`.

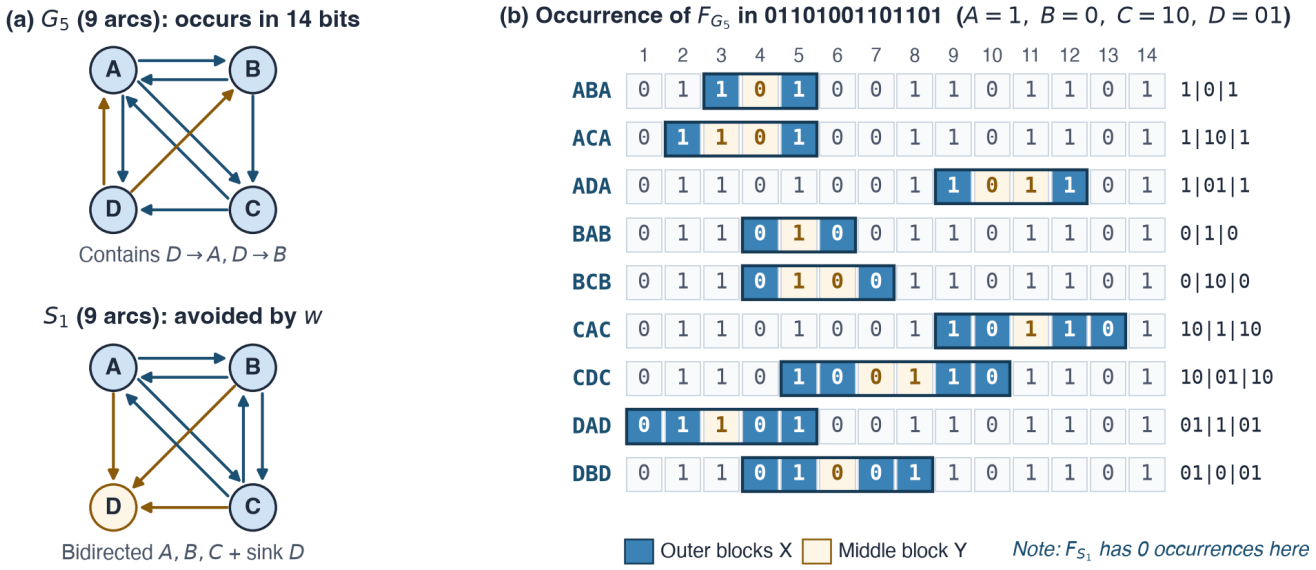


Figure 2. Worked 9-fragment contrast. (a) The 9-arc digraphs G_5 (top, containing $D \to A$ and $D \to B$) and S_1 (bottom, a bidirected triangle on A, B, C with D as a pure sink receiving arcs from A, B, C). (b) An explicit occurrence of F_{G_5} in the 14-letter binary word `01101001101101` under $A = 1, B = 0, C = 10, D = 01$: each row displays the same 14 letters with the outer blocks X in blue and the middle block Y in amber. The same 14-letter word contains zero occurrences of F_{S_1}.

The avoiding infinite words. A non-erasing endomorphism $f : \Sigma \to \Sigma^+$ with $f(0) = 0t$ ($t \neq \varepsilon$) generates a unique infinite fixed point $f^\omega(0) : \mathbb{N} \to \Sigma$ (`fixedPoint`). Our primary witness in Theorem 2 is the binary fixed point $w = \sigma^\omega(0)$ of

$$\sigma : 0 \mapsto 001011, \qquad 1 \mapsto 0011001.$$

Its iterates $\sigma^k(0)$ have lengths $L_k = 1, 6, 39, 252, 1629, 10530$ for $k = 0, \ldots, 5$, satisfying the exact linear recurrence $L_{k+2} = 6L_{k+1} + 3L_k$ (since the incidence matrix $\left(\begin{smallmatrix} 3 & 4 \\ 3 & 3 \end{smallmatrix}\right)$ has trace 6 and determinant -3). Its leading 39 letters are

`001011001011001100100101100110011001101.`

For comparison and independent redundancy, we also consider *Narayana's word* [6, 3] $\mathrm{nar} = f_{\mathrm{nar}}^\omega(0) = 0120010120120\ldots$, generated by $f_{\mathrm{nar}} : 0 \mapsto 01, 1 \mapsto 2, 2 \mapsto 0$, with the two binary morphisms (`Morphisms.lean`)

$$\alpha : 0 \mapsto 0, \quad 1 \mapsto 11, \quad 2 \mapsto 01,$$
$$\gamma : 0 \mapsto 001, \quad 1 \mapsto 1001, \quad 2 \mapsto 101,$$

whose images $\alpha(\mathrm{nar}) = 0110100110110101101\ldots$ and $\gamma(\mathrm{nar}) = 00110011010010011001\ldots$ also avoid F_{S_1}.

3. PROOF OF THE LOWER BOUND AND ALGEBRAIC DESCENT FRAMEWORK

We now prove Lemmas 3–6 and derive Theorem 1 from the five maximal tree certificates of Proposition 7. Every combinatorial step in this section is formalized unconditionally in `lean/XYXAvoid/Census.lean` and `Descent.lean`.

Orbit partition (Proof of Lemma 3). Fix the vertex set $V_4 = \{0, 1, 2, 3\}$ (written A, B, C, D in formulas). There are $4 \cdot 3 = 12$ ordered pairs (u, v) of distinct vertices, indexed bijectively by Fin 12 (`arcOfIdx`, `idxOfArc`). Every loopless digraph G on V_4 is encoded faithfully by a 12-bit mask $m \in \mathrm{Mask}_4$ ($0 \le m \le 4095$, `exists_mask_iso`); digraphs on fewer than 4 vertices are represented by adding isolated vertices, which contribute no fragments to F_G.

