MATHDISCOVERIES
Mathematical Preprint

Working Draft

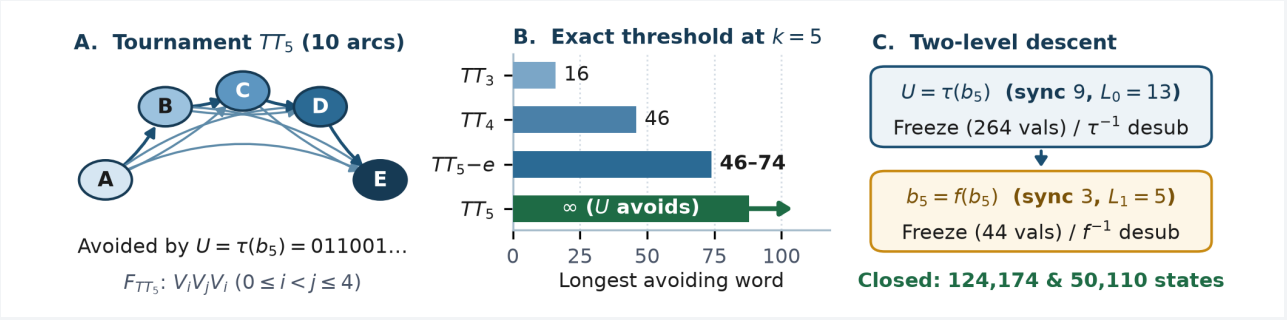
The Transitive Tournament on Five Vertices Is 2-Avoidable: xyx-Avoidability of TT_5

Tyler B. Hughes*,†

†Independent Researcher, United States

Supporting Information: `recompute.py`, 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: The xyx-formula F_G of a loopless digraph G has one fragment $V_iV_jV_i$ per arc $i \to j$, and is 2-avoidable when an infinite binary word avoids every non-empty assignment h placing all $h(i)h(j)h(i)$ as factors. Ochem and Rosenfeld (2021, Section 7) listed showing that the 10-fragment formula F_{TT_5} of the 5-vertex transitive tournament is 2-avoidable among problems attempted without success. Let $b_5 = f^\omega(0)$ for $f\colon 0\mapsto 01,\ 1\mapsto 23,\ 2\mapsto 4,\ 3\mapsto 21,\ 4\mapsto 0$ and $\tau\colon 0\mapsto 01,\ 1\mapsto 1001,\ 2\mapsto 0110,\ 3\mapsto 1,\ 4\mapsto 0$. Using two independently designed descent provers (124,174 states closed by a 12-part checker; 83,061, 107,485, and 50,110 states in a two-lemma prover), we prove that $U = \tau(b_5)$ avoids F_{TT_5}, so F_{TT_5} is 2-avoidable. Exhaustive trees show every binary word of length 17 meets F_{TT_3} (unconditional in Lean 4 + Mathlib) and length 47 meets F_{TT_4}, giving the exact threshold: for $k \geq 3$, F_{TT_k} is 2-avoidable if and only if $k \geq 5$ (bridged in Lean 4 + Mathlib). Deleting any fragment of F_{TT_5} is unavoidable on $\{0,1\}$ (longest avoiding lengths 46–74).

Verification Scope (G11): WARN: Human referee: pending
Main results: 2 of 5 in Lean Kernel (40%) · 1 Lean + Cert · 2 Cert Only · 1 open (not counted)
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:** combinatorics on words, xyx-formulas, avoidability index, transitive tournaments, morphic words, morphism descent, Lean 4 formalization

1. INTRODUCTION AND MAIN RESULTS

In combinatorics on words, a *formula* is a finite collection of *fragments* over an alphabet of variables, and an infinite word *avoids* the formula when no non-empty substitution of the variables places every fragment simultaneously as a contiguous factor of the word. Ochem and Rosenfeld [1] introduced xyx-*formulas*, in which every fragment has the palindromic shape XYX. Any loopless directed graph $G = (V, E)$ determines an xyx-formula F_G having one fragment $V_iV_jV_i$ for each directed arc $i \to j$ of G, and F_G is 2-*avoidable* (has avoidability index 2) if some infinite binary word over $\{0,1\}$ avoids F_G.

For an integer $k \geq 1$, the *transitive tournament* TT_k on vertex set $\{0,\dots,k-1\}$ has directed arc $i \to j$ for every $0 \leq i < j \leq k-1$. In particular, TT_5 has 10 arcs, and with variables A, B, C, D, E its xyx-formula is

$$F_{\mathrm{TT}_5} = ABA.ACA.ADA.AEA.BCB.BDB.BEB.CDC.CEC.DED. \quad (1)$$

In Section 7 (Discussion) of [1], Ochem and Rosenfeld explicitly recorded:

> *"Let us briefly mention the things that we have attempted to do in this paper, without success.* [...] *Show that the xyx-formula associated to the transitive tournament on 5 vertices is 2-avoidable."*

Prior to this question, Ochem and Rosenfeld [1] proved that Thue's ternary word w_3 avoids F_{TT_3} (Theorem 15, giving avoidability index at most 3 for every TT_k with $k \geq 3$), that every directed cycle has avoidability index 3 (Theorem 19), and that the complete 4-vertex loopless digraph with all 12 arcs is 2-avoidable (Theorem 23).

We resolve Ochem and Rosenfeld's transitive-tournament question affirmatively by exhibiting an explicit binary morphic image of their 5-letter word b_5, establishing the exact 2-avoidability threshold across all TT_k ($k \geq 3$), and proving that F_{TT_5} is fragment-minimal.

Define the prolongable endomorphism f on $\{0,1,2,3,4\}^*$ (introduced in Section 2 of [1]) and the binary morphism τ

into $\{0,1\}^*$ by

$$\begin{aligned} f&: \quad 0\mapsto 01,\ \ 1\mapsto 23,\ \ 2\mapsto 4,\ \ 3\mapsto 21,\ \ 4\mapsto 0,\\ \tau&: \quad 0\mapsto 01,\ \ 1\mapsto 1001,\ \ 2\mapsto 0110,\ \ 3\mapsto 1,\ \ 4\mapsto 0. \end{aligned} \tag{2}$$

Let $b_5 = f^\omega(0) = 01234210423010421\ldots$ be the unique infinite fixed point of f beginning with 0, and set $U = \tau(b_5) = 01100101101001101001\ldots$

Theorem 1 (2-Avoidability of F_{TT_5} via $U = \tau(b_5)$ — Cert Only). *With f and τ defined by (2), the infinite binary word $U = \tau(b_5)$ contains no five non-empty binary words $u_0, u_1, u_2, u_3, u_4 \in \{0,1\}^+$ such that $u_i u_j u_i$ is a factor of U for all $0 \le i < j \le 4$. Consequently, F_{TT_5} is 2-avoidable.*

Combining Theorem 1 with homomorphism monotonicity (Lemma 18 of [1]) and sharp binary decision-tree obstructions for $k = 3$ and $k = 4$ settles the full family of transitive tournaments.

Theorem 2 (Exact 2-Avoidability Threshold for TT_k — Lean + Cert). *For every integer $k \ge 3$, the* xyx*-formula F_{TT_k} is 2-avoidable if and only if $k \ge 5$.*

Since Thue's ternary word w_3 avoids F_{TT_3} (Theorem 15 of [1]) and hence F_{TT_4}, Theorem 2 implies that F_{TT_k} has avoidability index 3 for $k \in \{3,4\}$ and avoidability index 2 for every $k \ge 5$.

Theorem 3 (Unconditional Length-17 Unavoidability of TT_3 — Lean Kernel). *Every binary word $w \in \{0,1\}^{17}$ contains an occurrence of F_{TT_3}.*

In particular, F_{TT_3} is not 2-avoidable (formalized as `tt3_not_avoidable2`). Moreover, the bound 17 is sharp: the length-16 word

$$w_{16} = 0011010110101100$$

avoids both F_{TT_3} and the cyclic triangle F_{C_3}, which is likewise unavoidable at length 17. Both sharpness and unavoidability are proved unconditionally in Lean 4 + Mathlib (`lean/XYXAvoid/Tournaments.lean`).

