

An exact computer-assisted certificate for

$$R(5, 6) \leq 84$$

Article and explanatory supplements

András Pongrácz

University of Debrecen, Institute of Mathematics

30 October 2026

Status and accompanying verification files

The complete verification archive is available at <https://github.com/apongracz86/Ramsey56/releases/tag/v1.0-review>. The three parts reproduce the article, the worked explanation, and the detailed mathematical specification. The archive `verification.tar.xz` preserves the verification files and their directory structure. Its SHA-256 digest is

88ded8db2254dec75be80e476d684334a685654e3c5a4108cf9c10422bdd3d67

File names cited in the three parts refer to that distribution. Its README and the specification give the reproduction recipes. The article states the external mathematical inputs. The specification explains the calculations performed by each verification program.

Contents

I	The article	5
I.1	Graphs, counts, and the statement	5
I.2	Degrees and the first counting identity	6
I.3	What the small graph catalogues supply	7
I.4	From possible neighborhoods to a counting system	8
I.4.1	Profiles and containing polygons	8
I.4.2	What the profile variables mean	9
I.4.3	Recovering counts and imposing equations	9
I.5	How a certificate proves a bound	11
I.6	Obtaining all required local profile bounds	12
I.6.1	The four edge bounds	13
I.6.2	Five inequalities retaining triangles and adjacent edges	14
I.7	A distribution on seven-vertex graphs	14
I.7.1	How a local fact becomes a coefficient row	15
I.7.2	The local relations used in these rows	16
I.8	Why sums of squares give further inequalities	16
I.9	The final rational sum and its contradiction	17
I.10	What an independent reader should check	18
I.10.1	Reading the mathematics	18
I.10.2	Replaying the supplied calculations	19
II	The method and a worked local example	22
II.1	Notation and the information retained from a graph	23
II.2	What the worked example establishes	23
II.3	The first triangle double count	24
II.4	The worked example: its claim and local inputs	25
II.5	A second identity, counted around triangles	26
II.6	The complete weighted counting argument	28
II.7	Checking the ten local inequalities	29
II.8	Where the profile restrictions come from	30
II.9	Profile variables and the saved equation multipliers	31
II.10	The multipliers and their normalization	34
II.11	What the remaining local computations add	35
II.12	Why the global proof uses seven-vertex graphs	37
II.13	The positive-semidefinite inequalities are sums of squares	38
II.14	The final exact separation	39
II.15	A practical verification route	40
II.16	Mathematical review and finite verification	42
III	Detailed mathematical specification	43
III.1	Notation and external inputs	44
III.1.1	Complement profiles	45
III.2	Proof structure and verification	45
III.2.1	A route through the mathematics and the programs	46
III.2.2	Locating the files	46
III.3	The first counting reduction	47
III.4	Variables of the local profile model	47