The symmetric group S_4 of 24 vertex permutations acts on Mask_4 by relabelling arcs (`permMask`), and we define the canonical representative $\mathrm{canon}_4(m) = \min_{\pi \in S_4} \pi \cdot m$. To verify the orbit partition inside Lean's pure kernel without relying on compiled reflection, `Census.lean` exhibits the explicit strictly sorted list `canon218Nat` of 218 canonical integers and checks by kernel reduction that:

1. every $c \in$ `canon218Nat` satisfies $\mathrm{canon}_4(c) = c$, making the orbits $\mathrm{Orb}(c) = \{\pi \cdot c : \pi \in S_4\}$ pairwise disjoint (`orbitFinset_disjoint`);
2. the exact orbit sizes sum to the full space (`canon218Nat_orbit_sum`):

$$\sum_{c \in \mathtt{canon218Nat}} |\mathrm{Orb}(c)| = 4096 = |\mathrm{Mask}_4|.$$

Hence $\bigcup_c \mathrm{Orb}(c) = \mathrm{Mask}_4$ (`biUnion_orbitFinset_eq_univ`), proving that `allClasses4` has cardinality 218. Removing the empty mask 0 yields exact cardinality 217 for `nonemptyClasses4` (`card_classes_4`), proving Lemma 3. □

Dichotomy and $\le$ 8-arc reduction (Proof of Lemmas 4 and 5). For $m_1, m_2 \in \mathrm{Mask}_4$, write $m_1 \preceq m_2$ (`embedsMask`) if some $\pi \in S_4$ maps every arc of m_1 into m_2; in `Census.lean`, we prove that $m_1 \preceq m_2$ holds if and only if there is a vertex-bijective subgraph embedding DigraphEmb from maskToDigraph(m_1) to maskToDigraph(m_2).

Table 1 lists the 5 maximal unavoidable digraphs $M_1, \ldots, M_5$ together with our 9-fragment witness S_1 (the full

upward boundary consists of 7 minimal digraphs: 6 with 9 arcs—$S_1, G_2, G_3, G_4, G_5, G_6$—and 1 with 10 arcs, S_{10}). Direct kernel checks prove that the 7 minimal masks form a strict antichain (`min7_antichain`), the 5 maximal masks form a strict antichain (`max5_antichain`), and no minimal mask embeds into any maximal mask (`no_min7_embeds_max5`). By transitivity of $\preceq$, no mask lies simultaneously above a minimal class and below a maximal class (`not_above_and_below`).

For exhaustiveness, `Census.lean` supplies two disjoint strictly sorted witness lists: `avoidWits12` (length 12, giving for each class c an index $i \in \mathrm{Fin}\,7$ and $\pi \in S_4$ with $\pi \cdot \min_7(i) \subseteq c$) and `unavoidWits205` (length 205, giving an index $j \in \mathrm{Fin}\,5$ and $\pi \in S_4$ with $\pi \cdot c \subseteq \max_5(j)$). A linear sorted-merge check (`mergeCheck_sound`) proves that the 217 non-empty canonical masks equal the disjoint union of these 12 and 205 masks, while the empty mask 0 embeds in M_1. Since every $m \in \mathrm{Mask}_4$ is S_4-equivalent to $\mathrm{canon}_4(m)$, every $m \in \mathrm{Mask}_4$ lies exclusively above one of the 7 minimal classes or below one of $M_1, \dots, M_5$ (`mask4_dichotomy`), proving Lemma 4.

Inspecting the 12 canonical masks in `avoidWits12` shows that every class in `avoidableClasses4` has at least 9 arcs (`arcCount_ge_9_of_avoidable`; Table 2). Therefore, if $c \in$ `nonemptyClasses4` has $\mathrm{arcCount}(c) \le 8$, then c cannot satisfy `isAboveMin7`, so $c \in$ `unavoidableClasses4` (`all_le8_arcs_unavoidable`), proving Lemma 5. In particular, all 197 non-empty classes with at most 8 arcs embed into at least one of $M_1, \dots, M_5$. □

Lower-bound bridge (Proof of Theorem 1). Let $m \in \mathrm{Mask}_4$ satisfy $\mathrm{arcCount}(m) \le 8$, and assume `h_max5`: none of $M_1, \dots, M_5$ is 2-avoidable. Suppose for contradiction that $\mathrm{maskToDigraph}(m)$ is 2-avoidable. By Lemma 4 (`mask4_dichotomy`), either m lies above some minimal class $\min_7(i)$ or m embeds into some maximal class M_j.

- In the former case, $\mathrm{canon}_4(m)$ is non-empty (since the empty mask is not above $\min_7$), lies in `avoidableClasses4`, and satisfies $\mathrm{arcCount}(\mathrm{canon}_4(m)) = \mathrm{arcCount}(m) \le 8$ (because S_4 permutes the 12 arc positions bijectively), contradicting `arcCount_ge_9_of_avoidable`.
- In the latter case, there exists a subgraph embedding $\mathrm{maskToDigraph}(m) \hookrightarrow M_j$, so Lemma 18 of [1] (`avoidable2_of_emb`) implies that M_j is 2-avoidable, contradicting `h_max5`.