Proposition 4 (Length-47 Unavoidability of TT_4 and Fragment Minimality of TT_5 — Cert Only). *Every binary word $w \in \{0,1\}^{47}$ contains an occurrence of F_{TT_4}, and this length is sharp (the longest binary words avoiding F_{TT_4} have length 46, verified across 23,770 tested words). Moreover, F_{TT_5} is fragment-minimal: removing any single arc e_{ij} $(0 \le i < j \le 4)$ produces a 9-fragment formula $F_{\mathrm{TT}_5 - e_{ij}}$ avoided by no infinite binary word, with longest avoiding binary words of lengths ranging from 46 to 74.*

Instant witness check. Although $U = \tau(b_5)$ avoids F_{TT_5} globally, it already contains occurrences of F_{C_3} in its first 7 bits, F_{TT_3} in its first 9 bits, and F_{TT_4} in its first 13 bits. The 8-line Python check below (executed verbatim in `recompute.py`) generates $b_5[0..17]$ and $U[0..13]$ and confirms the explicit 4-block occurrence $A = 0, B = 1, C = 01, D = 10$ of F_{TT_4}:

```
f = {'0':'01','1':'23','2':'4','3':'21','4':'0'}
tau = {'0':'01','1':'1001','2':'0110','3':'1','4':'0'}
b5 = '0'
for _ in range(5): b5 = ''.join(f[c] for c in b5)
U = ''.join(tau[c] for c in b5)
w = {'A':'0','B':'1','C':'01','D':'10'}
assert b5 == '01234210423010421' and U[:13] == '0110010110100'
assert all(w[i]+w[j]+w[i] in U[:13] for k,i in enumerate('ABCD') for
j in 'ABCD'[k+1:])
```

Verification scope and roadmap. Our verification architecture (Figure 1) combines machine-checked Lean 4 + Mathlib proofs (`lean/XYXAvoid/`, 217 public declarations, zero `sorry`, zero `native_decide`) with dual independently designed computer-assisted descent certificates. Section 2 fixes definitions matching the Lean 4 formalization verbatim. Section 3 proves the structural properties of b_5 and τ (Lemmas 5–6), Theorem 3, the general well-founded two-move descent soundness principle (Theorem 7), and the threshold bridge (Theorem 2). Section 4 details the dual descent provers, standalone certificate verification, control tests, descent-free factor-graph checks, and exhaustive decision trees behind Theorem 1 and Proposition 4. Section 5 presents the machine-generated Lean 4 Reader's Trust Boundary. Section 6 reviews prior work and states open questions (Conjecture 8).

2. DEFINITIONS, NOTATION, AND SETUP

Every definition in this section matches `lean/XYXAvoid/Basic.lean` and `lean/XYXAvoid/Morphisms.lean` verbatim.

Words and factors. For a finite alphabet Σ, let Σ^* denote the set of finite words over Σ, Σ^+ the set of non-empty finite words, and $\Sigma^\omega = (\mathbb{N} \to \Sigma)$ the set of right-infinite words. A word $u \in \Sigma^*$ is a *factor* of $w \in \Sigma^*$ (written `IsFactor u w`) if there exist $p, s \in \Sigma^*$ with $w = pus$. For $W \in \Sigma^\omega$ and $n \in \mathbb{N}$, let $\mathrm{prefixOf}(W,n) = W(0)W(1)\cdots W(n-1) \in \Sigma^n$. A finite word u is a *factor* of W (`IsFactorInf u W`) if there exists $i \in \mathbb{N}$ such that $W(i+j) = u[j]$ for all $0 \le j < |u|$; equivalently, u is a factor of $\mathrm{prefixOf}(W,n)$ for some $n \in \mathbb{N}$ (`isFactorInf_iff_exists_prefix`).

Loopless digraphs and xyx**-formulas.** A *loopless digraph* $G = (V, \mathrm{adj})$ (`LooplessDigraph V`) consists of a vertex set V and a relation $\mathrm{adj}: V \to V \to \mathrm{Prop}$ satisfying $\neg \mathrm{adj}(v,v)$ for all $v \in V$. An *occurrence* of F_G in a finite word $w \in \Sigma^*$ (`OccursDigraph G w`) is a map $h: V \to \Sigma^+$ assigning a *non-empty* word $h(v)$ to each vertex $v \in V$ such that $h(u)h(v)h(u)$ is a factor of w for every arc $u \to v$ of G. Distinct vertices are allowed to receive the same word; in particular, any cube xxx $(x \in \Sigma^+)$ gives an occurrence of every loopless digraph G via the constant assignment $h(v) = x$ (`occursDigraph_of_cube`), so every word avoiding F_G is cube-free.

Occurrences in an infinite word $W \in \Sigma^\omega$ (`OccursDigraphInf`) are defined identically with `IsFactorInf`. A finite word w or infinite word W *avoids* F_G (`AvoidsDigraph`, `AvoidsDigraphInf`) if it contains no occurrence of F_G. The formula F_G is 2-*avoidable* (`Avoidable2 G`) if some infinite binary word $W \in \{0,1\}^\omega$ avoids F_G.

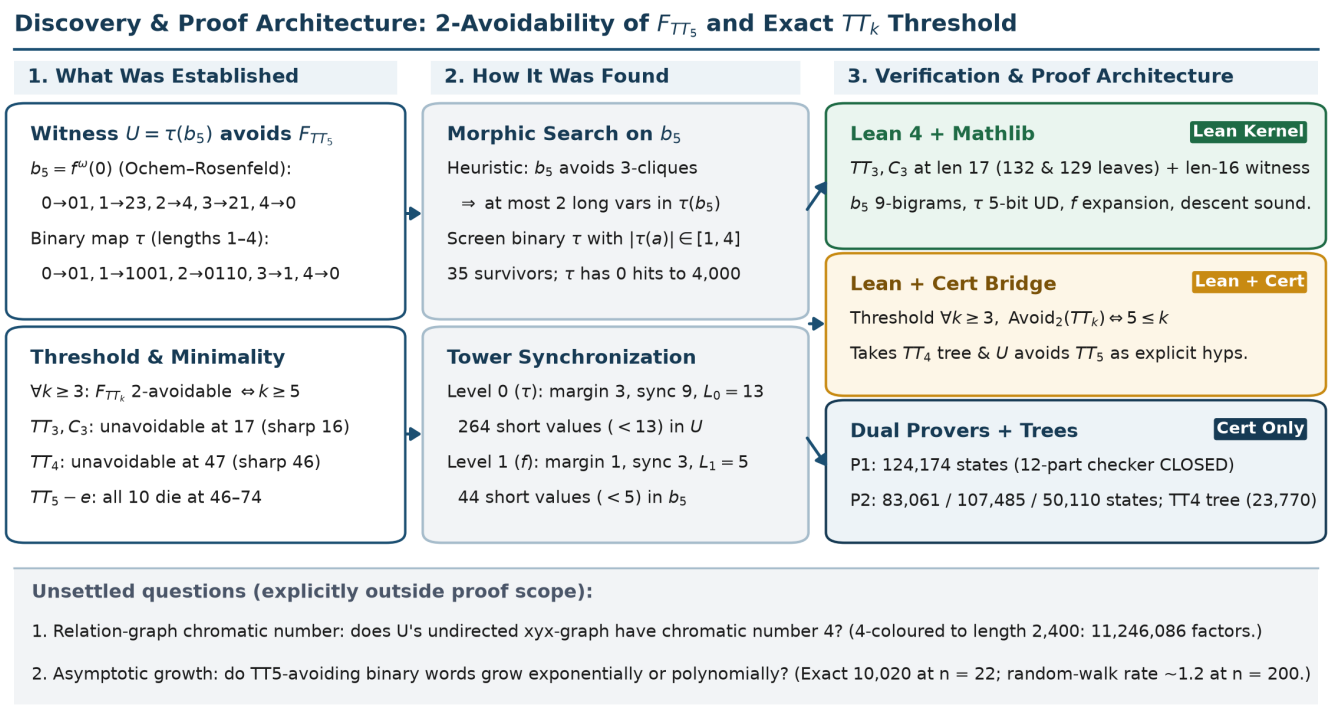


Figure 1. Discovery and proof architecture for the 2-avoidability of F_{TT_5} and the exact TT_k threshold. Every verified claim is separated by its exact trust tier: unconditional Lean 4 + Mathlib kernel theorems, the Lean 4 + Mathlib threshold bridge parameterized by explicit external certificate hypotheses, dual independently designed Cassaigne-style morphism descent provers plus exhaustive decision trees, and unsettled questions explicitly outside the proof scope.

