MATHDISCOVERIES
Mathematical Preprint

Working Draft

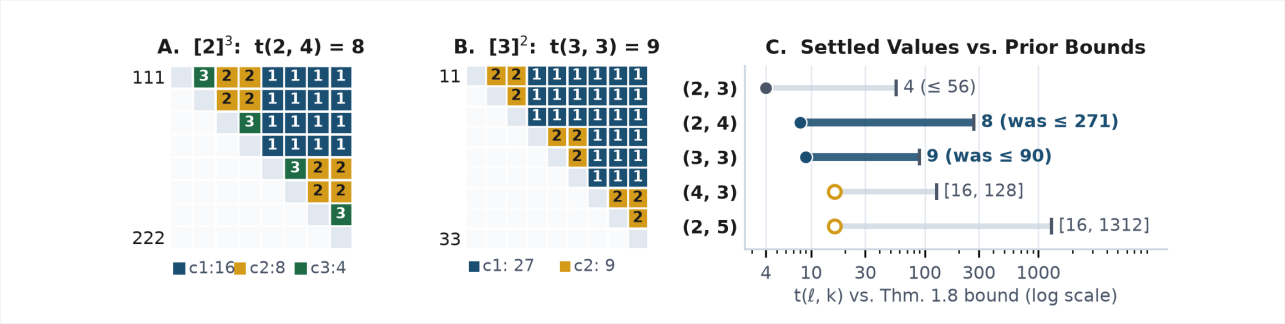
Flashes and Rainbows in Edge-Coloured Tournaments: Exact Thresholds $t(2,4) = 8$ and $t(3,3) = 9$

Tyler B. Hughes*,†

†Independent Researcher, United States

Supporting Information: `recompute.py`, `lean/TournamentEdgeColor/`, and SAT/LRAT logs

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



ABSTRACT: In an edge-coloured tournament, an ℓ-flash is a monochromatic directed walk of length ℓ and a k-rainbow is a directed walk of length k with pairwise distinct arc colours. Girão, Illingworth, Michel, Savery, and Scott (2024) conjectured that the maximum order $t(\ell,k)$ of an avoiding tournament equals ℓ^{k-1} for walks (Conjecture 1.7) and paths (Conjecture 5.3), leaving all $\ell \geq 2$, $k \geq 3$ open. We establish $t(2,4) = 8$ and $t(3,3) = 9$ for walks and paths, improving prior bounds 271 and 90. Unconditionally in Lean 4 + Mathlib (`lean/TournamentEdgeColor/`, 114 declarations, zero `sorry`, zero `native_decide`), we prove $t(\ell,k) \geq \ell^{k-1}$, that d-colourings avoiding ℓ-flashes correspond exactly to grid embeddings in $\{1,\dots,\ell\}^d$, small cases through $t(2,3) = 4$, and CNF equisatisfiability, symmetry-breaking, and RUP-soundness bridges. Coupling these bridges with four externally checked UNSAT proofs (`drat-trim`, 0 RAT core lemmas; `lrat-check`) settles $t(2,4) = 8$ and $t(3,3) = 9$. Exhaustive enumeration confirms walk-good equals grid-embeddable for $(2,3)$, $(2,4)$, and $(3,3)$ (43 and 107 extremal classes).

Verification Scope (G11): WARN: Human referee: pending
Main results: 3 of 6 in Lean Kernel (50%) · 2 Lean + Cert · 1 Cert Only · 1 open (not counted)
Abstract claims: 2 of 5 in Lean Kernel (40%) · 2 Lean + Cert · 1 Cert Only

Full Trust Boundary: Section 5.

MSC 2020: 05C20, 05C55, 05D10, 68V15 **Keywords:** tournaments, edge colourings, monochromatic walks, rainbow paths, SAT certificates, Lean 4 formalization

1. INTRODUCTION AND MAIN RESULTS

Hold a round-robin tournament among n players, orienting each pair from winner to loser, and paint every arc with a colour from an unrestricted palette. An ℓ-*flash* is a directed walk of ℓ arcs all sharing one colour, and a k-*rainbow* is a directed walk of k arcs carrying k pairwise distinct colours. How large can n be before every edge colouring of every n-vertex tournament is forced to contain either an ℓ-flash or a k-rainbow?

For transitive tournaments (strict rankings), Lefmann, Rödl, and Thomas [2] (`[to check]`) introduced the corresponding threshold $f(\ell,k)$, constructed a lexicographic grid colouring showing $f(\ell,k) \geq \ell^{k-1}$, conjectured $f(\ell,k) = \ell^{k-1}$ for all positive integers ℓ, k, and proved the small cases $\ell \leq 2$ or $k \leq 4$ (as reported in [1]). Girão, Illingworth, Michel, Savery, and Scott [1] extended the question to arbitrary tournaments, defining $t_{\text{walk}}(\ell,k)$ (denoted $t(\ell,k)$ in [1]) as the maximum order of a tournament admitting a colouring with no ℓ-flash walk and no k-rainbow walk, and $t_{\text{path}}(\ell,k)$ for the stricter variant forbidding only monochromatic ℓ-paths and rainbow k-paths. Because every path is a walk, $t_{\text{path}}(\ell,k) \geq t_{\text{walk}}(\ell,k) \geq f(\ell,k) \geq \ell^{k-1}$. Girão et al. [1] proved the general upper bound

$$t_{\text{walk}}(\ell,k) \leq \left(1 + \frac{2}{\sqrt{\ell}}\right)^k \ell^{k-1}$$

(Theorem 1.8 of [1]) and posed Conjecture 1.7 ($t_{\text{walk}}(\ell,k) = \ell^{k-1}$) together with Conjecture 5.3 ($t_{\text{path}}(\ell,k) = \ell^{k-1}$), remarking in Section 5 (and in Section 6.4 of Michel's 2026 DPhil thesis [3]) that Conjecture 1.7 remained wide open for all $\ell \geq 2$ and $k \geq 3$.

We settle the first two open non-trivial pairs $(2,4)$ and $(3,3)$ for both walks and paths simultaneously, closing the prior intervals $8 \leq t(2,4) \leq 271$ and $9 \leq t(3,3) \leq 90$ to exact equality.

Theorem 1 (Exact Threshold $t(2,4) = 8$ — Lean + SAT Certificate). *Every edge colouring of every 9-vertex tournament contains either a monochromatic directed 2-path or a rainbow directed 4-path (and hence a 2-flash or 4-rainbow walk), whereas the lexicographic grid tournament on $\{1,2\}^3$ (8 vertices) admits a 3-colouring avoiding both 2-flashes and 4-rainbows. Consequently,*

$$t_{\text{walk}}(2,4) = t_{\text{path}}(2,4) = 8,$$

settling Conjectures 1.7 and 5.3 of [1] for $(\ell,k) = (2,4)$.

Theorem 2 (Exact Threshold $t(3,3) = 9$ — Lean + SAT Certificate). *Every edge colouring of every 10-vertex tournament contains either a monochromatic directed 3-path or a rainbow directed 3-path (and hence a 3-flash or 3-rainbow walk), whereas the lexicographic grid tournament on $\{1,2,3\}^2$ (9 vertices) admits a 2-colouring avoiding both 3-flashes and 3-rainbows. Consequently,*

$$t_{\text{walk}}(3,3) = t_{\text{path}}(3,3) = 9,$$

settling Conjectures 1.7 and 5.3 of [1] for $(\ell,k) = (3,3)$.

Beside these two finite thresholds, we establish two general structural theorems unconditionally inside the pure Lean 4 kernel (with Mathlib), together with an exhaustive structural classification of good tournaments at $(2,3)$, $(2,4)$, and $(3,3)$. A *grid embedding* of an n-vertex tournament T into $\{1,\dots,\ell\}^d$ is an injective map $\varphi : V(T) \to \{1,\dots,\ell\}^d$ such that every arc $u \to v$ of T strictly increases at least one coordinate: $\varphi(u)_j < \varphi(v)_j$ for some $j \in \{1,\dots,d\}$. A *grid tournament* on $\{1,\dots,\ell\}^d$ is a tournament on ℓ^d vertices admitting a (necessarily bijective) grid embedding into $\{1,\dots,\ell\}^d$.