This completes the proof of `r10_2_1_le8_unavoidable`. Moreover, since every binary word of length 165 contains an occurrence of each M_j (Proposition 7), pulling back along the embedding $m \hookrightarrow M_j$ (`occurs_of_hom`) shows that every loopless 4-variable formula with at most 8 fragments occurs in every binary word of length 165. □

Coding injectivity and descent soundness (Proof of Lemma 6). In `Descent.lean`, we prove unconditionally that every prefix code and every suffix code induces an injective morphism on all finite words. Observe that our primary morphism $\sigma : 0 \mapsto 001011,\ 1 \mapsto 0011001$ is a *bifix code* (both a prefix code and a suffix code): its two codewords differ at position 4 (0 versus 1) and at their penultimate bit (1 versus 0), so neither codeword is a prefix or a suffix of the other. Likewise, `Descent.lean` proves that f_{nar} and α are suffix codes and γ is a prefix code, hence all three induce injective morphisms on finite words. Global injectivity guarantees unique parsing of sufficiently long factors back to preimages under σ (or $\alpha, \gamma, f_{\mathrm{nar}}$) up to bounded boundary pieces.

For Lemma 6 (`freeze_or_desub_soundness`), equip $\mathbb{N} \times \mathbb{N}$ with the lexicographic order $\prec_{\mathrm{lex}}$, which is well-founded (`WellFounded.prod_lex`). By well-founded induction on $p = (\mathrm{freeVars}(s), \mu)$, no $s \in C$ satisfies $\mathrm{Sat}(s, \mu)$: if $\mathrm{Sat}(s, \mu)$ held, then either a freeze step yields (s', μ') with $\mathrm{freeVars}(s') < \mathrm{freeVars}(s)$, or a desubstitution step yields (s', μ') with $\mathrm{freeVars}(s') = \mathrm{freeVars}(s)$ and $\mu' < \mu$; in either case $(\mathrm{freeVars}(s'), \mu') \prec_{\mathrm{lex}} (\mathrm{freeVars}(s), \mu)$, contradicting the induction hypothesis. Applying this to $s_0 \in C$ rules out every occurrence h of F_G in W. □

4. COMPUTATIONAL CERTIFICATES AND INDEPENDENT VERIFICATION

We now detail the deterministic computational certificates discharging Proposition 7 and Theorem 2, together with our standalone exact verification in `recompute.py`.

Lower-bound trees for $M_1, \dots, M_5$ (Proposition 7). To certify that a digraph G is not 2-avoidable, an unpruned depth-first search grows binary words one bit at a time from the empty word, maintaining the exact relation $P(w) = \{(x, y) : xyx \text{ is a factor of } w\}$ and pruning any word w admitting a homomorphism $G \to P(w)$ through a newly added suffix pair xyx. When the search tree is finite of depth L, every binary word of length $L + 1$ contains an occurrence of F_G.

For all 5 maximal unavoidable digraphs $M_1, \dots, M_5$ (Figure 3, top), three independent C/Python implementations agreed identically on every maximum avoiding length, avoiding-word count, and total tested-word count (Table 1):

- M_1 (10 arcs): max length 81, 31,058 avoiding words, 62,118 tested (peak 752 at $n = 41$, replayed in full by `recompute.py`);
- M_2 (9 arcs): max length 88, 61,494 avoiding words, 122,990 tested;
- M_3 (9 arcs): max length 101, 363,932 avoiding words, 727,866 tested;
- M_4 (9 arcs): max length 164, 322,872 avoiding words, 645,746 tested;
- M_5 (9 arcs): max length 163, 318,378 avoiding words, 636,758 tested.

In addition, an independent blind re-review dumped every leaf of all five search trees and verified with a plain-string checker that the leaves form a complete binary prefix code with an explicit occurrence of M_j at every leaf. Our standalone script `recompute.py` independently verifies all 5 logged maximum-length witnesses (confirming each avoids M_j while both one-bit extensions $w0$ and $w1$ contain occurrences) and replays the entire unpruned M_1 binary tree to depth 81 (62,118 words tested, Figure 3(b)). Because M_4 has maximum depth 164, every binary word of length 165

Table 1. The 9-fragment witness digraph S_1 and the 5 maximal unavoidable digraphs $M_1, \ldots, M_5$ governing the lower bound, with exact 12-bit Lean masks, xyx-formulas, and boundary certificate metrics.

Class	Arcs	Mask	xyx-formula ($A=0, B=1, C=2, D=3$)	Role	Witness / Exhaustive tree bound
S_1	9	511	*ABA.ACA.ADA.BAB.BCB.BDB.CAC.CBC.CDC*	9-frag. witness	$w = \sigma^\omega(0)$, α(nar), γ(nar)
M_1	10	1791	*ABA.ACA.ADA.BAB.BCB.BDB.CAC.CBC.DAD.DBD*	Max. dead	max len 81 (31,058 avoid., 62,118 tested)
M_2	9	1407	*ABA.ACA.ADA.BAB.BCB.BDB.CAC.CDC.DBD*	Max. dead	max len 88 (61,494 avoid., 122,990 tested)
M_3	9	2431	*ABA.ACA.ADA.BAB.BCB.BDB.CAC.CDC.DCD*	Max. dead	max len 101 (363,932 avoid., 727,866 tested)
M_4	9	3423	*ABA.ACA.ADA.BAB.BCB.CAC.CDC.DBD.DCD*	Max. dead	max len 164 (322,872 avoid., 645,746 tested)
M_5	9	3487	*ABA.ACA.ADA.BAB.BCB.CBC.CDC.DBD.DCD*	Max. dead	max len 163 (318,378 avoid., 636,758 tested)