Homomorphism monotonicity and tournaments. A *digraph homomorphism* $\varphi\colon H \to G$ (`DigraphHom H G`) maps vertices of H to vertices of G while preserving arcs. As in Lemma 18 of [1], if $\varphi\colon H \to G$ is a digraph homomorphism, then any occurrence h of F_G pulls back to the occurrence $h \circ \varphi$ of F_H, and hence 2-avoidability pushes forward: $\text{Avoidable}_2(H) \implies \text{Avoidable}_2(G)$ (`avoidable2_of_hom`). For $k \in \mathbb{N}$, the transitive tournament TT_k (`TT k`) has vertex set $\mathrm{Fin}\, k$ and $\mathrm{adj}(i,j) \iff i < j$. The cyclic triangle C_3 (`C3`) has vertex set $\mathrm{Fin}\, 3$ and arcs $0 \to 1 \to 2 \to 0$. Whenever $m \leq n$, the canonical inclusion $\mathrm{Fin}\, m \hookrightarrow \mathrm{Fin}\, n$ is a digraph homomorphism $\mathrm{TT}_m \to \mathrm{TT}_n$ (`ttHomOfLe`).

The witness word $U = \tau(b_5)$. Let f and τ be the non-erasing morphisms in (2). Since $f(0) = 01$ starts with 0 and has length $2 > 1$, f is prolongable at 0: each iterate $f^k(0)$ is a strict prefix of $f^{k+1}(0)$, with lengths $1, 2, 4, 7, 11, 17, 27, 44, 72$ for $k = 0, \ldots, 8$ (`b5_iter_lengths`) and $f^5(0) = 01234210423010421$ (`b5_prefix_17`). Applying τ letter-by-letter yields the infinite binary word $U = \tau(b_5)$ (`U`), whose first 36 letters are

$$U[0..36] = 011001011010011010010100110101100101,$$

and whose first 100 letters are kernel-verified in `U_prefix_100` (with 99 zeros and 101 ones among the first 200 letters). Figure 2 displays the iterates of f, the τ-coding into U, and the shortest prefixes of U containing F_{TT_3} (9 bits) and F_{TT_4} (13 bits). Both occurrences are kernel-verified in `Morphisms.lean` (`C3_occurs_U_prefix7`, `TT3_occurs_U_prefix9`, and `TT4_occurs_U_prefix13`), and `recompute.py` confirms that prefixes of lengths 6, 8, and 12 avoid C_3, TT_3, and TT_4 respectively.

3. STRUCTURAL FOUNDATIONS, SHARP TT_3 UNAVOIDABILITY, AND THE THRESHOLD BRIDGE

Every result in this section is formalized in `lean/XYXAvoid/` using Lean 4 + Mathlib with zero `sorry` and zero `native_decide`.

Coding, 9-Bigram Invariant, and Unique Decipherability. Although f is a suffix code (no codeword in $\{01, 23, 4, 21, 0\}$ is a suffix of another) and therefore injective on all of $\{0,1,2,3,4\}^*$ (`applyMorph_fB5_injective`), the outer binary morphism τ is *neither* a prefix code ($\tau(4) = 0$ is a prefix of $\tau(0) = 01$, which is a prefix of $\tau(2) = 0110$; $\tau(3) = 1$ is a prefix of $\tau(1) = 1001$) *nor* a suffix code ($\tau(3) = 1$ is a suffix of $\tau(0) = 01$ and $\tau(1) = 1001$; $\tau(4) = 0$ is a suffix of $\tau(2) = 0110$). Nevertheless, the factor language of b_5 obeys a tight bigram constraint that makes τ uniquely decipherable on all b_5-factors and makes f strictly expanding on all b_5-factors of length at least 3.

Define the 9-element bigram set

$$\mathcal{B}_5 = \{01, 04, 10, 12, 21, 23, 30, 34, 42\} \subset \{0,1,2,3,4\}^2,$$

and call a finite word $w \in \{0,1,2,3,4\}^*$ *$\mathcal{B}_5$-valid* (`ValidB5 w`) if every adjacent pair of letters in w belongs to $\mathcal{B}_5$. Every finite factor of b_5 is $\mathcal{B}_5$-valid (`validB5_of_isFactorInf_b5`), because each single-letter image $f(a)$ is $\mathcal{B}_5$-valid and every boundary pair across $f(a)f(b)$ for $ab \in \mathcal{B}_5$ again lies in $\mathcal{B}_5$:

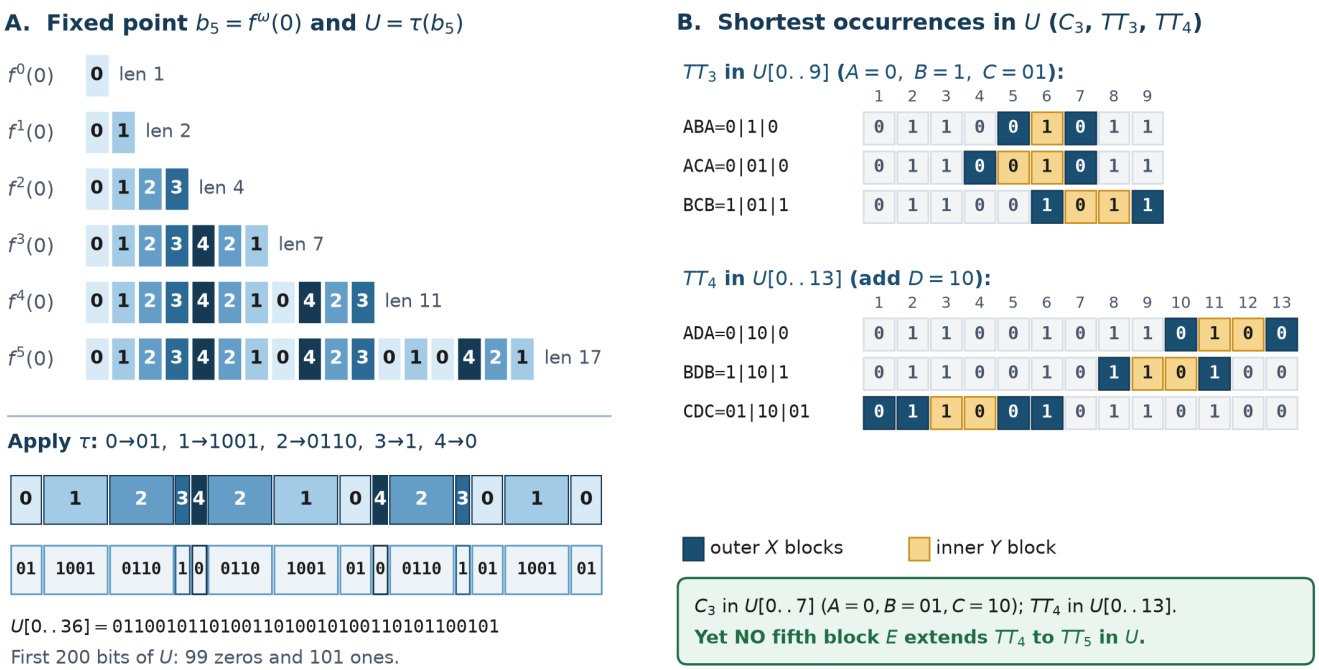


Figure 2. Panel A: Iterates $f^0(0), \ldots, f^5(0)$ converging to Ochem and Rosenfeld's 5-letter fixed point $b_5 = f^\omega(0)$ (lengths $1, 2, 4, 7, 11, 17$) and the τ-block substitution producing $U[0..36]$. Panel B: Exact shortest-prefix occurrences of F_{TT_3} in $U[0..9]$ ($A = 0, B = 1, C = 01$) and F_{TT_4} in $U[0..13]$ (adding $D = 10$); F_{C_3} already occurs in $U[0..7]$ ($A = 0, B = 01, C = 10$). Despite this dense short-block structure, Theorem 1 shows that no fifth block E extends to F_{TT_5} anywhere in U.