Theorem 3 (General Lexicographic Grid Lower Bound). *For all integers $\ell \geq 1$ and $k \geq 1$, the transitive lexicographic grid tournament on $\{1,\dots,\ell\}^{k-1}$ admits an edge colouring avoiding both ℓ-flashes and k-rainbows for directed walks (and hence directed paths). Thus*

$$t_{\text{path}}(\ell,k) \geq t_{\text{walk}}(\ell,k) \geq \ell^{k-1}.$$

Theorem 4 (Colour-Reduction and Grid-Embedding Equivalence). *For all integers $\ell \geq 1$ and $d \geq 1$, a finite tournament T admits an edge colouring using at most d colours that avoids ℓ-flashes and $(d+1)$-rainbows (for directed walks) if and only if T admits a grid embedding into $\{1,\dots,\ell\}^d$. In particular, any such tournament has at most ℓ^d vertices.*

Corollary 5 (Colour Reduction Implies Conjecture 1.7). *Fix integers $\ell \geq 1$ and $k \geq 2$. If every tournament admitting an (ℓ,k)-walk-good edge colouring over an arbitrary palette also admits one using at most $k-1$ colours, then $t_{\text{walk}}(\ell,k) = \ell^{k-1}$.*

Proposition 6 (Exhaustive Grid Classification for $(2,3)$, $(2,4)$, and $(3,3)$ — Cert Only). *For $(\ell,k) \in \{(2,3),(2,4),(3,3)\}$ and every order $1 \leq n \leq \ell^{k-1}$, a tournament on n vertices is (ℓ,k)-walk-good if and only if it embeds into a grid tournament on $\{1,\dots,\ell\}^{k-1}$. The exact isomorphism-class counts for $n = 1,\dots,\ell^{k-1}$ are $1,1,1,1$ for $(2,3)$; $1,1,2,4,11,43,75,43$ for $(2,4)$; and $1,1,2,4,11,39,116,185,107$ for $(3,3)$. At maximal order, the walk-good tournaments are precisely the 43 grid-tournament classes on $\{1,2\}^3$ and the 107 grid-tournament classes on $\{1,2,3\}^2$.*

Together, Proposition 6 and Theorem 4 motivate a structural strengthening of Conjecture 1.7 that remains open in general.

Conjecture 7 (Colour-Reduction Conjecture — Unsettled). *For all integers $\ell \geq 1$ and $k \geq 2$, every tournament admitting an (ℓ,k)-walk-good edge colouring over an arbitrary palette admits one using at most $k-1$ colours (equivalently, by Theorem 4, embeds into $\{1,\dots,\ell\}^{k-1}$).*

Figure 1 summarizes what was found, how each component was discovered, and the exact verification tier of each claim. The eight-line Python check below (verified verbatim by `recompute.py`) constructs and checks the lexicographic grid colourings witnessing $t(2,4) \geq 8$ and $t(3,3) \geq 9$.

Concept in Action: Lexicographic Grid Witnesses ($n = 8, 9$)

```
import itertools
for l, k in ((2, 4), (3, 3)):
    pts = list(itertools.product(range(1, l+1), repeat=k-1))
    col = {(u, v): next(i+1 for i in range(k-1)
                        if pts[v][i] > pts[u][i])
           for u in range(len(pts)) for v in range(u+1, len(pts))}
    assert len(pts) == l**(k-1) and max(col.values()) == k - 1
    assert all(pts[v][c-1] > pts[u][c-1] for (u, v), c in col.items())
```

Roadmap. Section 2 fixes exact conventions and states the Lean-proved tournament preliminaries. Section 3 proves Theorems 3 and 4, Corollary 5, and the exact small thresholds $t(\ell,2) = \ell$ and $t(2,3) = 4$. Section 4 describes the palette-free SAT encoding, symmetry-breaking reductions, external DRAT/LRAT certificate checks, and exhaustive grid classification completing Theorems 1 and 2 and Proposition 6. Section 5 presents the machine-generated Lean 4 Trust Boundary. Sections 6 and 7 review prior work, open problems, reproducibility, and AI assistance.

2. DEFINITIONS, NOTATION, AND PRELIMINARIES

A *tournament* T on vertex set $V = \{0,\dots,n-1\}$ is a complete, irreflexive, asymmetric binary relation $u \to v$ (for every distinct pair $u \neq v$, exactly one of $u \to v$ or $v \to u$ holds). An *edge colouring* of T is a map $c : E(T) \to C$ into an arbitrary colour set C. A *directed walk* of length $m \geq 0$ is a sequence of vertices $(w_0, w_1, \dots, w_m)$ with $w_i \to w_{i+1}$ for all $0 \leq i < m$; repeated vertices and arcs are permitted (Definition 1.6 of [1]). A *directed path* of length m is a directed walk whose $m+1$ vertices are pairwise distinct.

For integers $\ell, k \geq 1$, an *ℓ-flash* is a directed walk of length ℓ whose ℓ arcs all receive the same colour, and a *k-rainbow* is a directed walk of length k whose k arcs receive k pairwise distinct colours. A colouring c of T is *(ℓ,k)-walk-good* if it contains neither an ℓ-flash walk nor a k-rainbow walk, and *(ℓ,k)-path-good* if it contains neither a monochromatic directed ℓ-path nor a rainbow directed k-path. A tournament T is *walk-good* (respectively *path-good*) for (ℓ,k) if it admits such a colouring, and *bad* otherwise. Define

$$t_{\text{walk}}(\ell,k) = \max\{n : \exists\ (\ell,k)\text{-walk-good } n\text{-vertex } T\},$$
$$t_{\text{path}}(\ell,k) = \max\{n : \exists\ (\ell,k)\text{-path-good } n\text{-vertex } T\}.$$

Because deleting vertices from a good tournament preserves goodness (*heredity*), $t_{\text{walk}}(\ell,k)$ coincides with Girão et al.'s definition as the least t such that every colouring of every

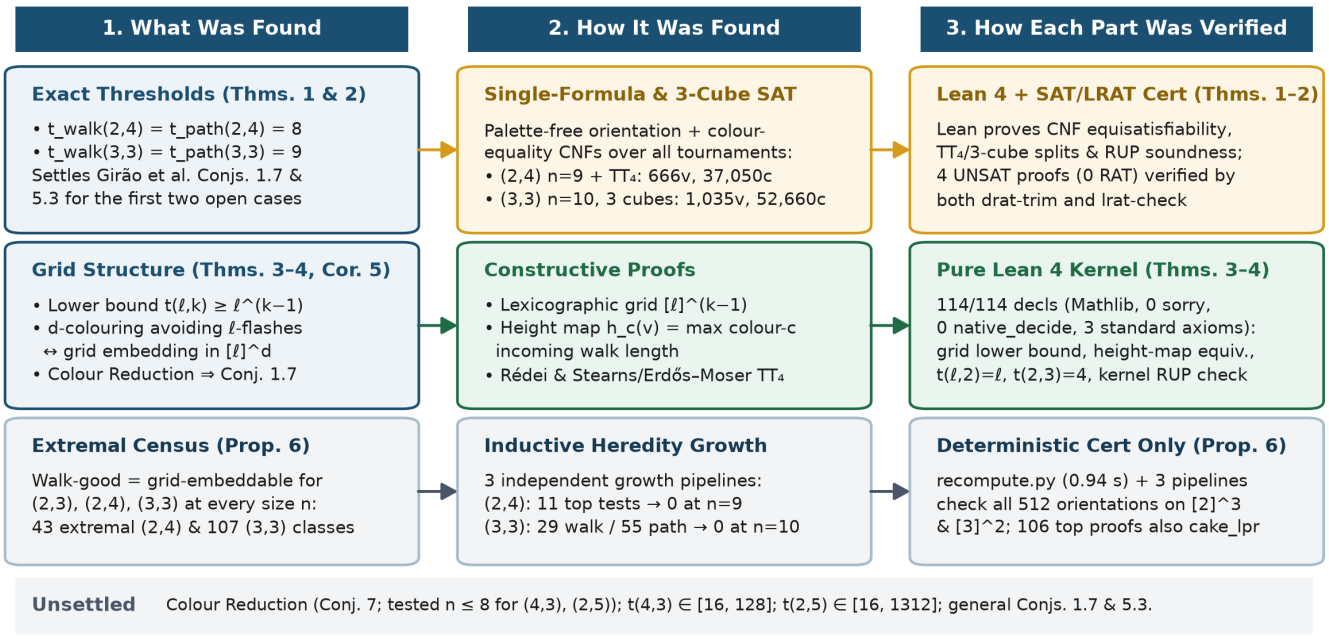


Figure 1. Discovery and verification architecture separating exact findings (Column 1), computational and constructive methods (Column 2), and formal verification tiers (Column 3), with unsettled problems listed in the flat footer strip. Pure Lean 4 + Mathlib proofs cover 114/114 audited declarations; external SAT certificates checked by `drat-trim` and `lrat-check` supply the upper-bound refutations for $(2,4)$ at $n = 9$ and $(3,3)$ at $n = 10$.

tournament on more than t vertices contains an ℓ-flash or k-rainbow.