III.4.1	Actual profile sets and their outer polygons	48
III.4.2	The finite coordinate sets	48
III.4.3	From a graph to nonnegative weights	49
III.4.4	A catalogue example of fractional weights	50
III.4.5	Ambient statistics as linear functionals	50
III.5	Four-vertex variables and bulk equations	50
III.6	Triple types and rooted incidence equations	51
III.7	Joint degrees, disjoint edges, and paths	52
III.8	Checking a bound by a weighted sum	53
III.8.1	The principle before matrix notation	53
III.8.2	How the equations become a matrix	54
III.8.3	What signs must be checked	54
III.8.4	Matching the displayed equations to the stored rows	54
III.8.5	Files, exact arithmetic, and the check to perform	55
III.9	Recursive profile coverage and exact polygons	55
III.10	Endpoints, integrality, and the five local facets	56
III.11	Seven-vertex probabilities and completeness	59
III.12	Exact finite sampling	59
III.13	Root conventions and canonical masks	60
III.14	Base degree and common-neighborhood rows	61
III.14.1	Degree-square equalities	61
III.15	Mixed common-neighborhood cells	62
III.16	Cell integrality and binary degree products	62
III.16.1	Products of degrees	63
III.17	Edge and profile inequalities inside cells	63
III.17.1	Full cell profiles	64
III.17.2	Small-order outer sets	64
III.18	Restricted edge cores and critical order 17	65
III.18.1	A cubic inequality from the seven critical graphs	65
III.19	Lifting the four local edge bounds and five facets	66
III.20	The exact finite Gram identity	66
III.21	Integer blocks, square vectors, and overflow checks	68
III.22	Row normalization and exact source linkage	68
III.23	The separating sum and its strict margin	69
III.24	Which rows have nonzero final weights	70
III.25	Catalogue extraction and its precise trust boundary	71
III.26	Environment and safe execution	72
III.27	Execution recipes and expected output	72
III.28	The 18-job suite: local and catalogue checks	74
III.28.1	Required local dual replay	74
III.28.2	Recursive polygons	74
III.28.3	Dependency closure	74
III.28.4	Small catalogue records	75
III.28.5	The (4, 5) catalogues	75
III.29	The 18-job suite: global coefficients	75
III.29.1	Seven-vertex completeness	75
III.29.2	Base rows	75
III.29.3	Mixed cells and general profiles	75
III.29.4	Integrality and binary degrees	75
III.29.5	Critical cores	75
III.29.6	Cell edge supports	75

III.29.7	Neighborhood edges	76
III.29.8	Neighborhood profile features	76
III.29.9	Restricted edge cores	76
III.30	The 18-job suite: Gram matrices and final sum	76
III.30.1	Integer Gram reconstruction	76
III.30.2	Direct separation replay	76
III.30.3	Arbitrary-precision replay	76
III.30.4	A further global auditor	76
III.31	Writing a verifier from this specification	77
III.32	Small arithmetic checks and common errors	77
III.33	What a successful reproduction establishes	78

Part I

An exact computer-assisted certificate for $R(5, 6) \leq 84$

András Pongrácz

University of Debrecen, Institute of Mathematics

30 October 2026

Abstract

We give an exact computer-assisted proof of $R(5, 6) \leq 84$. We derive a system of linear inequalities that must be satisfied by any graph on 84 vertices containing neither a five-vertex clique nor a six-vertex independent set. We provide explicit nonnegative rational weights for which adding these inequalities yields a contradiction. Published small Ramsey bounds and complete catalogues of smaller Ramsey graphs supply the external inputs. The accompanying verification programs check the derivation of the inequalities and their weighted sum using exact arithmetic. Two supplements give a worked local calculation and a detailed mathematical specification of the complete proof.

I.1 Graphs, counts, and the statement

All graphs in this paper are finite, undirected, and simple: there are no loops or multiple edges. For a graph X , write $V(X)$ for its vertex set, $E(X)$ for its edge set, and $n = |V(X)|$ for its order. For a vertex $v \in V(X)$, its neighborhood $N_X(v)$ is the set of vertices adjacent to v , and $d_v = d_X(v) = |N_X(v)|$ is its degree. For $S \subseteq V(X)$, $X[S]$ denotes the induced subgraph: its vertices are S and its edges are all the edges of X with both ends in S . The complement $\bar{X}$ has the same vertices and exactly the pairs which are not edges of X .

A clique is a vertex set whose distinct vertices are pairwise adjacent; an independent set is a vertex set containing no edge. Define

$$\mathcal{R}(s, r, n) = \{X : |V(X)| = n, X \text{ has no clique of size } s \text{ and no independent set of size } r\}.$$

We also call a member of this family an (s, r) -graph of order n . The Ramsey number $R(s, r)$ is the least n for which $\mathcal{R}(s, r, n)$ is empty.

We use the following statistics of a graph:

- $e(X) = |E(X)|$, the number of edges;
- $t(X)$, the number of unordered three-element vertex sets inducing a triangle;
- $w(X) = \sum_{v \in V(X)} \binom{d_v}{2}$, the number of choices of a vertex and an unordered pair of its neighbors;
- $q(X) = w(X) - t(X)$.