$01 \mapsto 12$, $04 \mapsto 10$, $10 \mapsto 30$, $12 \mapsto 34$, $21 \mapsto 42$, $23 \mapsto 42$, $30 \mapsto 10$, $34 \mapsto 10$, $42 \mapsto 04$.

Lemma 5 (Strict Expansion of f on $\mathcal{B}_5$-Valid Words — Lean Kernel). *Every $\mathcal{B}_5$-valid word $u \in \{0,1,2,3,4\}^*$ of length $|u| \geq 3$ satisfies $|u| < |f(u)|$.*

Proof. Note that $|f(a)| = 2$ for $a \in \{0,1,3\}$ and $|f(a)| = 1$ for $a \in \{2,4\}$. In $\mathcal{B}_5$, the only bigram within $\{2,4\}^2$ is 42, and no bigram in $\mathcal{B}_5$ begins with 2 and ends in $\{2,4\}$ (the outgoing arcs from 2 are 21 and 23). Consequently, among the 16 $\mathcal{B}_5$-valid triples abc, at least one letter lies in $\{0,1,3\}$, giving $|f(abc)| \geq 4 > 3$. Writing any $\mathcal{B}_5$-valid word of length ≥ 3 as $u = abc \cdot r$ and using $|f(r)| \geq |r|$ yields $|f(u)| = |f(abc)| + |f(r)| \geq 4 + |r| > 3 + |r| = |u|$ (`fB5_strictly_expanding_on_validB5`). In addition, $|f^3(a)| \geq 2$ for every $a \in \{0,1,2,3,4\}$, so $|f^3(w)| \geq 2|w|$ for all w (`iterMorph_fB5_3_length_ge`). □

Lemma 6 (Unique Decipherability of τ on Factors of b_5 — Lean Kernel). *If $u, v \in \{0,1,2,3,4\}^*$ are finite factors of b_5 satisfying $\tau(u) = \tau(v)$, then $u = v$.*

Proof. By Lemma 5, both u and v are $\mathcal{B}_5$-valid. It suffices to recover the leading letter a of any non-empty $\mathcal{B}_5$-valid word aw deterministically from at most the first 5 bits of $\tau(aw)$ (`tauHeadTag` in `Descent.lean`):

- Prefix 0 (exact) or $00\cdots \implies a = 4$ (since $42 \in \mathcal{B}_5$ gives $\tau(42) = 00110$, whereas no other $\tau(b)$ starts with 00).
- Prefix 01 (exact), $010\cdots$, or $01100\cdots \implies a = 0$ (as $01, 04 \in \mathcal{B}_5$ yield $\tau(01) = 011001$ and $\tau(04) = 010$).
- Prefix 0110 (exact) or $01101\cdots \implies a = 2$ (as $21, 23 \in \mathcal{B}_5$ yield $\tau(21) = 01101001$ and $\tau(23) = 01101$).
- Prefix 1001 (exact) or $10010\cdots \implies a = 1$ (as $10, 12 \in \mathcal{B}_5$ yield $\tau(10) = 100101$ and $\tau(12) = 10010110$).
- Prefix 1 (exact), 10 (exact), $101\cdots$, or $10011\cdots \implies a = 3$ (since $30, 34 \in \mathcal{B}_5$ yield $\tau(30) = 101$ and $\tau(34) = 10$, while the unique $\mathcal{B}_5$-valid triple aa_1a_2 with $|\tau(aa_1a_2)| < 5$ is $304 \mapsto 1010$, which already begins with 101 and therefore tags as 3 under every continuation).

By induction on $|u|$, peeling off the unique leading letter a and canceling $\tau(a)$ on the left yields $u = v$ (`applyMorph_tau_injective_on_b5_factors`). □

Sharp Length-17 Unavoidability of TT_3 and C_3.

Proof of Theorem 3. Let $w \in \{0,1\}^{17}$. By bit-flip symmetry (`occursDigraph_map_not_iff`), replacing w by its bitwise complement $\overline{w}$ preserves occurrences of F_{TT_3}, so we may assume without loss of generality that w begins with 0. (Formal proof in `Tournaments.lean` branches directly on the first bit and invokes `occursDigraph_map_not_iff` on the 1-branch.)

Define a binary decision tree (`CertTree3`) whose internal nodes branch on the next bit of a prefix p and whose leaves store a triple of non-empty binary words (a, b, c) together with three indices (p_1, p_2, p_3) witnessing that aba, aca, and bcb occur in p starting at p_1, p_2, p_3. By structural induction on the tree (`checkTT3Tree_sound`), if the boolean verifier `checkTT3Tree` accepts $(T, [0], 16)$, then every word $w \in \{0,1\}^{17}$ starting with 0 contains an occurrence of F_{TT_3}.

In `Tournaments.lean`, the explicit 132-leaf tree `tt3CertTree` passes `checkTT3Tree` on `[false]` at depth 16 by kernel reduction (`rfl`), proving `tt3_unavoidable_17`. Any infinite binary word W has a prefix of length 17, so `not_avoidable2_of_forall_len` immediately yields $\neg\text{Avoidable}_2(\text{TT}_3)$ (`tt3_not_avoidable2`).

An analogous 129-leaf tree `c3CertTree` proves unconditionally that every $w \in \{0,1\}^{17}$ contains F_{C_3} (`c3_unavoidable_17`). Sharpness of 17 for both formulas is certified by the length-16 word $w_{16} = 0011010110101100$: enumerating all 65 non-empty xyx-pairs via the verified complete extractor `xyxPairs` confirms zero occurrences of both F_{TT_3} and F_{C_3} (`tt3_sharp_16`, `c3_sharp_16`). □

Well-Founded Two-Move Descent Soundness. Before turning to the concrete descent certificates in Section 4, we record the abstract soundness principle behind both descent provers, proved unconditionally in `lean/XYXAvoid/Descent.lean`.

Theorem 7 (Well-Founded Two-Move Descent Soundness — Lean Kernel). *Let $G = (V, E)$ be a finite loopless digraph, $W \in \Sigma^\omega$ an infinite word, $\mathcal{S}$ a set of states equipped with a free-variable count* $\text{fv}\colon \mathcal{S} \to \mathbb{N}$, *a satisfiability relation* $\text{Sat}\colon \mathcal{S} \to \mathbb{N} \to \text{Prop}$, *and a certificate set $C \subseteq \mathcal{S}$ with initial state $s_0 \in C$. Suppose that:*

1. *Every occurrence $h\colon V \to \Sigma^+$ of F_G in W satisfies* $\text{Sat}(s_0, \sum_{v \in V} |h(v)|)$.
2. *For every $s \in C$ and $\mu \in \mathbb{N}$ with* $\text{Sat}(s, \mu)$, *either there exists $s' \in C$ with* $\text{fv}(s') < \text{fv}(s)$ *and some $\mu' \in \mathbb{N}$ satisfying* $\text{Sat}(s', \mu')$ (freeze), *or there exists $s' \in C$ with* $\text{fv}(s') = \text{fv}(s)$ *and some $\mu' < \mu$ satisfying* $\text{Sat}(s', \mu')$ (desubstitute).