Parameter convention in Lean 4. Throughout our Lean 4 library `lean/TournamentEdgeColor/`, the ordered pair is written (k, l) with k denoting the forbidden monochromatic length and l denoting the forbidden rainbow length; thus Lean's `t_walk k l` and `t_path k l` match $t_{\text{walk}}(\ell, k)$ and $t_{\text{path}}(\ell, k)$ via $(k_{\text{Lean}}, l_{\text{Lean}}) = (\ell, k)$ (for instance, `t_walk 2 4 = 8` and `t_walk 3 3 = 9`).

Basic structural properties. The following three properties are proved unconditionally in `Defs.lean`:

1. *Walk–path sandwich:* Walk-good implies path-good, so $t_{\text{walk}}(\ell, k) \le t_{\text{path}}(\ell, k)$. Exhibiting an M-vertex (ℓ, k)-walk-good tournament while ruling out any $(M+1)$-vertex (ℓ, k)-path-good tournament proves $t_{\text{walk}}(\ell, k) = t_{\text{path}}(\ell, k) = M$ at once.
2. *Finite palettes:* On an n-vertex tournament, avoidance depends only on colour equality among arc pairs, so compressing any palette C onto the $\binom{n}{2} = n(n-1)/2$ pairs loses zero generality.
3. *Classical substructures:* Every tournament on $n \ge 1$ vertices has a Hamiltonian directed path (Rédei [4], `[to check]`; proved in `Defs.lean`), and every tournament on $n \ge 2^{m-1}$ vertices contains a transitive TT_m (Stearns [5], `[to check]`; proved in `Defs.lean`), yielding TT_3 for $n \ge 4$ and TT_4 for $n \ge 8$. For $n \ge 10$, averaging out-degrees gives a vertex v_0 of out-degree at least $\lceil (n-1)/2 \rceil \ge 5$; its out-neighbourhood contains a TT_3 on $\{v_1, v_2, v_3\}$ plus two vertices $\{v_4, v_5\}$ with $v_4 \to v_5$, giving a TT_4 on $\{v_0, v_1, v_2, v_3\}$ with $v_0 \to v_4$, $v_0 \to v_5$, and $v_4 \to v_5$.

3. CONSTRUCTIVE PROOFS AND SMALL THRESHOLDS

We first prove the general lower bound (Theorem 3), the height-map characterization of bounded-palette colourings (Theorem 4), and its implication for Conjecture 1.7 (Corollary 5).

Proof of Theorem 3. Fix integers $\ell \ge 1$ and $k \ge 1$, and set $d = k - 1$. If $d = 0$, the single-vertex tournament on $\ell^0 = 1$ vertex has no arcs and trivially avoids every walk of positive length. Assume $d \ge 1$, and identify the vertex set with the grid $V = \{1, \dots, \ell\}^d$ of cardinality ℓ^{k-1}. Orient every distinct pair $u \ne v$ by the strict lexicographic order $u <_{\text{lex}} v$ (that is, $u_j < v_j$ at the least coordinate $j \in \{1, \dots, d\}$ where u and v differ), and assign arc $u \to v$ the colour $c(u, v) = j \in \{1, \dots, d\}$.

This colouring uses only $d = k - 1$ colours, so by the pigeonhole principle no directed walk of k arcs can carry k pairwise distinct colours (no k-rainbow). Along any monochromatic directed walk $w_0 \to w_1 \to \dots \to w_m$ of colour j, the j-th coordinate strictly increases at every step: $(w_0)_j < (w_1)_j < \cdots < (w_m)_j$ inside $\{1, \dots, \ell\}$. Because $\{1, \dots, \ell\}$ contains only ℓ values, $m + 1 \le \ell$, hence $m \le \ell - 1$. Thus no ℓ-flash exists. Therefore the lexicographic grid tournament is (ℓ, k)-walk-good (and hence path-good), giving $t_{\text{path}}(\ell, k) \ge t_{\text{walk}}(\ell, k) \ge \ell^{k-1}$. □

Proof of Theorem 4. Fix $\ell \ge 1$ and $d \ge 1$, and let T be an n-vertex tournament.

($\Leftarrow$) Suppose $\varphi : V(T) \to \{1, \dots, \ell\}^d$ is a grid embedding of T. For each arc $u \to v$, choose a coordinate $c(u, v) \in \{1, \dots, d\}$

with $\varphi(u)_{c(u,v)} < \varphi(v)_{c(u,v)}$. Exactly as in the proof of Theorem 3, using only d colours forbids every $(d+1)$-rainbow walk, while strict coordinate ascent along colour j limits every monochromatic walk to length at most $\ell - 1$, forbidding every ℓ-flash walk.

($\Rightarrow$) Conversely, let $c : E(T) \to \{1, \dots, d\}$ be a colouring of T with no ℓ-flash walk. For each colour $j \in \{1, \dots, d\}$ and each vertex $v \in V(T)$, define the *height*

$$h_j(v) = 1 + \max\{m \geq 0 : \exists \text{ colour-}j\ m\text{-walk ending at } v\}.$$

The zero-length walk (v) shows the set is nonempty, and absence of an ℓ-flash forces $m \leq \ell - 1$, so $h_j(v) \in \{1, \dots, \ell\}$ is well-defined. Set $\varphi(v) = (h_1(v), \dots, h_d(v)) \in \{1, \dots, \ell\}^d$. If $u \to v$ has colour j, appending $u \to v$ to a longest colour-j walk ending at u yields a colour-j walk of length $h_j(u)$ ending at v, so $h_j(u) < h_j(v)$. Thus every arc strictly increases the coordinate of its colour. Finally, for any two distinct vertices $u \neq v$, one of $u \to v$ or $v \to u$ is an arc of T, so $\varphi(u)$ and $\varphi(v)$ differ in the coordinate of that arc's colour. Hence φ is injective and is a grid embedding of T into $\{1, \dots, \ell\}^d$. Injectivity immediately gives $n \leq \ell^d$. □

Proof of Corollary 5. Fix $\ell \geq 1$ and $k \geq 2$, and set $d = k - 1 \geq 1$. Theorem 3 gives $t_{\text{walk}}(\ell, k) \geq \ell^{k-1}$. Conversely, if an n-vertex tournament T is (ℓ, k)-walk-good, the Colour-Reduction hypothesis yields an (ℓ, k)-walk-good colouring of T with $d = k - 1$ colours, whence Theorem 4 gives $n \leq \ell^d = \ell^{k-1}$. Thus $t_{\text{walk}}(\ell, k) = \ell^{k-1}$. □

As warm-up sanity checks (all proved unconditionally in `lean/TournamentEdgeColor/GridLowerBound.lean`, along with the trivial boundary cases $t(\ell, 1) = t(1, k) = 1$), we record the exact laws for $k = 2$ and $(\ell, k) = (2, 3)$.

Proposition 8 (Exact $k = 2$ Law). *For every integer $\ell \geq 1$, $t_{\text{walk}}(\ell, 2) = t_{\text{path}}(\ell, 2) = \ell$.*

Proof. Theorem 3 gives $t_{\text{walk}}(\ell, 2) \geq \ell$. For the upper bound, let T be any edge-coloured tournament on $\ell + 1$ vertices. By Rédei's theorem, T has a Hamiltonian directed path $v_0 \to v_1 \to \dots \to v_\ell$ of length ℓ. If all ℓ arcs along this path share one colour, it is a monochromatic directed ℓ-path; otherwise two consecutive arcs $v_i \to v_{i+1} \to v_{i+2}$ have distinct colours, forming a rainbow directed 2-path. The walk–path sandwich finishes the proof. □

Proposition 9 (Exact $(2, 3)$ Law). *For both walks and paths, $t_{\text{walk}}(2, 3) = t_{\text{path}}(2, 3) = 4$.*

Proof. Theorem 3 gives $t_{\text{walk}}(2, 3) \geq 2^2 = 4$. For the stronger path upper bound, suppose for contradiction that a 5-vertex tournament T has a $(2, 3)$-path-good colouring c. Two rules follow immediately: (i) along any directed 2-path $x \to y \to z$, $c(x, y) \neq c(y, z)$ (no monochromatic 2-path); (ii) along any directed 3-path $a \to b \to d \to e$, $c(a, b) = c(d, e)$ (since consecutive arcs differ by (i) and three distinct colours would form a rainbow 3-path).