The pair of neighbors counted by $w(X)$ may itself be adjacent. Thus a triangle contributes three to $w(X)$, while an induced three-vertex path contributes one. Consequently

$$q(X) = 2t(X) + \#\{\text{three-element sets inducing a path in } X\}.$$

For a triangle, $(e, t, q) = (3, 1, 2)$; for a three-vertex path, $(e, t, q) = (2, 0, 1)$. All these counts are integers. When the graph is specified, we abbreviate them to e, t, w, q .

For every vertex v of the graph under discussion, put

$$A_v = X[N_X(v)], \quad B_v = X[V(X) \setminus (N_X(v) \cup \{v\})].$$

These are the induced subgraphs on the neighbors and on the nonneighbors of v , respectively; neither contains v . Their orders are d_v and $n - 1 - d_v$. The same definitions apply when X is a smaller graph considered in an intermediate calculation.

Theorem. Subject to the small Ramsey bounds and complete catalogue inputs stated in Section I.3, $\mathcal{R}(5, 6, 84)$ is empty. Therefore $R(5, 6) \leq 84$.

The small Ramsey bounds and catalogue completeness are established in the cited sources. The verification programs check the inequalities derived from these inputs and the final contradiction by exact arithmetic.

The route through the proof

Suppose that a counterexample $X \in \mathcal{R}(5, 6, 84)$ exists. We first determine its possible vertex degrees. We then obtain restrictions on the counts e, t, q in each A_v and B_v . The word *local* refers to these neighborhoods or to smaller specified subsets of $V(X)$.

We write equations that every possible graph X and each of its neighborhoods would have to satisfy. We allow additional solutions that may not come from graphs. This larger system is called a *relaxation*. An inconsistency in the larger system rules out the existence of graphs as well.

The final system describes the proportions of the different graphs induced by seven-element subsets of $V(X)$. Its contradiction is a weighted sum of valid inequalities. The weights are supplied in the electronic supplement. Understanding why an inequality is valid and checking the arithmetic of its saved weights are separate tasks; the reader's instructions in Section I.10 explain both.

I.2 Degrees and the first counting identity

Let $X \in \mathcal{R}(5, 6, 84)$. For a $v \in V(X)$, a four-clique in A_v , together with v , would be a five-clique in X . An independent six-set in A_v would already be one in X . Hence $A_v \in \mathcal{R}(4, 6, d_v)$. Similarly, an independent five-set in B_v , together with v , would be an independent six-set in X , so $B_v \in \mathcal{R}(5, 5, 83 - d_v)$.

The bounds $R(4, 6) \leq 40$ and $R(5, 5) \leq 46$ (see [I.1, Table Ib, p. 7] and [I.2, Theorem 1.1]) now give $d_v \leq 39$ and $83 - d_v \leq 45$. Thus every vertex has degree 38 or 39. The four possible local families are precisely

d_v	A_v	B_v
38	$\mathcal{R}(4, 6, 38)$	$\mathcal{R}(5, 5, 45)$
39	$\mathcal{R}(4, 6, 39)$	$\mathcal{R}(5, 5, 44)$.

If x vertices have degree 39, the sum of the degrees is $38(84 - x) + 39x = 3192 + x = 2e(X)$. Therefore x is even and $e(X) = 1596 + x/2$.

The first relation between the two kinds of neighborhoods is

$$\sum_{v \in V(X)} [e(B_v) - e(A_v) - d_v(42 - d_v)] = 0. \quad (\text{I.1})$$

We give the double count, since it is also the first example of the identities used in the proof. In an arbitrary graph of order n , each triangle contributes one edge to each of three neighborhoods. Thus $\sum_v e(A_v) = 3t(X)$. For an edge uw , write $c_{uw} = |N_X(u) \cap N_X(w)|$. The number of vertices adjacent to neither u nor w is $n - d_u - d_w + c_{uw}$. Counting an edge inside a nonneighborhood in this way gives

$$\begin{aligned} \sum_v e(B_v) &= \sum_{uw \in E(X)} (n - d_u - d_w + c_{uw}) \\ &= ne(X) - \sum_v d_v^2 + 3t(X). \end{aligned}$$