Then W avoids F_G.

Proof. The lexicographic order $\prec$ on $\mathbb{N} \times \mathbb{N}$ is well-founded (`WellFounded.prod_lex`). By well-founded induction on $p = (\text{fv}(s), \mu) \in \mathbb{N} \times \mathbb{N}$, no state $s \in C$ can satisfy $\text{Sat}(s, \mu)$, since condition (2) produces a child (s', μ') with $s' \in C$ and $(\text{fv}(s'), \mu') \prec (\text{fv}(s), \mu)$. Applying this to $(s_0, \sum_{v \in V} |h(v)|)$ rules out every occurrence h of F_G in W (`freeze_or_desub_soundness`). □

The Exact Threshold Bridge.

Proof of Theorem 2. Fix an integer $k \geq 3$. If $k \geq 5$, the canonical inclusion $\text{TT}_5 \hookrightarrow \text{TT}_k$ is a digraph homomorphism; since $U = \tau(b_5)$ avoids F_{TT_5} (Theorem 1), homomorphism monotonicity (`avoidable2_of_hom`) implies that U also avoids F_{TT_k}, so F_{TT_k} is 2-avoidable (`tt_k_avoidable2_of_U_avoids_tt5`). Conversely, suppose F_{TT_k} is 2-avoidable and $k < 5$; then $k = 3$ or $k = 4$. The case $k = 3$ is ruled out unconditionally by Theorem 3 (`tt3_not_avoidable2`), and the case $k = 4$ is ruled out by Proposition 4 ($\neg\text{Avoidable}_2(\text{TT}_4)$). This exact implication is formalized as `XYXAvoid.tt_k_avoidability_threshold`, taking `h_tt4_unavoid` and `h_U_avoids_tt5` as explicit named hypotheses. □

4. DUAL MORPHISM DESCENT PROOFS, EXHAUSTIVE TREES, AND EMPIRICAL VERIFICATION

We now describe the computational proofs establishing Theorem 1 and Proposition 4, together with their independent cross-checks and descent-free sanity tests.

How the Witness τ Was Found. In a morphic word $g(w)$, long factors synchronize to block boundaries of g, allowing long variable values in an occurrence of F_G to be desubstituted into related factors of w. Because b_5 avoids three pairwise xyx-related long factors, at most two of the five variables in a putative occurrence of F_{TT_5} in a binary image $\tau(b_5)$ can be simultaneously long, forcing the remaining three variables into a finite set of short binary factors. Screening binary morphisms $\tau\colon \{0,1,2,3,4\} \to \{0,1\}^+$ with codeword lengths in $\{1,2,3,4\}$ against bounded prefix searches yielded 35 surviving morphisms; among them, τ from (2) exhibited zero occurrences of F_{TT_5} on long prefixes and clean two-level synchronization parameters.

Two-Level Synchronization and Margin Descent. To apply Cassaigne-style morphism descent [6] (`[to check]`) to $U = \tau(b_5)$ without requiring τ to be a prefix or suffix code, we use *synchronization with a margin* $r \geq 0$: every factor w of length m in the upper word $g(w_{\text{low}})$ has a unique set of g-block boundaries inside the interior window $[r, m-r]$. Any variable value X of length at least $L = \max(m, 2r + 2M - 1)$ (where $M = \max_a |g(a)|$) therefore has identical interior block cuts across all its occurrences, containing at least two cuts, and decomposes uniquely as

$$X = s\, g(X')\, p,$$

where s is a suffix piece of length in $[r, r + M - 1]$ ending at the first cut $\geq r$, p is a prefix piece starting at the last cut $\leq |X| - r$, and $X' \neq \varepsilon$ is a unique non-empty factor of w_{low} (uniqueness follows from Lemma 6 at Level 0 and injectivity of f at Level 1).

As verified directly in `recompute.py` from the exact factor sets of b_5 and U up to length 52 (5,564 non-empty factors of b_5, or 5,565 including the empty word; 5,384 non-empty factors of U, or 5,385 including the empty word):

- **Level 0** ($U = \tau(b_5)$, $M_0 = 4$)**:** margin $r_0 = 3$, synchronization length $m_0 = 9$ (fails at $m = 8$), length threshold $L_0 = 13$, yielding 264 short non-empty binary factors of length < 13, 15 suffix pieces, and 12 prefix pieces.
- **Level 1** ($b_5 = f(b_5)$, $M_1 = 2$)**:** margin $r_1 = 1$, synchronization length $m_1 = 3$ (fails at $r = 0$), length threshold $L_1 = 5$, yielding 44 short non-empty factors of length < 5, 7 suffix pieces, and 6 prefix pieces.

Whenever a state contains a variable shorter than L_i, a *freeze* move branches over all short factors (264 at Level 0, 44 at Level 1), reducing the free-variable count fv by 1. Whenever all live variables have length at least L_i, a *desubstitution* move assigns boundary types (s, p) and lifts every constraint to the lower level; because f strictly expands every $\mathcal{B}_5$-valid factor of length ≥ 3 (Lemma 5), repeated desubstitution strictly decreases total variable length, matching Theorem 7.

Dual Independently Designed Provers and Standalone Verification. Theorem 1 is established by two independently designed descent provers plus a non-searching certificate checker and a third independent reimplementation (summary in Table 1):

1. **Prover 1 and standalone 12-part checker:** Exploring the two-level tower directly from F_{TT_5} closes with 124,174 states expanded, 28,189,348 freeze branches (28,125,844