By Rédei's theorem, choose a Hamiltonian directed path $v_0 \to v_1 \to v_2 \to v_3 \to v_4$. Set $A = c(v_0, v_1)$ and $B = c(v_1, v_2)$, so $A \neq B$ by (i). Rule (ii) on $v_0 \to v_1 \to v_2 \to v_3$ and $v_1 \to v_2 \to v_3 \to v_4$ forces $c(v_2, v_3) = A$ and $c(v_3, v_4) = B$. Now orient the chords step by step:

- If $v_1 \to v_3$, then rule (ii) on $v_0 \to v_1 \to v_3 \to v_4$ forces $A = c(v_0, v_1) = c(v_3, v_4) = B$, contradiction. Hence $v_3 \to v_1$. Let $C = c(v_3, v_1)$; rule (i) on $v_2 \to v_3 \to v_1$ and $v_3 \to v_1 \to v_2$ forces $C \neq A$ and $C \neq B$.
- If $v_0 \to v_2$, rule (ii) on $v_0 \to v_2 \to v_3 \to v_1$ and $v_0 \to v_2 \to v_3 \to v_4$ gives $c(v_0, v_2) = C$ and $c(v_0, v_2) = B$, contradicting $C \neq B$. Hence $v_2 \to v_0$.
- If $v_2 \to v_4$, rule (ii) on $v_3 \to v_1 \to v_2 \to v_4$ and $v_0 \to v_1 \to v_2 \to v_4$ gives $c(v_2, v_4) = C$ and $c(v_2, v_4) = A$, contradicting $C \neq A$. Hence $v_4 \to v_2$.

Finally, rule (ii) on $v_4 \to v_2 \to v_0 \to v_1$ gives $c(v_4, v_2) = A$, while rule (ii) on $v_4 \to v_2 \to v_3 \to v_1$ gives $c(v_4, v_2) = C$, contradicting $C \neq A$. Thus no 5-vertex tournament is $(2, 3)$-path-good, and the walk–path sandwich gives $t_{\text{walk}}(2, 3) = t_{\text{path}}(2, 3) = 4$. □

4. SAT ENCODING, SYMMETRY BREAKING, AND COMPUTATIONAL VERIFICATION

At $n = 9$ for $(2, 4)$ and $n = 10$ for $(3, 3)$, the numbers of non-isomorphic tournaments reach 191,536 and 9,733,056 (OEIS A000568), and no short proof like Proposition 9 was found. We settled both upper bounds by two independent computational routes: (A) a single palette-free SAT formula covering all labelled tournaments at once (with Lean-proved equisatisfiability, symmetry breaking, and externally checked DRAT/LRAT proofs), and (B) three independent heredity-growth pipelines cross-checked against exhaustive grid-tournament enumeration.

Palette-Free CNF Encoding and Lean Soundness Bridge. For integers $n \geq 1$ and $\ell, k \geq 1$, we define a propositional CNF formula tourneyPathCNF(n, ℓ, k) over two finite variable families (formalized verbatim in `SATCert.lean`):

- *Orientation variables* orient(i, j) for $0 \leq i < j < n$, where orient(i, j) = true encodes the arc $i \to j$ and false encodes $j \to i$ ($\binom{n}{2}$ variables).
- *Colour-equality variables* same(e_1, e_2) for unordered pairs of distinct edges $e_1 < e_2$ among the $\binom{n}{2}$ pairs, encoding whether arcs e_1 and e_2 share a colour ($\binom{\binom{n}{2}}{2}$ variables).

For $n = 9$ this yields $\binom{9}{2} + \binom{36}{2} = 36 + 630 = 666$ variables; for $n = 10$ it yields $\binom{10}{2} + \binom{45}{2} = 45 + 990 = 1{,}035$ variables. Three clause families enforce an exact (ℓ, k)-path-good coloured tournament:

1. *Equivalence transitivity:* For every triple of distinct unordered pairs $e_1 < e_2 < e_3$, the three clauses
$$\neg\text{same}(e_1, e_2) \vee \neg\text{same}(e_2, e_3) \vee \text{same}(e_1, e_3)$$
(and its two permutations) ensure that same defines an equivalence relation on arcs (recovering a concrete colouring via the quotient setoid, without restricting the palette size).
2. *No monochromatic directed ℓ-path:* For every sequence of $\ell + 1$ pairwise distinct vertices $(v_0, \dots, v_\ell)$, one clause forbids all ℓ arcs $v_i \to v_{i+1}$ from being present while every adjacent arc pair satisfies same(e_i, e_{i+1}).
3. *No rainbow directed k-path:* For every sequence of $k + 1$ pairwise distinct vertices $(v_0, \dots, v_k)$, one clause forbids

Table 1. Four independently verified UNSAT certificates establishing $t_{\text{path}}(2,4) \leq 8$ and $t_{\text{path}}(3,3) \leq 9$. Every CNF matches its generator byte-for-byte; every core proof uses 0 RAT lemmas (pure RUP) and passes both `drat-trim` [6] (`[to check]`) and `lrat-check` [7] (`[to check]`).

Formula	Vars	Clauses	CNF SHA-256	Core / Total Lemmas	Res. Steps	LRAT (MB)	`drat-trim`	`lrat-check`
`s_2_4_9_path`	666	37,050	`c2c268bde822`	628,600/937,845	23,468,170	194.9	121.59 s	11.24 s
`c33A`	1,035	52,660	`927f758a9f5e`	328,584/568,809	10,568,793	91.5	80.99 s	3.49 s
`c33B`	1,035	52,660	`afe54228037d`	247,831/417,845	7,302,917	64.9	68.56 s	7.24 s
`c33C`	1,035	52,660	`37a91373af95`	952,729/1,467,806	32,479,591	292.7	401.23 s	7.35 s

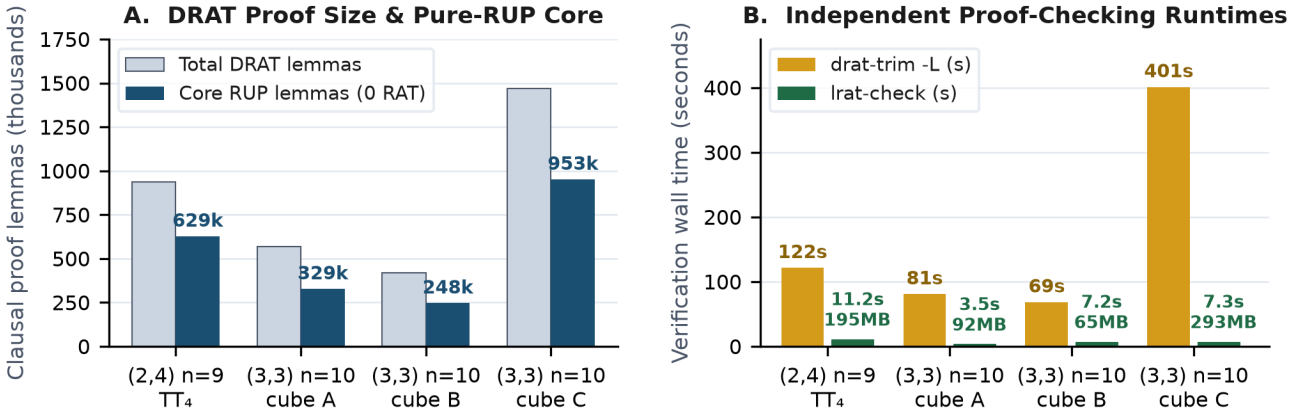


Figure 2. Quantitative summary of the four UNSAT certificates in Table 1. Panel A compares total DRAT lemmas against trimmed core lemmas (every core contains 0 RAT lemmas, confirming pure RUP derivations). Panel B compares `drat-trim -L` trimming/checking wall time against independent linear-time `lrat-check` verification across the 64.9–292.7 MB LRAT files.

all k arcs $v_i \to v_{i+1}$ from being present while every pair of arcs e_r, e_s $(0 \leq r < s < k)$ satisfies $\neg\text{same}(e_r, e_s)$.