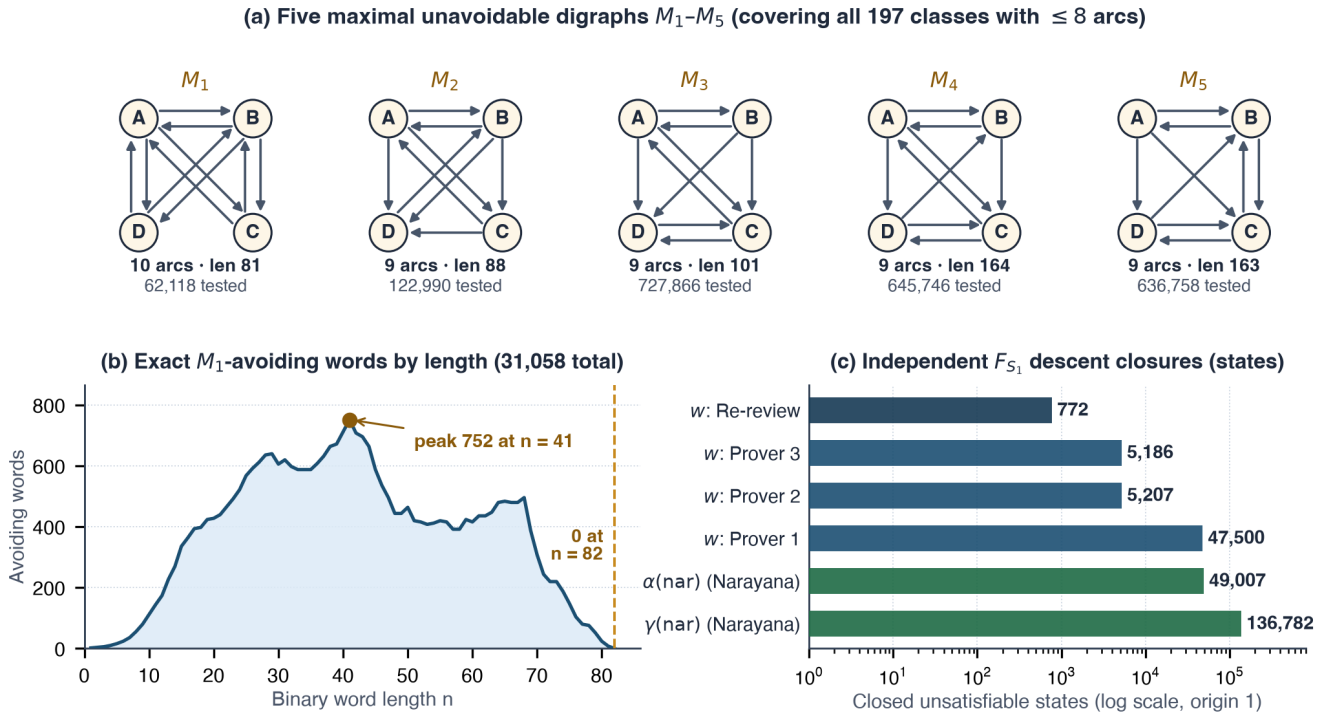


Figure 3. Computational certificates for the lower and upper bounds. Top: the 5 maximal unavoidable digraphs $M_1, \ldots, M_5$ covering all 197 classes with at most 8 arcs, with their exact maximum avoiding lengths (81 to 164) and tested word counts (62,118 to 727,866). (b) Exact number of binary words of length n avoiding M_1 (replayed in full by `recompute.py`: 31,058 avoiding words total, peaking at 752 for $n = 41$ and reaching 0 at $n = 82$). (c) Closed unsatisfiable state counts (log scale anchored at 1) across independent Cassaigne-type descent provers for F_{S_1} on $w = \sigma^\omega(0)$, α(nar), and γ(nar).

meets all five maximal formulas, and hence every loopless 4-variable formula with at most 8 fragments.

Upper-bound descent certificates for F_{S_1} (Theorem 2). To prove that an infinite morphic word avoids F_{S_1}, the provers apply Cassaigne's descent method as used by Ochem and Rosenfeld [1] for Theorem 23. Suppose $w = \sigma^\omega(0)$ contains an occurrence h of F_{S_1}. Each variable image $h(v)$ is either *short* (below a fixed synchronization threshold, in which case we branch over all factors of w of that length, reducing the number of free variables in a *freeze* move) or *long* (in which case unique parsing of the bifix code σ lifts $h(v)$ to a shorter σ-preimage in w up to bounded boundary pieces, strictly reducing the total variable length in a *desubstitution* move). By Lemma 6, closing the reachable state set with zero satisfiable states proves that w avoids F_{S_1}.

Table 3 and Figure 3(c) summarize the independent descent verifications. Five independent programs proved that $w = \sigma^\omega(0)$ avoids F_{S_1}: three separate initial implementations closing in 47,500, 5,207, and 5,186 states, followed by two separate blind re-review implementations written from scratch closing in 772 states via a compact desubstitution-closure formulation. Independently, the same formula F_{S_1} was proved avoided by two binary images of Narayana's word: α(nar) (49,007 states) and γ(nar) (136,782 states). Controls verified that the descent provers reproduced Theorem 23 of [1] (in 272 states) and detected real occurrences with zero false avoidances across 23 test formulas known to occur.

Exact 1,000-letter non-descent prefix scans. As a direct check independent of any descent code, `recompute.py` enumerates every non-empty pair (x, y) with xyx a factor of the leading 1,000 letters of each witness word (with no bound

Table 2. Exact census of all 217 non-empty isomorphism classes of loopless digraphs on 4 vertices by arc (fragment) count.