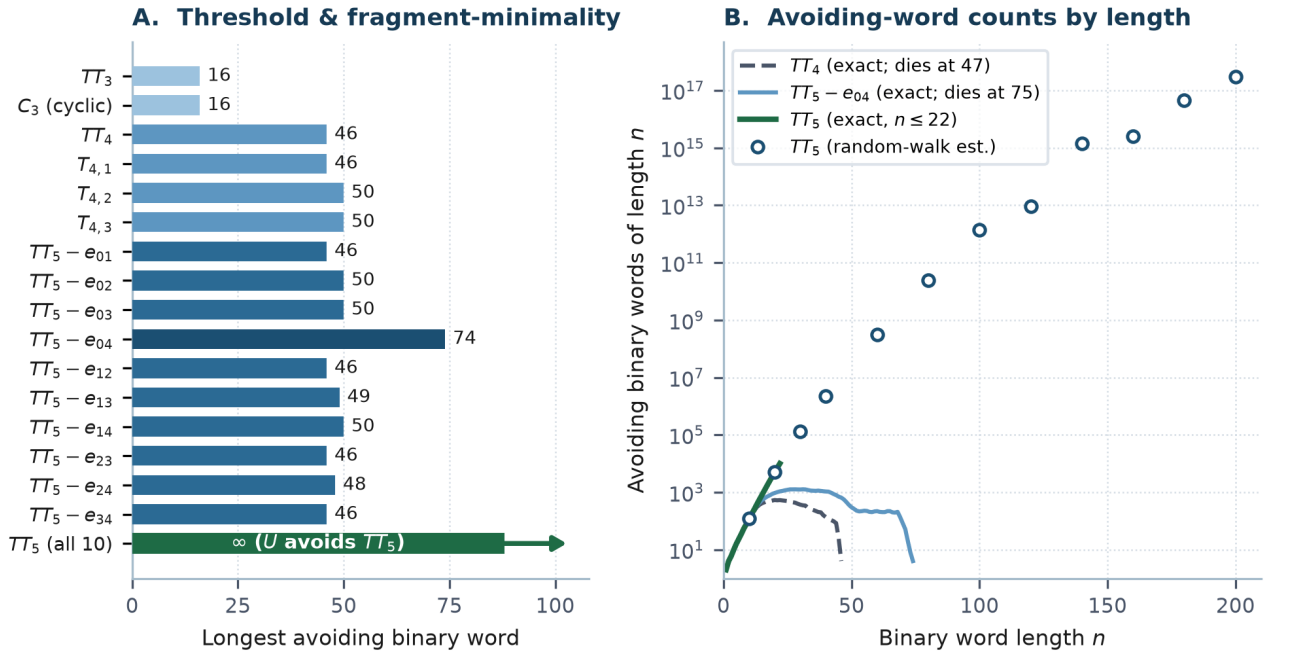


Figure 3. Panel A: Exact longest binary word avoiding each formula: 3-vertex tournaments (TT_3, C_3: 16), all four 4-vertex tournaments (46–50), all 10 single-fragment deletions $TT_5 - e_{ij}$ (46–74), and TT_5 (∞, avoided by $U = \tau(b_5)$). Panel B: Log-scale counts of avoiding binary words of length n for TT_4 (exact; vanishes at $n = 47$), $TT_5 - e_{04}$ (exact; vanishes at $n = 75$), and TT_5 (exact for $n \le 22$, reaching 10,020, plus hollow-circle random-walk estimates up to $n = 200$, plotted without joining lines to distinguish estimates from exact counts).

Table 1. Independent computer-assisted verifications of Theorem 1 ($U = \tau(b_5)$ avoids F_{TT_5}).

Verifier	Design / Scope	States	Outcome
Prover 1	Direct 2-level tower	124,174	AVOIDED
Checker	Non-searching, 12 parts	124,174	12/12 CLOSED
Re-impl.	Independent 2-level code	124,174	AVOIDED
Prover 2 (L1)	Long TT_3 ($\lvert u_i\rvert \ge 16$)	83,061	AVOIDED
Prover 2 (L2)	Long C_3 ($\lvert u_i\rvert \ge 16$)	107,485	AVOIDED
Prover 2 (Main)	TT_5 (≤ 2 vars ≥ 16)	50,110	AVOIDED

immediately dead), and 124,267 desubstitutions, emitting a closed state certificate with zero satisfiable terminal states. A completely separate non-searching checker—sharing only the certificate file and using independently constructed factor tables—verified across 12 partition runs (12/12 CLOSED) that the initial state is covered, no state is terminal, and every freeze (28,189,348 total, 28,125,844 dead) and desubstitution (124,277 checks, 94,085,851 dead, 863,735 cover tests) either dies or lands inside a covered state. An independent reimplementation of the same two-level algorithm recovered the exact same 124,174-state closed set up to variable renaming and verified all factor tables and cut rules by brute force on a 400,000-letter prefix of b_5 and a 10^6-letter prefix of U.

2. **Prover 2 (independent two-lemma design):** Written without reading Prover 1's code, Prover 2 uses a distinct per-arc template representation (four template types per arc depending on which endpoints are frozen, with context slots truncated to fixed width and synchronization length $B = 2(D + M)$) and decomposes the proof into three runs:
   - *Lemma L1:* U contains no occurrence of F_{TT_3} in which all three words have length ≥ 16 (83,061 states, 0 hits).
   - *Lemma L2:* U contains no occurrence of F_{C_3} in which all three words have length ≥ 16 (107,485 states, 0 hits).
   - *Main run:* Because every 3-vertex subtournament of any tournament is isomorphic to TT_3 or C_3, Lemmas L1 and L2 force any putative occurrence of F_{TT_5} in U to have at most two variables of length ≥ 16; pruning states with three long variables closes F_{TT_5} in 50,110 states (0 hits).

   An independent rerun of Prover 2's unchanged source code reproduced the exact state counts (83,061, 107,485, 50,110) and audited the code line by line.
3. **Companion remark (all 12 five-vertex tournaments):** Under Prover 2's Lemmas L1–L2, the exact same pruning applies to all 12 non-isomorphic tournaments on 5 vertices, and all 12 runs terminate AVOIDED in 30,041 to 50,110 states. Thus U is cube-free and avoids every 5-vertex tournament formula.

Control tests and descent-free evidence. Both provers were calibrated against known positive and negative benchmarks: Prover 1 reproduces Theorem 15 of [1] (w_3 avoids F_{TT_3}, 9,293 states) and outputs concrete occurrences on positive controls; Prover 2 reproduces Theorems 20 and 23 of [1], finds occurrences on Thue–Morse, TT_4, and $TT_5 - e$, and

matches direct search across 24 small test formulas with 0 disagreements.

Independently of all descent logic, four direct searches confirmed consistency on finite factor sets of U: (i) our standalone `recompute.py` confirms 0 cubes and 0 occurrences of F_{TT_5} in $U[0..500]$ (182,236 xyx-pairs); (ii) direct prefix searches find 0 occurrences in $U[0..1{,}000]$, $U[0..3{,}000]$ (8,315,810 arcs), and $U[0..4{,}000]$ (15,229,452 arcs); (iii) an 8,000-letter prefix covering every factor of U up to length 1,060 contains 0 cubes and 0 occurrences of F_{TT_5}; and (iv) an exact factor-graph construction covering *every* factor of the infinite word U of length up to 2,400 (11,519,976 factors, 28,495,404 directed xyx-arcs, 27,963,777 undirected related pairs) found 0 cubes, 1,989,484 triangles, 37,924 4-cliques, and 0 5-cliques—and among factors of length ≥ 16, 0 triangles (0 K_3's), directly corroborating Lemmas L1 and L2 up to fragment length 2,400.

Exhaustive Trees for TT_4 and Fragment-Minimality. To prove Proposition 4, unpruned depth-first binary trees test every binary word w by maintaining the exact xyx-pair relation $P(w)$ incrementally as each bit is appended (exploiting bit-flip symmetry). In `recompute.py`, all six 3- and 4-vertex tournaments are recomputed live from scratch (total runtime $\approx$ 8 s), confirming exact agreement at every length with two prior independent tree implementations (Table 2 and Figure 3):

- TT_3 and C_3 die at length 17 (longest avoiding length 16; 526 and 514 tested words; peak widths 40 and 38).
- TT_4 dies at length 47 (longest avoiding length 46; 23,770 tested words; peak width 536 at length 20); the remaining three 4-vertex tournaments $T_{4,1}, T_{4,2}, T_{4,3}$ die at lengths $47, 51, 51$ (14,390, 14,726, and 18,674 tested words).
- Deleting any single arc e_{ij} from TT_5 leaves a 9-arc formula whose binary tree dies (checked by two agreeing exhaustive C programs): longest avoiding words have lengths 46 ($e_{01}, e_{12}, e_{23}, e_{34}$), 48 ($e_{24}$), 49 ($e_{13}$), 50 ($e_{02}, e_{03}, e_{14}$), and 74 ($e_{04}$, peak width 1,270 at length 31).

By contrast, the exact count of length-n binary words avoiding F_{TT_5} grows through $n = 1 \ldots 22$ as 2, 4, 6, 10, 16, 24, 36, 56, 80, 118, 174, 254, 378, 554, 802, 1,168 (verified in `recompute.py` for $n \leq 16$) and 1,700, 2,450, 3,502, 4,992, 7,094, 10,020 for $n = 17 \ldots 22$.

Table 2. Exhaustive binary-tree bounds for 3- and 4-vertex tournaments and TT_5 minus one arc.