Proposition 10 (Exact CNF Equisatisfiability). *For all integers $n \geq 0$, $\ell \geq 1$, and $k \geq 0$, there exists an n-vertex (ℓ, k)-path-good edge-coloured tournament if and only if* tourneyPathCNF(n, ℓ, k) *is satisfiable.*

Proposition 10 is proved in *both* directions in `SATCert.lean` (Section 5). The converse constructs an explicit tournament T and quotient-setoid colouring c from any satisfying assignment, ruling out over-constraining encoding bugs. In addition, `SATCert.lean` proves injective variable-renaming invariance (bridging structured variables to DIMACS integer IDs), proves kernel soundness of a Reverse Unit Propagation (RUP) proof checker, and executes a pure-kernel `rfl` 8-step RUP refutation of the $(3,2,2)$ instance without `native_decide`.

Symmetry Breaking and Four Verified UNSAT Certificates. To make the $n = 9$ and $n = 10$ formulas tractable without losing any tournament or colouring, we adjoin Lean-verified symmetry-breaking unit clauses:

- *Case $(2,4)$ at $n = 9$:* By Stearns's theorem ($n = 9 \geq 2^{4-1} = 8$), every 9-vertex tournament contains a TT_4. Relabelling that TT_4 to $\{0,1,2,3\}$ adjoins the six unit clauses $i \to j$ $(0 \leq i < j \leq 3)$, forming the formula `tourneyPathCNF_TT4` at $(9,2,4)$ (666 variables, 37,050 clauses; proved sound in `not_existsAvoidingPath_of_unsat_TT4`).
- *Case $(3,3)$ at $n = 10$:* By our $n \geq 10$ out-degree lemma (`exists_TT4_and_out2_of_ge_10`), every 10-vertex tournament relabels so that $\{0,1,2,3\}$ induces TT_4 and $0 \to 4$, $0 \to 5$, $4 \to 5$. Along the directed 3-path $0 \to 1 \to 2 \to 3$, avoiding a rainbow 3-path forces at least one of $c(0,1) = c(1,2)$ (Cube A), $c(1,2) = c(2,3)$ (Cube B), or $c(0,1) = c(2,3)$ (Cube C). Each cube has 1,035 variables and 52,660 clauses; exhaustiveness is proved in `not_existsAvoidingPath_of_unsat_cube33`.

Table 1 and Figure 2 record the verified certificate receipts from `docs/shadow/round19/R19-3/lrat-check/lrat-check.md`. Each CNF was solved to UNSAT, its DRAT refutation was checked and trimmed to LRAT by `drat-trim` [6] (`[to check]`, exiting 0 with `s VERIFIED` and `0 RAT lemmas in core`), and the LRAT proof was independently checked by `lrat-check` [7] (`[to check]`, exiting 0 with `c VERIFIED`). Because every core proof is pure RUP (zero RAT lemmas), these certificates discharge the RUP hypotheses of `t_2_4_exact_of_tt4_rup` and `t_3_3_exact_of_cube33_rup` in Lean 4 (noting that the 64.9–292.7 MB LRAT proofs are checked externally rather than inside Lean's kernel, and that the formally verified CakeML checker `cake_lpr` [8] [`[to check]`] ran on the 106 top-level proofs of the independent growth route below rather than on these four files). Sanity runs of the same generator at $n = 8$ for $(2,4)$ and $n = 9$ for $(3,3)$ return SAT and decode into verified avoiding tournaments. Combining Theorem 3 with these four verified UNSAT certificates proves Theorems 1 and 2.

Inductive Heredity Growth and Grid Classification. Independently of the single-formula SAT route, three separate heredity-growth pipelines (two in Python using dis-

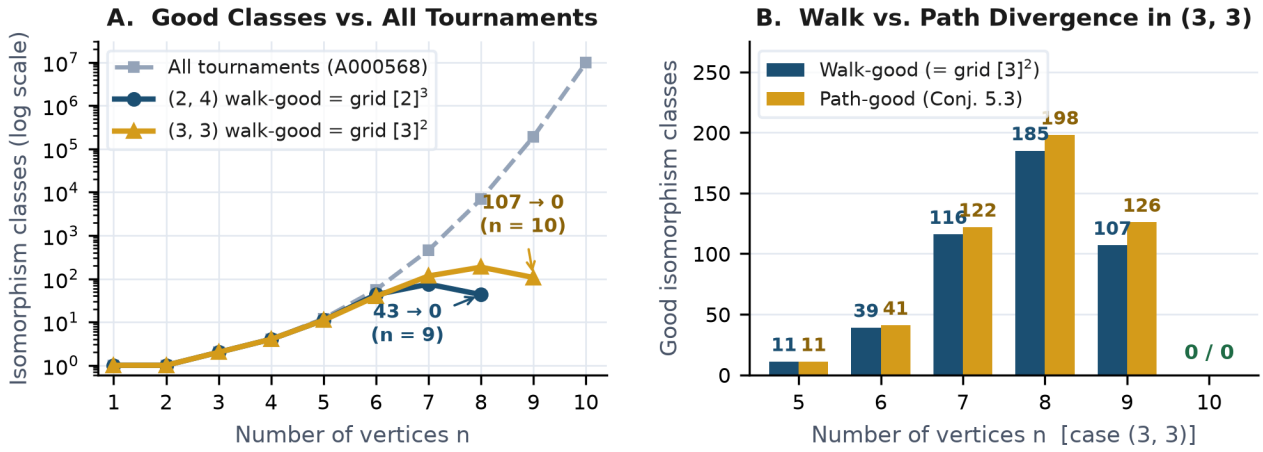


Figure 3. Exhaustive isomorphism-class counts by vertex order n (Table 2). Panel A contrasts the super-exponential growth of all tournaments (OEIS A000568) against the peak and abrupt collapse of walk-good (= grid-embeddable) classes for $(2,4)$ ($43 \to 0$ at $n = 9$) and $(3,3)$ ($107 \to 0$ at $n = 10$). Panel B shows the walk-versus-path divergence in $(3,3)$ starting at $n = 6$ (39 versus 41) before both drop to 0 at $n = 10$.

tinct canonicalizers and SAT encodings, and one in C) grew all isomorphism classes of good tournaments one vertex at a time from $n = 1$, discarding any n-vertex candidate that possessed a bad $(n-1)$-vertex induced subtournament and deciding the remaining candidates by SAT with DRAT checking. For $(2,4)$, heredity reduced the 191,536 tournament classes on 9 vertices to 7,024 one-vertex extensions and only 11 heredity-surviving candidates, all UNSAT. For $(3,3)$, heredity reduced the 9,733,056 tournament classes on 10 vertices to 40,690 (walks) and 43,302 (paths) extensions and only 29 (walks) and 55 (paths) heredity-surviving candidates, all UNSAT. All $11+11+29+55 = 106$ top-level UNSAT proofs from an independent palette-free growth pipeline were additionally converted to LRAT and verified with zero failures by the formally verified checker `cake_lpr` [8] (`[to check]`).