Arcs	Total classes	Below M_1–M_5	Above $\min_7$
1	1	1	0
2	5	5	0
3	13	13	0
4	27	27	0
5	38	38	0
6	48	48	0
7	38	38	0
8	27	27	0
≤ 8 subtotal	197	197	0
9	13	7	6
10	5	1	4
11	1	0	1
12	1	0	1
Total	217	205	12

Table 3. Independent Cassaigne-type descent state counts proving Theorem 2 (F_{S_1} avoidance; all implementations use one underlying descent method).

Witness	Prover	States	Notes
$w = \sigma^\omega(0)$	Prover 1	47,500	Initial prover
$w = \sigma^\omega(0)$	Prover 2	5,207	Independent rewrite
$w = \sigma^\omega(0)$	Prover 3	5,186	Third prover
$w = \sigma^\omega(0)$	Re-review (2 codes)	772	Compact closure
α(nar)	Narayana prover	49,007	Also G_2, G_3, G_4, G_6
γ(nar)	Narayana prover	136,782	Also G_6, S_{10}

on block length) and searches exhaustively for any homomorphism $S_1 \to P(w[:1000])$:

- $w = \sigma^\omega(0)$: 982,223 distinct xyx-pairs, 0 occurrences of F_{S_1} (and 0 occurrences of the 10-fragment formula $F_{S_{10}}$), whereas the other five 9-arc minimal classes G_2, G_3, G_4, G_5, G_6 occur in w within 22, 22, 22, 22, and 141 letters, respectively;
- α(nar): 1,084,177 distinct xyx-pairs, 0 occurrences of F_{S_1};
- γ(nar): 1,043,908 distinct xyx-pairs, 0 occurrences of F_{S_1}.

Finite prefix scans provide strong consistency evidence but do not prove infinite-word avoidance; the proof of Theorem 2 rests on the descent closures of Table 3.

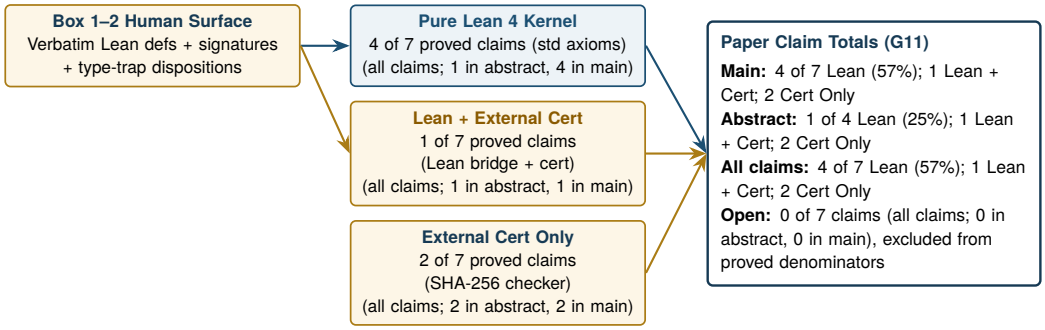
Honest methodological limitation. Although five independent agent programs verified that $w = \sigma^\omega(0)$ avoids F_{S_1} and two Narayana images provide alternative witnesses, every program implements the same overarching proof method (Cassaigne-type morphism descent), and every program and checker was written by AI agents. No proof by hand and no verification by an independent method (such as a Walnut automatic-sequence decision procedure) has yet been produced.

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 4d34d065283c (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:** 2 SHA-256-pinned external artifact(s) plus any claims marked Written or Unsettled in Table 5 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: 217/217 public theorems (source-level approx) · WARN: Fidelity: 5/5 exact (draft; auditor/formalizer relationship unknown) · WARN: Witnesses: pending gate replay · WARN: Human referee: pending

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

XYXAvoid.IsFactorInf (XYXAvoid/Basic.lean:163-164) — Section 2 (contiguous factor of an infinite word W : N -> alpha)

Context binders in scope: `variable {α β : Type*}`

```
def IsFactorInf (u : List α) (W : ℕ → α) : Prop :=
  ∃ i : ℕ, ∀ j (hj : j < u.length), W (i + j) = u[j]
```

XYXAvoid.LooplessDigraph (XYXAvoid/Basic.lean:207-209) — Section 2 (irreflexive directed graph excluding self-loops XXX)

```
structure LooplessDigraph (V : Type*) where
  adj : V → V → Prop
  loopless : ∀ v : V, ¬ adj v v
```

XYXAvoid.OccursDigraphInf (XYXAvoid/Basic.lean:258-259) — Section 2 (Ochem-Rosenfeld XYX-formula occurrence in an infinite word)

```
def OccursDigraphInf {V α : Type*} (G : LooplessDigraph V) (W : ℕ → α) : Prop :=
  ∃ h : V → List α, (∀ v, h v ≠ []) ∧ ∀ u v, G.adj u v → IsFactorInf (h u ++ h v ++ h
    ↪ u) W
```

XYXAvoid.AvoidsDigraphInf (XYXAvoid/Basic.lean:266-267) — Section 2 (negation of OccursDigraphInf)

```
def AvoidsDigraphInf {V α : Type*} (G : LooplessDigraph V) (W : ℕ → α) : Prop :=
  ¬ OccursDigraphInf G W
```

XYXAvoid.Avoidable2 (XYXAvoid/Basic.lean:270-271) — Section 2 (existence of an infinite binary word avoiding F_G)

```
def Avoidable2 {V : Type*} (G : LooplessDigraph V) : Prop :=
  ∃ W : ℕ → Bool, AvoidsDigraphInf G W
```