Here d_v occurs once for each of the d_v edges at v , and every triangle contributes one common neighbor to each of its three edges. Subtract $\sum_v e(A_v)$ and use $e(X) = \frac{1}{2} \sum_v d_v$. The result is

$$\sum_v [e(B_v) - e(A_v) - d_v(n/2 - d_v)] = 0.$$

Putting $n = 84$ proves (I.1). This is the simplified form of [I.3, Theorem 2.2, $m = 2$], also stated in [I.2, equation (1.2)]. The last term in (I.1) is 152 for degree 38 and 117 for degree 39.

To use this identity we need information about the possible edge and triangle counts inside A_v and B_v . The next sections explain where that information comes from, before introducing its numerical bounds.

I.3 What the small graph catalogues supply

A *complete catalogue* for $\mathcal{R}(s, r, m)$ lists one representative of every graph in that family up to isomorphism. Two graphs are isomorphic if a bijection of their vertex sets preserves adjacency. Counts such as e, t, q are unchanged by such a bijection, so one representative suffices for checking an inequality involving these counts. The catalogues used here are supplied on Brendan McKay's Ramsey-graph page [I.4]. They cover the following families:

Family	Orders or edge classes used
$\mathcal{R}(3, 5, m)$	$1 \leq m \leq 13$
$\mathcal{R}(3, 6, m)$	$13 \leq m \leq 17$
$\mathcal{R}(4, 4, m)$	$5 \leq m \leq 17$
$\mathcal{R}(4, 5, 24)$	the complete order-24 family
$\mathcal{R}(4, 5, m), 19 \leq m \leq 23$	26 specified extreme-edge classes

An *edge class* fixes both the order m and the exact edge count. The last line supplies complete catalogues only for those 26 classes, not for every $(4, 5)$ -graph on 19 through 23 vertices. The files from `r45extreme.tar.gz` identify each class by its order and edge count; the packaged distribution contains the extracted graph6 files. The other required edge classes are handled by proved outer bounds, described in Section I.6.

For example, `r36_13.g6.gz` encodes the complete 13-vertex $(3, 6)$ catalogue in compressed graph6 format. Graph6 encodes graph adjacency. The file `r36_13.g6.gz.profiles` records the distinct tuples $(m, e(Y), t(Y), q(Y))$ extracted from those graphs. Several nonisomorphic graphs may have the same tuple. Once the profile has been computed, a linear inequality in these statistics can be checked on one copy of each distinct profile.

The published small Ramsey bounds used with these data are

$$R(3, 5) = 14, \quad R(3, 6) = 18, \quad R(4, 4) = 18, \quad R(4, 5) = 25,$$

and the two bounds used in Section I.2. The $R(4, 6) \leq 40$ result is recorded in the survey [I.1], which attributes the computations to Angeltveit and McKay through personal communications from 2020 and 2024. A separate full publication of that computation is not supplied here. The bound $R(5, 5) \leq 46$ is [I.2, Theorem 1.1].

We also use the edge-count intervals recorded in [I.2, Appendix A, Table 1]:

m	19	20	21	22	23	24
$e_{\min}$	57	68	77	88	101	116
$e_{\max}$	92	100	107	114	122	132.

These are bounds on every graph in the corresponding $\mathcal{R}(4, 5, m)$ family. Every integer edge count between the stated endpoints must be covered in the proof. Absence of a catalogue file is not evidence that an edge class is empty.

For the smaller common-neighborhood cells used later, we also need the established minimum-edge formula for $(3, 6)$ -graphs on at most 12 vertices, a specialization of [I.6, Theorem 2]. If Y is such a graph of order c , the lower bounds are $e(Y) \geq 0$ for $0 \leq c \leq 5$, $e(Y) \geq c - 5$ for $6 \leq c \leq 10$, and $e(Y) \geq 3c - 25$ for $11 \leq c \leq 12$. The specification states the additional elementary upper bounds and explains how these give outer profiles when the catalogue range $13 \leq c \leq 17$ does not apply.