Table 2. Isomorphism classes of all tournaments (OEIS A000568), walk-good (= grid-embeddable) classes for $(2,4)$ and $(3,3)$, and path-good classes for $(3,3)$.

n	All (A000568)	$(2,4)$ walk/grid	$(3,3)$ walk/grid	$(3,3)$ path
1	1	1	1	1
2	1	1	1	1
3	2	2	2	2
4	4	4	4	4
5	12	11	11	11
6	56	43	39	41
7	456	75	116	122
8	6,880	43	185	198
9	191,536	0	107	126
10	9,733,056	0	0	0

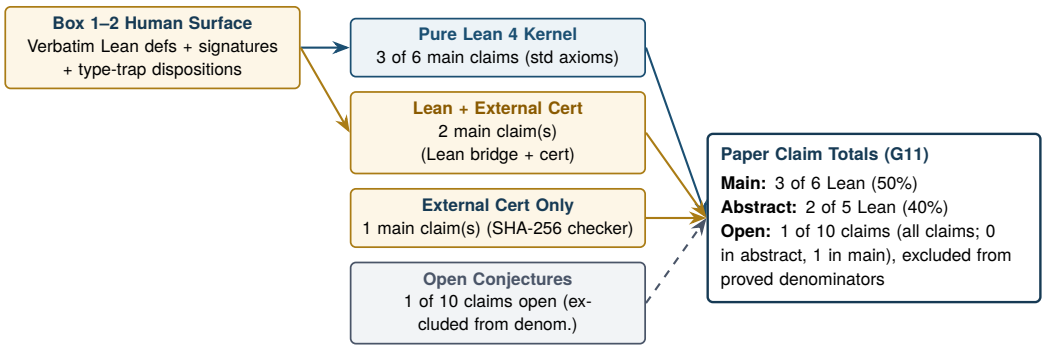
To establish Proposition 6, `recompute.py` (confirming two independent enumerators) enumerates every orientation of the free pairs in $[2]^2$ (5 forced, 1 free, 2 orientations), $[2]^3$ (19 forced, 9 free, 512 orientations), and $[3]^2$ (27 forced, 9 free, 512 orientations) across all vertex subsets, canonicalizes each tournament (verified against OEIS A000568 through $n = 5$: $1,1,2,4,12$), and checks that every grid class is walk-good (0 failures). Comparing canonical class sets against the walk-good lists from the growth pipelines confirms exact set equality at every $n \le \ell^{k-1}$ (Table 2, Figure 3). Already at $n = 5$, exactly one of the 12 tournament classes is bad for both $(2,4)$ and $(3,3)$: the unique regular 5-vertex tournament with score sequence $(2,2,2,2,2)$. For $(3,3)$ paths, grid equality fails starting at $n = 6$ (41 path-good versus 39 grid classes, up to 126 versus 107 at $n = 9$), because a monochromatic directed 3-cycle creates a 3-flash walk without creating a 3-flash path; nevertheless both sequences collapse to 0 at $n = 10$.

5. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel & Toolchain:** leanprover/lean4:v4.35.0; committed axiom receipt anchored to git commit 2f09f46076a2 (independent gate-box kernel replay pending); standard axioms only (propext, Classical.choice, Quot.sound).
2. **Core Definitions & Fidelity (Boxes 1–2):** Verify that the verbatim Lean definitions in Box 1 match Section 2 and that each Lean signature in Box 2 matches its stated theorem.
3. **External Certificates & Written Proofs:** 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: 114/114 public theorems (source-level approx) | WARN: Fidelity: 8/8 exact (draft cross-family)
WARN: Witnesses: pending gate replay | WARN: Human referee: pending

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

TournamentEdgeColor.AvoidsPath (TournamentEdgeColor/Defs.lean:114-115) — Section 2 (path-good colouring avoiding flash and rainbow)

```
def AvoidsPath (k l : ℕ) {n : ℕ} {C : Type*} (T : Tournament n) (c : Fin n → Fin n →
  ↪ C) : Prop :=
  ¬ HasMonoPath T c k ∧ ¬ HasRainbowPath T c l
```

TournamentEdgeColor.AvoidsWalk (TournamentEdgeColor/Defs.lean:109-110) — Section 2 (walk-good colouring avoiding flash and rainbow)

```
def AvoidsWalk (k l : ℕ) {n : ℕ} {C : Type*} (T : Tournament n) (c : Fin n → Fin n →
  ↪ C) : Prop :=
  ¬ HasMonoWalk T c k ∧ ¬ HasRainbowWalk T c l
```

TournamentEdgeColor.ColorReductionHolds (TournamentEdgeColor/GridLowerBound.lean:400-403) — Section 1 (Colour-Reduction property at (l, k))

```
def ColorReductionHolds (k l : ℕ) : Prop :=
  ∀ {n : ℕ} (T : Tournament n),
    (∃ (C : Type) (c : Fin n → Fin n → C), AvoidsWalk k l T c) →
    ∃ c' : Fin n → Fin n → Fin (l - 1), AvoidsWalk k l T c'
```

TournamentEdgeColor.DirPath (TournamentEdgeColor/Defs.lean:84-85) — Section 2 (directed path of length m with injective vertices)

```
structure DirPath {n : ℕ} (T : Tournament n) (m : ℕ) extends DirWalk T m where
  inj : Function.Injective vert
```

TournamentEdgeColor.DirWalk (TournamentEdgeColor/Defs.lean:79-81) — Section 2 (directed walk of length m with m+1 vertices)

```
structure DirWalk {n : ℕ} (T : Tournament n) (m : ℕ) where
  vert : Fin (m + 1) → Fin n
  step : ∀ i : Fin m, T.edge (vert i.castSucc) (vert i.succ)
```

TournamentEdgeColor.HasMonoWalk (TournamentEdgeColor/Defs.lean:89-90) — Section 2 (monochromatic directed walk of length l)

```
def HasMonoWalk {n : ℕ} {C : Type*} (T : Tournament n) (c : Fin n → Fin n → C) (k :
  ↪ ℕ) : Prop :=
  ∃ (w : DirWalk T k) (col : C), ∀ i : Fin k, c (w.vert i.castSucc) (w.vert i.succ) =
    ↪ col
```

TournamentEdgeColor.HasRainbowWalk (TournamentEdgeColor/Defs.lean:98-100) — Section 2 (rainbow directed walk of length k)

```
def HasRainbowWalk {n : ℕ} {C : Type*} (T : Tournament n) (c : Fin n → Fin n → C) (l
  ↪ : ℕ) : Prop :=
  ∃ w : DirWalk T l,
    ∀ i j : Fin l, i ≠ j → c (w.vert i.castSucc) (w.vert i.succ) ≠ c (w.vert
      ↪ j.castSucc) (w.vert j.succ)
```

TournamentEdgeColor.IsGridEmbedding (TournamentEdgeColor/GridLowerBound.lean:50-51) — Section 1 (injective map into [l]^d increasing >= 1 coordinate per arc)

```
def IsGridEmbedding {n : ℕ} (T : Tournament n) (k d : ℕ) (φ : Fin n → GridPoint k d)
  ↪ : Prop :=
  Function.Injective φ ∧ ∀ u v : Fin n, T.edge u v → ∃ j : Fin d, (φ u j).val < (φ v
    ↪ j).val
```

TournamentEdgeColor.Tournament (TournamentEdgeColor/Defs.lean:51-55) — Section 2 (complete, irreflexive, asymmetric directed graph on Fin n)

```
structure Tournament (n : ℕ) where
  edge : Fin n → Fin n → Prop
  irrefl : ∀ u, ¬ edge u u
  asym : ∀ u v, edge u v → ¬ edge v u
  complete : ∀ u v, u ≠ v → edge u v ∨ edge v u
```

TournamentEdgeColor.t_path (TournamentEdgeColor/Defs.lean:587-588) — Section 2 (path threshold function t_path)

```
noncomputable def t_path (k l : ℕ) : ℕ :=
  sSup { n : ℕ | ExistsAvoidingPath n k l }
```

TournamentEdgeColor.t_walk (TournamentEdgeColor/Defs.lean:583-584) — Section 2 (walk threshold function t_walk)

```
noncomputable def t_walk (k l : ℕ) : ℕ :=
  sSup { n : ℕ | ExistsAvoidingWalk n k l }
```

Supporting Semantic Closure Index (non-Box-1 helpers): TournamentEdgeColor.CNF (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.Clause (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.ClauseStatus (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.Eval (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.ExistsAvoidingPath (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.ExistsAvoidingWalk (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.GridPoint (TournamentEdgeColor/GridLowerBound.lean), TournamentEdgeColor.HasGridEmbedding (TournamentEdgeColor/GridLowerBound.lean), TournamentEdgeColor.HasMonoPath (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.HasRainbowPath (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.IsExactPathThreshold (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.IsExactWalkThreshold (TournamentEdgeColor/Defs.lean), TournamentEdgeColor.Lit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.Lit.negate (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.PassResult (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.Satisfies (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.SatisfiesClause (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.TourneyVar (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.adjNotSameLits (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.arcLit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.checkRUPProof (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.classifyClause (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.cube33A (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.cube33B (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.cube33Base (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.cube33C (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.distinctSeqs (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.edgeCodes (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.evalLit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.hasContradiction (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.isRUP (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.monoPathClause (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.noMonoPathClauses (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.noRainbowPathClauses (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.notArcLit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.notSameLit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.pairCode (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.pairSameLits (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.pathArcNegLits (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.pathEdges (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.rainbowPathClause (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.rupLoop (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.rupPassGo (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.sameLit (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.tourneyPathCNF (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.tourneyPathCNF_TT4 (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.transClauses (TournamentEdgeColor/SATCert.lean), TournamentEdgeColor.tt4Clauses (TournamentEdgeColor/SATCert.lean).