XYXAvoid.Mask4 (XYXAvoid/Census.lean:166-166) — Section 3 (12-bit representation Fin 4096 of 4-vertex loopless digraphs)

```
abbrev Mask4 := Fin 4096
```

XYXAvoid.maskToDigraph (XYXAvoid/Census.lean:377-379) — Section 3 (decoding a 12-bit mask to LooplessDigraph (Fin 4))

```
def maskToDigraph (m : Mask4) : LooplessDigraph (Fin 4) where
  adj u v := u ≠ v ∧ hasBit m.val (idxOfArc u v) = true
  loopless _ h := h.1 rfl
```

XYXAvoid.max5Graph (XYXAvoid/Census.lean:1249-1249) — Section 3 & Table 1 (the 5 maximal unavoidable digraphs M1..M5)

```
def max5Graph (j : Fin 5) : LooplessDigraph (Fin 4) := maskToDigraph (max5Masks j)
```

XYXAvoid.embedsMask (XYXAvoid/Census.lean:1087-1088) — Section 3 (subgraph embedding on Mask4 up to S_4 vertex relabelling)

```
def embedsMask (m₁ m₂ : Mask4) : Prop :=
  ∃ k : Fin 24, subMask (permMaskNat k m₁.val) m₂.val = true
```

XYXAvoid.isAboveMin7 (XYXAvoid/Census.lean:1337-1338) — Section 3 (upward closure of min7Masks under embedsMask)

```
def isAboveMin7 (c : Mask4) : Prop :=
  ∃ i : Fin 7, embedsMask (min7Masks i) c
```

XYXAvoid.isBelowMax5 (XYXAvoid/Census.lean:1341-1342) — Section 3 (downward closure of max5Masks under embedsMask)

```
def isBelowMax5 (c : Mask4) : Prop :=
  ∃ j : Fin 5, embedsMask c (max5Masks j)
```

XYXAvoid.nonemptyClasses4 (XYXAvoid/Census.lean:1006-1007) — Section 3 (the 217 non-empty canonical S_4 isomorphism classes)

```
def nonemptyClasses4 : Finset Mask4 :=
  allClasses4.filter (fun m => m.val ≠ 0)
```

XYXAvoid.unavoidableClasses4 (XYXAvoid/Census.lean:1425-1426) — Section 3 (the 205 non-empty classes outside the upward closure of min7)

```
def unavoidableClasses4 : Finset Mask4 :=
  nonemptyClasses4.filter (fun c => ¬ isAboveMin7 c)
```

XYXAvoid.arcCount (XYXAvoid/Census.lean:408-409) — Section 3 (number of arcs in a 12-bit mask)

```
def arcCount (m : ℕ) : ℕ :=
  (List.finRange 12).countP (fun i => hasBit m i)
```

XYXAvoid.occMeasure (XYXAvoid/Descent.lean:425-426) — Section 1 & 3 (total variable-word length sum_v |h(v)|)

```
def occMeasure {V α : Type*} [Fintype V] (h : V → List α) : ℕ :=
  ∑ v : V, (h v).length
```

Supporting Semantic Closure Index (non-Box-1 helpers): XYXAvoid.S1_mask (XYXAvoid/Census.lean), XYXAvoid.G2_mask (XYXAvoid/Census.lean), XYXAvoid.G3_mask (XYXAvoid/Census.lean), XYXAvoid.G4_mask (XYXAvoid/Census.lean), XYXAvoid.G5_mask (XYXAvoid/Census.lean), XYXAvoid.G6_mask (XYXAvoid/Census.lean), XYXAvoid.S10_mask (XYXAvoid/Census.lean), XYXAvoid.M1_mask (XYXAvoid/Census.lean), XYXAvoid.M2_mask (XYXAvoid/Census.lean), XYXAvoid.M3_mask (XYXAvoid/Census.lean), XYXAvoid.M4_mask (XYXAvoid/Census.lean), XYXAvoid.M5_mask (XYXAvoid/Census.lean), XYXAvoid.min7Masks (XYXAvoid/Census.lean), XYXAvoid.max5Masks (XYXAvoid/Census.lean), XYXAvoid.hasBit (XYXAvoid/Census.lean), XYXAvoid.pow2 (XYXAvoid/Census.lean), XYXAvoid.arcOfIdx (XYXAvoid/Census.lean), XYXAvoid.idxOfArc (XYXAvoid/Census.lean), XYXAvoid.subMask (XYXAvoid/Census.lean), XYXAvoid.packBits (XYXAvoid/Census.lean), XYXAvoid.mkPerm4 (XYXAvoid/Census.lean), XYXAvoid.perm4Fun (XYXAvoid/Census.lean), XYXAvoid.perm4InvIdx (XYXAvoid/Census.lean), XYXAvoid.permArc (XYXAvoid/Census.lean), XYXAvoid.permArcInv (XYXAvoid/Census.lean), XYXAvoid.permMaskNat (XYXAvoid/Census.lean), XYXAvoid.allClasses4 (XYXAvoid/Census.lean), XYXAvoid.canon4 (XYXAvoid/Census.lean).