The catalogue files are inputs. Their mathematical completeness is accepted from the cited work. Our checks count the supplied records and check their derived profiles. The optional full verification also recomputes the profiles of every supplied graph. Neither operation reproves that the catalogue contains every possible graph.

I.4 From possible neighborhoods to a counting system

This section applies to an arbitrary ambient graph $X \in \mathcal{R}(s, r, n)$. It will be used, separately, for the four local families in Section I.2. The notation A_v, B_v, d_v continues to refer to the neighborhoods and degrees in the currently specified graph X .

The construction has three steps. We retain a small number of statistics of each possible neighborhood. We introduce nonnegative variables to record how these possibilities occur around the vertices of X . We then impose equations that count the same configurations in two ways.

I.4.1 Profiles and containing polygons

For fixed parameters s', r', m and an integer edge count η , define

$$\mathcal{S}_{s', r', m, \eta} = \{(t(Y), q(Y)) : Y \in \mathcal{R}(s', r', m), e(Y) = \eta\}.$$

This is a finite set of points in the plane. The order m and edge count η are fixed, and the two coordinates record triangles and q . When a complete catalogue for this edge class is available, it determines $\mathcal{S}$ exactly.

For each needed edge class we use a bounded rational convex polygon $\mathcal{P}_{s', r', m, \eta}$ containing $\mathcal{S}_{s', r', m, \eta}$. Here a polygon may be a segment or a single point. A polygon is allowed to be empty only when the corresponding graph class has independently been ruled out. There are two ways to obtain it:

- From a complete catalogue, take the convex hull of its points: all their weighted averages with nonnegative weights summing to one.
- Without a complete catalogue, prove finitely many inequalities $L \leq \alpha t(Y) + \beta q(Y) \leq U$ for every graph in the edge class, and intersect the corresponding half-planes.

A point in either polygon need not be the profile of a graph: even averaging two catalogue profiles can produce a point with no realizing graph. If a linear inequality holds throughout the polygon, it holds for every actual profile inside it. Moreover, to check a linear inequality on a bounded convex polygon, it suffices to check its vertices: every point is a convex combination of the vertices, and a linear expression preserves that combination.

Write $\mathcal{V}_{s', r', m, \eta}$ for the finite vertex set of $\mathcal{P}_{s', r', m, \eta}$. Denote its coordinates by (τ, κ) . These rational coordinates may lie between integer graph profiles. Containment of all actual profiles is the property needed below.

I.4.2 What the profile variables mean

Let $D \subseteq \{0, \dots, n-1\}$ contain every possible degree of a graph in the ambient family. Define the two finite index sets

$$\begin{aligned}\mathcal{I}_A &= \{(d, \eta, \tau, \kappa) : d \in D, \quad \eta \in \mathbb{Z}, \quad 0 \leq \eta \leq \binom{d}{2}, \\ &\quad (\tau, \kappa) \in \mathcal{V}_{s-1, r, d, \eta}\}, \\ \mathcal{I}_B &= \{(h, \eta, \tau, \kappa) : h = n-1-d \text{ for some } d \in D, \\ &\quad \eta \in \mathbb{Z}, \quad 0 \leq \eta \leq \binom{h}{2}, \quad (\tau, \kappa) \in \mathcal{V}_{s, r-1, h, \eta}\}.\end{aligned}$$

The four entries specify the order of a local graph, its exact edge count, and one vertex of its profile polygon. Introduce a nonnegative real variable a_i for every $i \in \mathcal{I}_A$ and b_j for every $j \in \mathcal{I}_B$. These are variables of our counting system. Here is how an actual graph assigns values to them.