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

TournamentEdgeColor.t_2_4_exact_of_tt4_rup (TournamentEdgeColor/SATCert.lean:1457-1463) src:b2dc9778285df2

```
theorem t_2_4_exact_of_tt4_rup
    (h_rup_tt4 : ∃ proof, checkRUPProof (tourneyPathCNF_TT4 9 2 4) proof = true) :
    IsExactWalkThreshold 2 4 8 ∧ IsExactPathThreshold 2 4 8 ∧
      t_walk 2 4 = 8 ∧ t_path 2 4 = 8
```

Assumptions: External RUP certificate hypothesis h_rup_tt4 for tourneyPathCNF_TT4 9 2 4 (discharged externally by s_2_4_9_path.cnf and drat-trim/lrat-check logs with 0 RAT core lemmas); lower bound t(2,4) >= 8 and TT_4 reduction are unconditional in Lean. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

TournamentEdgeColor.t_3_3_exact_of_cube33_rup (TournamentEdgeColor/SATCert.lean:1488-1501) src:79bf6e6cae1461

```
theorem t_3_3_exact_of_cube33_rup
    (h_rup_A : ∃ proof, checkRUPProof (cube33A 10) proof = true)
    (h_rup_B : ∃ proof, checkRUPProof (cube33B 10) proof = true)
    (h_rup_C : ∃ proof, checkRUPProof (cube33C 10) proof = true) :
    IsExactWalkThreshold 3 3 9 ∧ IsExactPathThreshold 3 3 9 ∧
      t_walk 3 3 = 9 ∧ t_path 3 3 = 9
```

Assumptions: Three external RUP certificate hypotheses h_rup_A, h_rup_B, h_rup_C for cube33A/B/C 10 (discharged externally by c33A/B/C CNFs and drat-trim/lrat-check logs with 0 RAT core lemmas); lower bound t(3,3) >= 9 and 3-cube split are unconditional in Lean. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

TournamentEdgeColor.lexGrid_lower_bound (TournamentEdgeColor/GridLowerBound.lean:233-238) src:f76bc4679233c2

```
theorem lexGrid_lower_bound {k l : ℕ} (hk : 1 ≤ k) (hl : 1 ≤ l) :
    ExistsAvoidingWalk (k ^ (l - 1)) k l
```

Assumptions: Any natural numbers k >= 1 and l >= 1 (matching l_paper >= 1, k_paper >= 1). Proved unconditionally via lexGrid k (l - 1). Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

TournamentEdgeColor.bounded_palette_iff_gridEmbedding (TournamentEdgeColor/GridLowerBound.lean:383-387) src:8b3d53de286090

```
theorem bounded_palette_iff_gridEmbedding {n k d : ℕ} (hk : 1 ≤ k) (hd : 1 ≤ d)
    (T : Tournament n) :
    (∃ c : Fin n → Fin n → Fin d, AvoidsWalk k (d + 1) T c) ↔ HasGridEmbedding T k d
```

Assumptions: Any natural numbers n, k >= 1, d >= 1 and any n-vertex tournament T. Proved unconditionally via longest-incoming-walk height map. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

TournamentEdgeColor.cr_implies_conj_1_7 (TournamentEdgeColor/GridLowerBound.lean:407-416) src:1138508fc7b0aa

```
theorem cr_implies_conj_1_7 {k l : ℕ} (hk : 1 ≤ k) (hl : 2 ≤ l)
    (hCR : ColorReductionHolds k l) :
    IsExactWalkThreshold k l (k ^ (l - 1)) ∧ t_walk k l = k ^ (l - 1)
```

Assumptions: Any natural numbers k >= 1, l >= 2 satisfying ColorReductionHolds k l (conditional on the open Colour-Reduction property at (k, l); known to hold for (2,3), (2,4), (3,3)). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: ordinary_type).

TournamentEdgeColor.exact_k_2 (TournamentEdgeColor/GridLowerBound.lean:514-517) src:74e9d6db4444db

```
theorem exact_k_2 {k : ℕ} (hk : 1 ≤ k) :
    IsExactWalkThreshold k 2 k ∧ IsExactPathThreshold k 2 k ∧
      t_walk k 2 = k ∧ t_path k 2 = k
```

Assumptions: Any natural number k >= 1 (matching l_paper >= 1). Proved unconditionally via Redei Hamiltonian path theorem. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

TournamentEdgeColor.exact_2_3 (TournamentEdgeColor/GridLowerBound.lean:662-665) src:1be6b2bfa17fc4

```
theorem exact_2_3 :
    IsExactWalkThreshold 2 3 4 ∧ IsExactPathThreshold 2 3 4 ∧
      t_walk 2 3 = 4 ∧ t_path 2 3 = 4
```

Assumptions: Unconditional closed theorem settling t_walk(2,3) = t_path(2,3) = 4 inside the pure Lean 4 kernel. Applicability: Unconditional (closed statement; no outer premise witness needed).

TournamentEdgeColor.existsAvoidingPath_iff_sat_tourneyPathCNF (TournamentEdgeColor/SATCert.lean:1159-1162) src:dc75336d973d91

```
theorem existsAvoidingPath_iff_sat_tourneyPathCNF {n k l : ℕ} (hk : 1 ≤ k) :
    ExistsAvoidingPath n k l ↔ ∃ σ : Eval TourneyVar, Satisfies σ (tourneyPathCNF n k
      ↪ l)
```

Assumptions: Any natural numbers n, l and k >= 1 (matching l_paper >= 1). Both directions proved unconditionally in Lean. 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
`ColorReductionHolds`	`nat_sub`	`- 1`	scope_limitation	ColorReductionHolds is only invoked for l >= 2 (where l - 1 >= 1); degenerate l = 1 is settled separately by exact_k_1.
`cr_implies_conj_1_7`	`nat_sub`	`- 1`	intended_convention	Explicit hypothesis hl : 2 <= l guarantees l - 1 >= 1 with zero Nat underflow.
`cr_implies_conj_1_7`	`real_log_rpow`	`^ (l - 1)`	intended_convention	Lexical match on caret operator in Nat exponentiation k ^ (l - 1); no real-valued rpow or log used.
`lexGrid_lower_bound`	`nat_sub`	`- 1`	intended_convention	Explicit hypothesis hl : 1 <= l ensures exact non-underflowing exponent l - 1.