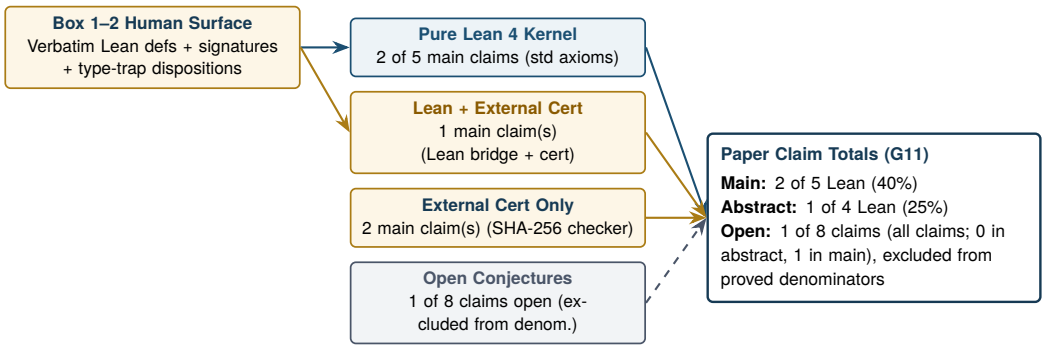
Formula	Longest	Peak	Verification
TT_3	16	40	Lean 4 + script (526)
C_3	16	38	Lean 4 + script (514)
TT_4	46	536	Script (23,770)
$T_{4,1}, T_{4,2}, T_{4,3}$	$46, 50, 50$	$304, 252, 366$	Script (exact)
$\mathrm{TT}_5 - e_{ij}$ (10 arcs)	46–74	536–1,270	Dual trees
TT_5	∞	$> 10{,}020$	Thm. 1 (U)

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:** 3 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: 217/217 public theorems (source-level approx) | WARN: Fidelity: 5/5 exact (draft cross-family)
WARN: Witnesses: pending gate replay | WARN: Human referee: pending

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

XYXAvoid.LooplessDigraph (XYXAvoid/Basic.lean:207-209) — Section 2 (loopless directed graph on vertex type V)

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

XYXAvoid.TT (XYXAvoid/Basic.lean:382-384) — Section 2 (transitive tournament TT_k on Fin k with arcs i < j)

```
def TT (k : ℕ) : LooplessDigraph (Fin k) where
  adj i j := i.val < j.val
  loopless v := Nat.lt_irrefl v.val
```

XYXAvoid.IsFactor (XYXAvoid/Basic.lean:44-45) — Section 2 (contiguous factor u of a finite word w)

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

```
def IsFactor (u w : List α) : Prop :=
  ∃ p s : List α, w = p ++ u ++ s
```

XYXAvoid.IsFactorInf (XYXAvoid/Basic.lean:163-164) — Section 2 (contiguous factor u of an infinite word W : Nat -> 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.OccursDigraph (XYXAvoid/Basic.lean:254-255) — Section 2 (occurrence of F_G in finite word w via non-empty h : V -> List alpha)

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

XYXAvoid.OccursDigraphInf (XYXAvoid/Basic.lean:258-259) — Section 2 (occurrence of F_G in infinite word W)

```
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 (infinite word W avoids F_G)

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

XYXAvoid.Avoidable2 (XYXAvoid/Basic.lean:270-271) — Section 2 (F_G is 2-avoidable: some infinite binary word avoids F_G)

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

XYXAvoid.fB5 (XYXAvoid/Morphisms.lean:260-265) — Section 1, eq. (2) (Ochem-Rosenfeld 5-letter morphism f)

```
def fB5 : Fin 5 → List (Fin 5)
  | 0 => [0, 1]
  | 1 => [2, 3]
  | 2 => [4]
  | 3 => [2, 1]
  | 4 => [0]
```

XYXAvoid.b5 (XYXAvoid/Morphisms.lean:274-275) — Section 1 & 2 (infinite fixed point b5 = f^omega(0))

```
def b5 : ℕ → Fin 5 :=
  fixedPoint fB5 0 fB5_nonempty fB5_prolongable
```

XYXAvoid.tau (XYXAvoid/Morphisms.lean:293-298) — Section 1, eq. (2) (binary coding morphism tau)

```
def tau : Fin 5 → List Bool
  | 0 => [false, true]
  | 1 => [true, false, false, true]
  | 2 => [false, true, true, false]
  | 3 => [true]
  | 4 => [false]
```

XYXAvoid.U (XYXAvoid/Morphisms.lean:304-305) — Section 1 & 2 (witness infinite binary word U = tau(b5))

```
def U : ℕ → Bool :=
  applyMorphInf tau tau_nonempty b5
```

XYXAvoid.ValidB5 (XYXAvoid/Descent.lean:207-210) — Section 3.1 (closed 9-bigram invariant B_5 of b5)

```
def ValidB5 : List (Fin 5) → Prop
  | [] => True
  | [_] => True
  | a :: b :: t => b5Bigram a b = true ∧ ValidB5 (b :: t)
```

XYXAvoid.occMeasure (XYXAvoid/Descent.lean:425-426) — Section 3.3 (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.applyMorph (XYXAvoid/Morphisms.lean), XYXAvoid.iterMorph (XYXAvoid/Morphisms.lean), XYXAvoid.fixedPoint (XYXAvoid/Morphisms.lean), XYXAvoid.prefixOf (XYXAvoid/Basic.lean), XYXAvoid.applyMorphInf (XYXAvoid/Morphisms.lean), XYXAvoid.b5Bigram (XYXAvoid/Descent.lean).

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

XYXAvoid.tt_k_avoidability_threshold (XYXAvoid/Tournaments.lean:1712-1724) src:ff6d63af4f2fa3

```
theorem tt_k_avoidability_threshold
    (h_tt4_unavoid : ¬ Avoidable2 (TT 4))
    (h_U_avoids_tt5 : AvoidsDigraphInf (TT 5) U) :
    ∀ k : ℕ, 3 ≤ k → (Avoidable2 (TT k) ↔ 5 ≤ k)
```

Assumptions: Takes external hypotheses h_tt4_unavoid (length-47 exhaustive tree in recompute.py) and h_U_avoids_tt5 (dual descent certificates); unconditionally proves k = 3 and k >= 5 upward lift. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.tt3_unavoidable_17 (XYXAvoid/Tournaments.lean:1515-1527) src:2417fac5572c2f

```
theorem tt3_unavoidable_17 (w : List Bool) (hw : w.length = 17) :
    OccursDigraph (TT 3) w
```

Assumptions: Any finite binary word w : List Bool of length 17 (unconditional; zero external hypotheses). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.fB5_strictly_expanding_on_validB5 (XYXAvoid/Descent.lean:311-321) src:13b467a9bfb730

```
theorem fB5_strictly_expanding_on_validB5 {u : List (Fin 5)} (hu : ValidB5 u)
    (hlen : 3 ≤ u.length) : u.length < (applyMorph fB5 u).length
```

Assumptions: Any 5-letter word u satisfying the 9-bigram invariant ValidB5 with length >= 3 (unconditional). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

XYXAvoid.applyMorph_tau_injective_on_b5_factors (XYXAvoid/Descent.lean:414-418) src:f6d18e02ac02fb