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

XYXAvoid.r10_2_1_le8_unavoidable (XYXAvoid/Census.lean:1568-1610) src:9bd2551cce52f3

```
theorem r10_2_1_le8_unavoidable
    (h_max5 : ∀ j : Fin 5, ¬ Avoidable2 (max5Graph j))
    (m : Mask4) (h8 : arcCount m.val ≤ 8) :
    ¬ Avoidable2 (maskToDigraph m)
```

Assumptions: Explicit external certificate hypothesis h_max5 (5 maximal unavoidable digraphs M1..M5 via exhaustive binary trees), applied to any 12-bit mask m : Mask4 with arcCount m.val <= 8. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.card_classes_4 (XYXAvoid/Census.lean:1012-1025) src:0dad8dd92dcfe7

```
theorem card_classes_4 : nonemptyClasses4.card = 217
```

Assumptions: Closed equality nonemptyClasses4.card = 217 over all 4,096 loopless 4-vertex masks; companion card_all_classes_4 proves 218 with the empty digraph. Applicability: Unconditional (closed statement; no outer premise witness needed).

XYXAvoid.mask4_dichotomy (XYXAvoid/Census.lean:1392-1418) src:71ecf9a03087e2

```
theorem mask4_dichotomy (m : Mask4) : isAboveMin7 m ∨ isBelowMax5 m
```

Assumptions: Universal over all 4,096 masks m : Mask4 (exclusivity proved separately by XYXAvoid.not_above_and_below). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.all_le8_arcs_unavoidable (XYXAvoid/Census.lean:1493-1498) src:1230ff41a54192

```
theorem all_le8_arcs_unavoidable {c : Mask4} (hc : c ∈ nonemptyClasses4) (h8 :
  ↪ arcCount c.val ≤ 8) :
    c ∈ unavoidableClasses4
```

Assumptions: Universal over all non-empty canonical classes c in nonemptyClasses4 with arcCount c.val <= 8. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.freeze_or_desub_soundness (XYXAvoid/Descent.lean:466-494) src:ca13a7e72b416d

```
theorem freeze_or_desub_soundness {V α State : Type*} [Fintype V]
    (G : LooplessDigraph V) (W : ℕ → α)
    (freeVars : State → ℕ) (Sat : State → ℕ → Prop) (Cert : Set State) (s₀ : State)
    (hstart : s₀ ∈ Cert)
    (hinit : ∀ h : V → List α, (∀ v, h v ≠ []) →
      (∀ u v, G.adj u v → IsFactorInf (h u ++ h v ++ h u) W) →
      Sat s₀ (occMeasure h))
    (hclosed : ∀ s ∈ Cert, ∀ μ : ℕ, Sat s μ →
      (∃ s' ∈ Cert, freeVars s' < freeVars s ∧ ∃ μ' : ℕ, Sat s' μ') ∨
      (∃ s' ∈ Cert, freeVars s' = freeVars s ∧ ∃ μ' < μ, Sat s' μ')) :
    AvoidsDigraphInf G W
```

Assumptions: Any finite vertex type V, alphabet alpha, state type State, measure occMeasure, and state set Cert containing s0 closed under freeze or desubstitution moves. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

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

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`hasBit`	`nat_div`	`/ pow2`	intended_convention	Exact binary shift m / 2^i where pow2 i = 2^i >= 1 for all i : Fin 12 (verified by pow 2_eq).
`hasBit`	`nat_mod`	`% 2`	intended_convention	Standard parity extraction % 2 with constant nonzero modulus 2.
`instDecidableEmbedsMask`	`local_instance_or_notation`	`instDecidableEmbedsMask`	intended_convention	Standard Decidable instance inst DecidableEmbedsMask; carries no custom axioms or notation overrides.
`instDecidableIsAboveMin7`	`local_instance_or_notation`	`instDecidableIsAboveMin7`	intended_convention	Standard Decidable instance inst DecidableIsAboveMin7; carries no custom axioms or notation overrides.
`instDecidableIsBelowMax5`	`local_instance_or_notation`	`instDecidableIsBelowMax5`	intended_convention	Standard Decidable instance inst DecidableIsBelowMax5; carries no custom axioms or notation 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 5. 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 12-bit loopless 4-vertex digraph with at most 8 arcs defines a non-2-avoidable xyx-formula (under h_max 5)	Lean + Cert	`XYXAvoid.r10_2_1_le8_unavoidable` + recompute.py (c63b38f85a8f) [archived; replay pending]; census.py (f7b2f9b25803) [regenerable; replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib kernel proves all <= 8-arc masks embed into M1..M5; external binary trees (Prop. 7) discharge h_max5 (bound 165).
Thm. 2: Fixed point w of 0 -> 001011, 1 -> 0011001 (and alpha(nar), gamma(nar)) avoids the 9-fragment formula F_S1; min fragments = 9	Cert Only	recompute.py (c63b38f85a8f) [archived; replay pending]; census.py (f7b2f9b25803) [regenerable; replay pending]	External cert only; fidelity: n/a	Cassaigne-type morphism descent verified by 5 independent programs on w (772..47,500 states) and 2 Narayana images (49,007 and 136,782 states), all 1 method; barrier: descent state graphs not yet replayed inside Lean kernel.
Lem. 3: Exact cardinality 217 of non-empty S_4 isomorphism classes on 4 vertices (218 with empty)	Lean Kernel	`XYXAvoid.card_classes_4`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib orbit-partition proof (4,096 = sum of 218 disjoint S_4 orbit sizes).
Lem. 4: Every 12-bit loopless 4-vertex digraph lies exclusively above min7 (all >= 9 arcs) or below max5 (M1..M5)	Lean Kernel	`XYXAvoid.mask4_dichotomy`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib pure-kernel proof over all 4,096 masks (exclusivity by not_above_and_below).