For each $v \in V(X)$, the pair $(t(A_v), q(A_v))$ lies in the polygon with order d_v and edge count $e(A_v)$. Express that pair as a convex combination of the polygon's vertices. Let $\lambda_{v,i}^A$ be the coefficient assigned to index i , with coefficient zero for all indices having any other order or edge count. Then

$$\sum_{i \in \mathcal{I}_A} \lambda_{v,i}^A = 1, \quad \sum_{i=(d,\eta,\tau,\kappa) \in \mathcal{I}_A} \lambda_{v,i}^A (t(A_v), q(A_v)) = (t(A_v), q(A_v)).$$

Choose coefficients $\lambda_{v,j}^B$ in the same way for B_v , and set

$$a_i = \sum_{v \in V(X)} \lambda_{v,i}^A, \quad b_j = \sum_{v \in V(X)} \lambda_{v,j}^B.$$

Thus a_i is an aggregate weight assigned to one allowed neighborhood profile vertex; b_j has the corresponding meaning for nonneighborhoods. They need not be integers, even when they arise from an actual graph.

For a concrete example, the catalogue profiles with order 13 and 16 edges in `r36_13.gz.profiles` have exactly the pairs $(0, 25)$, $(0, 26)$, $(0, 28)$. The polygon is the segment with endpoints $(0, 25)$ and $(0, 28)$, and

$$(0, 26) = \frac{2}{3}(0, 25) + \frac{1}{3}(0, 28).$$

A single vertex whose neighborhood has profile $(13, 16, 0, 26)$ therefore contributes $2/3$ to $a_{13,16,0,25}$ and $1/3$ to $a_{13,16,0,28}$. Its total weight is one, and its weighted q coordinate is 26. This shows how fractional profile weights represent one vertex while preserving the statistics of its neighborhood.

I.4.3 Recovering counts and imposing equations

Let m_d be the sum of all a variables with first coordinate d . For an assignment obtained from a graph, m_d is the number of vertices of degree d . The sum of the b variables with first coordinate $n-1-d$ has the same value. The system therefore includes these matching equations and $\sum_{d \in D} m_d = n$.

Define four linear expressions in the a variables:

$$\begin{aligned}\mathcal{E}(a) &= \frac{1}{2} \sum_{\mathcal{I}_A} d a_{d,\eta,\tau,\kappa}, & \mathcal{T}(a) &= \frac{1}{3} \sum_{\mathcal{I}_A} \eta a_{d,\eta,\tau,\kappa}, \\ \mathcal{W}(a) &= \sum_{\mathcal{I}_A} \binom{d}{2} a_{d,\eta,\tau,\kappa}, & \mathcal{Q}(a) &= \mathcal{W}(a) - \mathcal{T}(a).\end{aligned}$$

Each sum runs over the entire index set $\mathcal{I}_A$. On the assignment constructed from X , these expressions equal $e(X)$, $t(X)$, $w(X)$, $q(X)$. The first assertion is handshaking; the second uses $\sum_v e(A_v) = 3t(X)$; the third counts pairs of neighbors at their center. More generally, any

expression affine in the local coordinates (τ, κ) is preserved by the convex combinations at each vertex. The order and edge count remain fixed within each combination.

We next introduce $z_F \geq 0$, one variable for each of the eleven isomorphism types F of graphs on four vertices. Its value for an actual X is the number of four-element subsets inducing type F . Examples are z_{K_4} for four-cliques and z_{4K_1} for independent four-sets. Counting four-sets and the edges, triangles and adjacent-edge pairs inside them gives

$$\begin{aligned} \sum_F z_F &= \binom{n}{4}, & \sum_F e(F) z_F &= \binom{n-2}{2} \mathcal{E}(a), \\ \sum_F t(F) z_F &= (n-3) \mathcal{T}(a), & \sum_F w(F) z_F &= (n-3) \mathcal{W}(a). \end{aligned}$$

For example, an edge of X is contained in exactly $\binom{n-2}{2}$ four-sets. Counting incidences between an edge and a containing four-set proves the second equation.