```
theorem applyMorph_tau_injective_on_b5_factors {u v : List (Fin 5)}
    (hu : IsFactorInf u b5) (hv : IsFactorInf v b5)
    (heq : applyMorph tau u = applyMorph tau v) : u = v
```

Assumptions: Any two finite factors u, v of b5 with applyMorph tau u = applyMorph tau v (unconditional). 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 loopless digraph G, infinite word W, state space, measure, and certificate set closed under freeze or desubstitution (unconditional). 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
`instDecidableEmbedsMask`	`local_instance_or_notation`	`instDecidableEmbedsMask`	intended_convention	Standard Decidable instance inst DecidableEmbedsMask in companion module Census.lean; carries no non-standard axioms or notation overrides.
`instDecidableIsAboveMin7`	`local_instance_or_notation`	`instDecidableIsAboveMin7`	intended_convention	Standard Decidable instance inst DecidableIsAboveMin7 in companion module Census.lean; carries no non-standard axioms or notation overrides.
`instDecidableIsBelowMax5`	`local_instance_or_notation`	`instDecidableIsBelowMax5`	intended_convention	Standard Decidable instance inst DecidableIsBelowMax5 in companion module Census.lean; carries no non-standard 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 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: U = tau(b5) avoids F_{TT5}, so F_{TT5} is 2-avoidable	Cert Only	proofs_r17.py (c336910e4420) [regenerable; replay pending]; recompute.py (4053fe00ef29) [regenerable; replay pending]	External cert only; fidelity: n/a	Dual independently designed morphism descent provers (124,174 states closed 12/12 by separate checker; 83,061, 107,485, and 50,110 states in second prover); barrier: 124,174-state and 28,189,348-freeze certificate exceeds pure Lean kernel reduction.
Thm. 2: For every k >= 3, F_{TT_k} is 2-avoidable if and only if k >= 5	Lean + Cert	`XYXAvoid.tt_k_avoidability_threshold` + recompute.py (4053fe00ef29) [regenerable; replay pending]; proofs_r17.py (c336910e4420) [regenerable; replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure-kernel bridge (unconditional for k = 3 and k >= 5 lift) taking external hypotheses h_tt4_unavoid (recompute.py, 23,770 words) and h_U_avoids_tt5 (dual descent provers).
Thm. 3: Every binary word of length 17 contains an occurrence of F_{TT3}	Lean Kernel	`XYXAvoid.tt3_unavoidable_17`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib kernel proof via 132-leaf decision tree and bit-flip symmetry (zero external hypotheses).

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Prop. 4: Every binary word of length 47 contains F_{TT4} (sharp at 46), and all 10 deletions TT5 - e_{ij} die at lengths 46 to 74	Cert Only	recompute.py (4053fe00ef29) [regenerable; replay pending]; threshold.py (711e17c76e5a) [regenerable; replay pending]	External cert only; fidelity: n/a	Exact exhaustive binary decision trees (TT4 recomputed live in recompute.py across 23,770 words; 10 deletions verified by two agreeing C tree programs in threshold.out).
Lem. 5: Strict expansion |u| < |f(u)| on ValidB5 words of length >= 3	Lean Kernel	`XYXAvoid.fB5_strictly_expanding_on_validB5`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib kernel proof on all ValidB5 words of length >= 3.
Lem. 6: Binary morphism tau is injective (uniquely decipherable) on all finite factors of b5	Lean Kernel	`XYXAvoid.applyMorph_tau_injective_on_b5_factors`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib kernel proof via verified 5-bit lookahead tagger on ValidB5 words.
Thm. 7: Well-founded two-move (freeze / desubstitute) lexicographic descent soundness	Lean Kernel	`XYXAvoid.freeze_or_desub_soundness`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Unconditional Lean 4 + Mathlib kernel proof via Prod.Lex well-founded induction on (free Vars, occMeasure).
Conj. 8: Chromatic number 4 for the relation graph of U and asymptotic growth rate of TT5-avoiding binary words	Unsettled	Open conjecture	Lean: none; fidelity: n/a	Explicitly open conjecture (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: f520d2653492 | Lean mode: committed_receipt | Closure SHA-256: 92ed2e615a56 | Git anchor: 4d34d065283c | Axioms SHA-256: 76884cfcb660 | Fidelity SHA-256: bf65e61408d2.

6. PRIOR WORK AND OPEN QUESTIONS

The xyx-formula framework, the digraph homomorphism lemma (Lemma 18), the 5-letter fixed point b_5, Theorems 15, 19, 20, and 23, and the open problem of showing 2-avoidability of F_{TT_5} (Section 7, Discussion) are all due to Ochem and Rosenfeld [1]. The morphism-descent method traces to Cassaigne [6] (`[to check]`). Closest subsequent works are Badkobeh and Ochem [2], who used two other binary images of b_5 ($h_{12}(b_5)$ and $c(b_5)$) to characterize morphic words via different formulas (exact factor-set comparison confirms $U = \tau(b_5)$ equals neither $h_{12}(b_5)$ nor $c(b_5)$, even up to reversal or bit complement), and Ochem [3], which treats doubled patterns with reversal and avoids F_{C_3} on 3 letters without addressing 4- or 5-vertex tournaments. Earlier avoidability studies on 2-variable and circular formulas appear in [4] (`[to check]`) and [5] (`[to check]`).

Prior-work search scope. Two independent logged literature searches passed `litsearch.py check` (exit 0: 37 searches evaluating 8 candidates and 26 searches evaluating 1 candidate across arXiv, OEIS, OpenAlex, Semantic Scholar, zbMATH Open, DataCite, Zenodo, HAL, OSF, and GitHub), followed by a targeted 7-query CORE search (44 records, 2026-11-21) and a 5-query Google Scholar sweep (2026-11-22). No prior proof of 2-avoidability of F_{TT_5} or construction of $\tau(b_5)$ was found in the searches logged. Remaining unsearched sources include MathOverflow and Math Stack Exchange (API rate-limited), direct proceedings sweeps of WORDS, DLT, and Journées Montoises (2021–2026) beyond indexed venues, and print books/theses (Lothaire [6], Cassaigne's 1994 thesis, and Clark's 2001 thesis, all `[to check]` in `references-checked.md`).

Open questions. Define the undirected *relation graph* of U to have vertex set equal to the non-empty factors of U, with an edge between x and y whenever xyx or yxy is a factor of U. Because U contains F_{TT_4} (Figure 2), this graph has clique number at least 4, and Theorem 1 (with the companion 12-tournament runs) shows its clique number is exactly 4. DSATUR colouring finds verified 4-colourings across all 11,246,086 related factors of U up to length 2,400 (whereas control words such as Thue–Morse, Fibonacci, and Narayana binary images require 10 to 12 colours under the same heuristic).

Conjecture 8 (Chromatic Number 4 and Growth Rate — Open).

1. *The undirected* xyx*-relation graph of* $U = \tau(b_5)$ *is* 4*-colourable* (*hence has chromatic number* 4, *which would imply that* U *avoids* F_G *for every digraph* G *with chromatic number at least* 5).
2. *Determine whether the number of binary words of length* n *avoiding* F_{TT_5} *grows exponentially or polynomially in* n (*exact counts reach* 10,020 *at* $n = 22$, *while random-walk estimates over* 30,000 *walks yield* $\approx 3 \times 10^{17}$ *at* $n = 200$, *corresponding to a growth ratio declining from* 1.48 *at* $n = 30$ *to* ≈ 1.2 *at* $n = 200$, *leaving both regimes open*).

Finally, a human-checkable proof of Theorem 1—or a Lean 4 kernel replay of the 124,174-state descent certificate and the 23,770-node TT_4 tree—remains a natural target for future work.

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

All draft files, figure generators, and standalone verification scripts live in `papers/transitive-tournament-five/draft1/`, and the Lean 4 + Mathlib library lives in `lean/XYXAvoid/` (217 verified public declarations, zero `sorry`, zero `native_decide`, standard axioms only). Running `python3 papers/transitive-tournament-five/draft1/recompute.py` independently reproduces the construction of b_5 and U, the 9-bigram invariant, τ's 5-bit lookahead unique decipherability, the synchronization bounds and short-factor counts (264 and 44), the shortest-prefix occurrences in U, the exhaustive binary decision trees for all six 3- and 4-vertex tournaments (TT_4 in 23,770 tested words), the exact TT_5-avoiding counts for $n \leq 16$, and the direct check on $U[0..500]$ in under 10 seconds.

AI research agents built and directed by Tyler B. Hughes generated the morphism search, dual descent provers, certificate checker, Lean 4 + Mathlib formalization, figures, and manuscript text. 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. arXiv:2005.05505.

(2) Badkobeh, G.; Ochem, P. More characterizations of morphic words. Preprint, **2023**. arXiv:2312.10757.

(3) Ochem, P. Doubled patterns with reversal and square-free doubled patterns. Preprint, **2021**. arXiv:2105.04673.

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

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

(6) Lothaire, M. *Algebraic Combinatorics on Words*; Encyclopedia of Mathematics and its Applications, Vol. 90; Cambridge University Press: Cambridge, **2002**. DOI: 10.1017/CBO9781107326019.