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Lem. 5: Every non-empty canonical 4-vertex class with <= 8 arcs belongs to the 205-class unavoidable complement	Lean Kernel	`XYXAvoid.all_le8_arcs_unavoidable`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib pure-kernel proof (all 197 non-empty <= 8-arc classes lie outside the upward closure of min7).
Lem. 6: Soundness of two-move (freeze or desubstitute) lexicographic morphism descent for AvoidsDigraphInf	Lean Kernel	`XYXAvoid.freeze_or_desub_soundness`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib well-founded induction on Prod. Lex (<) (<) over N x N.
Prop. 7: Five exhaustive binary search trees for M1.. M5 terminate at lengths 81, 88, 101, 164, 163 (bound 165)	Cert Only	recompute.py (c63b38f85a8f) [archived; replay pending]; census.py (f7b2f9b25803) [regenerable; replay pending]	External cert only; fidelity: n/a	External unpruned binary trees confirmed by 3 independent programs plus prefix-code leaf checkers (M1 tree and all 5 maximal witnesses replayed in recompute.py; barrier: 62k-728k trees beyond Lean kernel).

[a]Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: 63daec387f11 | Lean mode: committed_receipt | Closure SHA-256: 92ed2e615a56 | Git anchor: 4d34d065283c | Axioms SHA-256: 76884cfcb660 | Fidelity SHA-256: fec24239bf8f.

6. PRIOR WORK AND OPEN QUESTIONS

Ochem and Rosenfeld [1] introduced *xyx*-formulas, established the digraph correspondence (Section 5) and homomorphism monotonicity (Lemma 18), proved $H = ABA.BAB.ACA.CBC$ is not 2-avoidable (Theorem 19), proved the complete 12-arc digraph on 4 vertices is 2-avoidable (Theorem 23) via Cassaigne's descent method, and posed the fewer-fragments question in Section 7 (Discussion). Narayana's word is classical [6, 3]; related formula families include two-variable formulas [2] [to check], circular formulas [4] [to check], and avoidance bases with reversal [5] [to check]. Two primary search logs passed `litsearch.py check` (exit 0; 29 + 16 = 45 keyed queries across arXiv, OEIS, zbMATH Open, OpenAlex, Semantic Scholar, MathOverflow, and Math Stack Exchange; 18 candidates), supplemented by two companion four-variable logs (45 queries, 14 candidates; 90 queries and 32 candidates total), manual Zenodo/DataCite/HAL/OSF checks, a CORE sweep (7 requests, 23 records), and a Google Scholar sweep (6 queries). Neither printing of [1] contains a 2-avoidable 4-variable formula with fewer than 12 fragments, and this result was not found in the searches logged. *Unsearched or unread:* Lothaire's three volumes, Cassaigne's 1994 and Clark's 2001 theses, WORDS/DLT proceedings (2022–2026) beyond indexed archives, private talks beyond Ochem's May 2024 slides, and full texts of [2, 4, 5] [to check].

A companion preprint classifies all 217 non-empty loopless 4-vertex classes (12 avoidable versus 205 unavoidable), and a separate study resolves the 5-vertex transitive tournament TT_5 (10 fragments) from Section 7 of [1]. Open directions: (i) a non-descent (e.g., Walnut) or hand proof of Theorem 2; (ii) direct Lean 4 kernel replay of the 772-state descent closure for w and prefix-code leaf certificates for $M_1, \ldots, M_5$; and (iii) the minimum fragment count on 5 variables (bounded above by 9 via S_1, or 10 among connected 5-vertex digraphs via TT_5).

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

All manuscript files, figure scripts, and outputs reside in `papers/xyx-fewest-fragments/draft1/`, and the Lean 4 + Mathlib library resides in `lean/XYXAvoid/` (217 audited public theorems, 0 `sorry`, standard 3 axioms only). Running `python3 papers/xyx-fewest-fragments/draft1/recompute.py` replays the 4,096 → 217 census and 197-class ≤ 8-arc reduction, the 14-letter occurrence of F_{G_5} and Section 1 snippet, bifix coding of σ, exact 1,000-letter scans of w, $\alpha(\text{nar})$, and $\gamma(\text{nar})$, all 5 maximal witnesses and 1-bit extensions, and the full unpruned M_1 tree (62,118 tested words). AI research agents built and directed by Tyler B. Hughes discovered the witness morphism, ran the trees and descent provers, built the Lean 4 + Mathlib formalization, generated the figures, and drafted this manuscript. No human mathematician has yet refereed the proof.

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

REFERENCES

(1) Ochem, P.; Rosenfeld, M. Avoidability of palindrome patterns. *Electron. J. Combin.* **2021**, *28* (1), #P1.4. DOI: 10.37236/9593; arXiv:2005.05505.

(2) Ochem, P.; Rosenfeld, M. Avoidability of formulas with two variables. *Electron. J. Combin.* **2017**, *24* (4), #P4.30. arXiv:1606.03955.

(3) Shallit, J. The Narayana morphism and related words. Preprint, **2025**. arXiv:2503.01026.

(4) Gamard, G.; Ochem, P.; Richomme, G.; Séébold, P. Avoidability of circular formulas. *Theoret. Comput. Sci.* **2018**, *726*, 1–4. arXiv:1610.04439.

(5) Currie, J. D.; Mol, L.; Rampersad, N. Avoidance bases for formulas with reversal. *Theoret. Comput. Sci.* **2018**, *738*, 25–41.

arXiv:1711.08025.

(6) OEIS Foundation Inc. Entry A105083: Trajectory of 1 under the morphism $1 \mapsto 12$, $2 \mapsto 3$, $3 \mapsto 1$. *The On-Line Encyclopedia of Integer Sequences*. https://oeis.org/A105083 (accessed November 20, 2026).