[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: Exact threshold t_walk(2, 4) = t_path(2, 4) = 8	Lean + Cert	`TournamentEdgeColor.t_2_4_exact_of_tt4_rup` + run.sh (dcbbabd7aa88) [regenerable; replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib proves [2]^3 lower bound, CNF equisatisfiability, TT_4 reduction, and RUP soundness; external 194.9 MB LRAT UNSAT proof checked by drat-trim (0 RAT) and lrat-check.
Thm. 2: Exact threshold t_walk(3, 3) = t_path(3, 3) = 9	Lean + Cert	`TournamentEdgeColor.t_3_3_exact_of_cube33_rup` + run.sh (dcbbabd7aa88) [regenerable; replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib proves [3]^2 lower bound, CNF equisatisfiability, 3-cube split, and RUP soundness; three external LRAT UNSAT proofs (91.5, 64.9, 292.7 MB) checked by drat-trim (0 RAT) and lrat-check.
Thm. 3: General lexicographic grid lower bound t_path(l, k) >= t_walk(l, k) >= l^(k-1) for all l, k >= 1	Lean Kernel	`TournamentEdgeColor.lexGrid_lower_bound`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (standard axioms only) + Box 1 definitions for all l >= 1, k >= 1.
Thm. 4: d-colouring avoiding l-flashes and (d+1)-rainbows (walks) if and only if grid embedding in [l]^d (for all l, d >= 1)	Lean Kernel	`TournamentEdgeColor.bounded_palette_iff_gridEmbedding`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (standard axioms only) + Box 1 definitions for all l >= 1, d >= 1.
Cor. 5: Colour Reduction at (l, k) implies t_walk(l, k) = l^(k-1) (for all l >= 1, k >= 2)	Lean Kernel	`TournamentEdgeColor.cr_implies_conj_1_7`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel; conditional on ColorReduction Holds at (l, k).
Prop. 6: Exhaustive walk-good = grid-embeddable classification for (2,3), (2,4), (3,3) (43 and 107 extremal classes)	Cert Only	recompute.py (0c815dd65d4a) [regenerable; replay pending]	External cert only; fidelity: n/a	Deterministic grid enumeration in recompute.py (512 orientations on [2]^3 and [3]^2) matched against three independent heredity-growth SAT pipelines (106 top proofs also checked by cake_lpr).
Prop. 8: Exact k = 2 law t_walk(l, 2) = t_path(l, 2) = l for all l >= 1	Lean Kernel	`TournamentEdgeColor.exact_k_2`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel via Redei Hamiltonian path theorem.
Prop. 9: Exact (2, 3) law t_walk(2, 3) = t_path(2, 3) = 4	Lean Kernel	`TournamentEdgeColor.exact_2_3`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (5-vertex Hamiltonian-path contradiction).
Prop. 10: Exact two-way equisatisfiability of palette-free tourneyPath CNF(n, l, k) for l >= 1	Lean Kernel	`TournamentEdgeColor.existsAvoidingPath_iff_sat_tourneyPathCNF`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 + Mathlib pure kernel (both directions proved unconditionally).
Conj. 7: Colour-Reduction Conjecture (every (l,k)-walk-good tournament admits a (k-1)-colouring)	Unsettled	Open conjecture	Lean: none; fidelity: n/a	Explicitly unsettled conjecture beyond (2,3), (2,4), (3,3) (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: b954d8eecf16 | Lean mode: committed_receipt | Closure SHA-256: 7a9aa54d3792 | Git anchor: 2f09f46076a2 | Axioms SHA-256: 943afe069ee1 | Fidelity SHA-256: 5a8514712e30.

6. PRIOR WORK AND OPEN QUESTIONS

Girão, Illingworth, Michel, Savery, and Scott [1] formulated Conjectures 1.7 and 5.3, proved Theorem 1.8 ($t(\ell,k) \le (1 + 2/\sqrt{\ell})^k \ell^{k-1}$) and Theorem 1.10, and noted that $t(\ell,k) = \ell^{k-1}$ was wide open for all $\ell \ge 2$ and $k \ge 3$ (a status unchanged in Chapter 6, Section 6.4 of Michel's 2026 Oxford DPhil thesis [3], Conjectures 6.6 and 6.23). Their problem generalizes the transitive-tournament threshold $f(\ell,k)$ of Lefmann, Rödl, and Thomas [2] (`[to check]`), who proved $f(\ell,k) = \ell^{k-1}$ for $\ell \le 2$ or $k \le 4$ (as reported in [1]). Public 2026 GitHub notes (`graph-theory-AI/Graph-Theory-LLM-Proofs attacks/2305.13422__00` and `the-omega-institute/trureturing` issue `#8466`) recorded only the smaller case $t(2,3) = 4$ (re-proved unconditionally in our Proposition 9) and $t(\ell,2) = \ell$ (Proposition 8).

Two independent prior-work search logs passed `litsearch.py check` (exit 0; 38 + 13 = 51 structured queries across arXiv [24 rows], OEIS, zbMATH, Stack Exchange, MathOverflow, GitHub, DataCite, Zenodo, and HAL, plus a follow-up keyed CORE search of 7 requests returning 70 records). Exact values $t(2,4) = 8$ and $t(3,3) = 9$ for arbitrary tournaments were not found in the searches logged. *Not searched or unread* (`[to check]`): Google Scholar (monthly quota reached during the logged run), OpenAlex and Semantic Scholar citing-works lists beyond one saved reply (HTTP 429 rate-limited), Springer's typeset *Combinatorica* printing and its publisher erratum (`Zbl 1574.05208`, reported by zbMATH to affect Corollary 2.5 only), full texts of [2, 4, 5, 6, 7, 8], MathSciNet, books in print, theses other than [3], and private or unindexed repositories.

Open questions. Four concrete questions remain unsettled:

1. *Next thresholds $t(4,3)$ and $t(2,5)$:* Both are conjectured to equal 16; Theorem 1.8 of [1] currently leaves $16 \le t(4,3) \le 128$ and $16 \le t(2,5) \le 1{,}312$. The $[2]^4$ grid already has 2^{55} orientations of its free pairs, placing direct exhaustive growth beyond our current computational budget.
2. *Colour-Reduction Conjecture* (*Conjecture 7*): Exhaustive testing confirms Conjecture 7 for all tournaments on $n \le 8$ vertices at $(4,3)$ (17 non-grid classes on 8 vertices, all bad) and $(2,5)$ (all 6,880 classes embed in $[2]^4$), plus 31,401 sampled tournaments on 9–12 vertices and circulant/Paley tournaments up to 15 vertices (0 counterexamples; 13 circulant instances undecided at a 30 s solver cap). Proving or refuting Conjecture 7 at any pair beyond $(3,3)$ would either resolve or sharpen Conjecture 1.7.
3. *Human-readable proofs:* Does a short conceptual proof exist for $t(2,4) = 8$ or $t(3,3) = 9$ without SAT certificates?
4. *Formally verified LRAT replay on the single-formula proofs:* Running `cake_lpr` [8] (`[to check]`) on the four single-formula LRAT proofs in Table 1 would upgrade their external checker from `lrat-check` to a formally verified CakeML checker.

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

All manuscript sources, figures, ledgers, and verification scripts reside in `papers/flashes-and-rainbows-in-tournaments/draft1/`, the Lean 4 + Mathlib formalization resides in `lean/TournamentEdgeColor/`, and the committed SAT/LRAT verification logs reside in `docs/shadow/round19/R19-3/lrat-check/`. Running `python3 papers/flashes-and-rainbows-in-tournaments/draft1/recompute.py` reproduces in under 1 s all lexicographic grid witnesses, height-map inversions, exhaustive $n \le 5$ checks, exhaustive $[2]^2$, $[2]^3$, and $[3]^2$ grid-class counts, Theorem 1.8 bounds, and committed SAT/LRAT log SHA-256 checks.

AI research agents built and directed by Tyler B. Hughes generated and checked the computational searches, SAT certificates, Lean 4 formalization, figures, and manuscript text. No human mathematician has yet refereed the proof.

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

REFERENCES

(1) Girão, A.; Illingworth, F.; Michel, L.; Savery, M.; Scott, A. Flashes and Rainbows in Tournaments. *Combinatorica* **2024**, *44*, 675–690. DOI: 10.1007/s00493-024-00090-7. arXiv:2305.13422v2.

(2) Lefmann, H.; Rödl, V.; Thomas, R. Monochromatic vs multicolored paths. *Graphs and Combinatorics* **1992**, *8*, 323–332. DOI: 10.1007/BF02351589. [to check]

(3) Michel, L. *Putting the pieces together: from graph decompositions to Ramsey theory*. DPhil Thesis, University of Oxford, **2026**. DOI: 10.5287/ora-yqryd5jv2.

(4) Rédei, L. Ein kombinatorischer Satz. *Acta Litt. Sci. Szeged* **1934**, *7*, 39–43. Zbl 0009.14606. [to check]

(5) Stearns, R. The Voting Problem. *Amer. Math. Monthly* **1959**, *66*, 761–763. DOI: 10.1080/00029890.1959.11989405. [to check]

(6) Wetzler, N.; Heule, M. J. H.; Hunt, W. A. DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs. *Lecture Notes in Computer Science* **2014**, *8561*, 422–429. DOI: 10.1007/978-3-319-09284-3_31. [to check]

(7) Cruz-Filipe, L.; Heule, M. J. H.; Hunt, W. A.; Kaufmann, M.; Schneider-Kamp, P. Efficient Certified RAT Verification. *Lecture Notes in Computer Science* **2017**, *10395*, 220–236. DOI: 10.1007/978-3-319-63046-5_14. [to check]

(8) Tan, Y. K.; Heule, M. J. H.; Myreen, M. O. `cake_lpr`: Verified Propagation Redundancy Checking in CakeML. *Lecture Notes in Computer Science* **2021**, *12652*, 223–241. DOI: 10.1007/978-3-030-72013-1_12. [to check]