We also match triples counted inside neighborhoods with the same triples counted inside four-sets. To make this explicit, define

$$\begin{aligned} T_0(m, \eta, \tau, \kappa) &= \binom{m}{3} - (m-2)\eta + \kappa, \\ T_1(m, \eta, \tau, \kappa) &= (m-2)\eta - 2\kappa + \tau, \\ T_2(m, \eta, \tau, \kappa) &= \kappa - 2\tau, & T_3(m, \eta, \tau, \kappa) &= \tau. \end{aligned}$$

For a graph profile these are the counts of induced triples with 0, 1, 2, 3 edges. They follow by counting all triples, edge incidences in triples, and adjacent-edge pairs in triples. Let $U_k(F)$ count vertices of F adjacent to all other three vertices whose deletion leaves exactly k edges. Let $Z_k(F)$ count isolated vertices of F whose deletion leaves k edges. Then, for $k = 0, 1, 2, 3$,

$$\begin{aligned} \sum_{\mathcal{I}_A} T_k(d, \eta, \tau, \kappa) a_{d, \eta, \tau, \kappa} &= \sum_F U_k(F) z_F, \\ \sum_{\mathcal{I}_B} T_k(h, \eta, \tau, \kappa) b_{h, \eta, \tau, \kappa} &= \sum_F Z_k(F) z_F. \end{aligned}$$

For instance, a triple in A_v together with v is a four-set in which v is adjacent to the whole triple. Both sides count precisely those choices. These equations require the neighborhood and nonneighborhood statistics to fit into the *same* collection of four-sets.

The simplified triple identity is another equation of the system:

$$\sum_{\mathcal{I}_B} \eta b_{h, \eta, \tau, \kappa} - \sum_{\mathcal{I}_A} (\eta + d(n/2 - d)) a_{d, \eta, \tau, \kappa} = 0.$$

We also record edges by their endpoint degrees. For $d, f \in D$ with $d \leq f$, let $j_{d,f} \geq 0$ count such edges. Each edge within degree class d contributes two ends to that class; each edge to another class contributes one. Thus

$$dm_d = 2j_{d,d} + \sum_{f < d} j_{f,d} + \sum_{f > d} j_{d,f}.$$

The sums here are over $f \in D$. If $p(F)$ denotes the number of unoriented paths of three edges on four distinct vertices of F , with additional edges allowed, the path equation is

$$\sum_F p(F) z_F = \sum_{d \leq f} (d-1)(f-1) j_{d,f} - 3\mathcal{T}(a).$$

To see the correction term, choose the middle edge of a path and then its two outer endpoints. Coincident outer endpoints correspond to triangles; there are $3t(X)$ of these choices.

To restrict the model to a fixed edge count $e(X) = e_0$, we impose the linear equation $\mathcal{E}(a) = e_0$. Only in this fixed-edge model do we also include the equation counting pairs of disjoint edges:

$$\mathcal{W}(a) + \sum_F \left(\binom{e(F)}{2} - w(F) \right) z_F = \binom{e_0}{2}.$$

The first term counts pairs sharing a vertex. Each remaining pair has four distinct endpoints and is counted by exactly one four-set. Fixing e_0 makes the right side a constant, so this is a linear equation.

These equations are necessary conditions on every graph in the chosen family. We allow the variables to be arbitrary nonnegative real numbers satisfying them. We do not claim that every solution comes from a graph. Bounds valid for every solution therefore apply to every graph in the family. The accompanying specification gives the full row list, its order in the files, and the distinction between versions 3 and 4.

I.5 How a certificate proves a bound

A certificate in this paper is a finite list of rational multipliers for already justified equations or inequalities. Checking a certificate means checking what their weighted sum says. The optimization algorithm which found the multipliers is not part of that check.

Consider this elementary example, unrelated to the Ramsey data. Suppose $u_1, u_2, u_3 \geq 0$ and

$$u_1 + u_2 = 5, \quad u_2 + u_3 = 4.$$

Twice the first equation plus the second gives $2u_1 + 3u_2 + u_3 = 14$. Therefore

$$3u_1 + 3u_2 + u_3 = 14 + u_1 \geq 14.$$

The multipliers $(2, 1)$ and the nonnegative remainder u_1 are a complete proof of this lower bound for every feasible triple (u_1, u_2, u_3) .

The actual certificates do the same thing with many more variables. First remove any variable already known to be zero. For example, $z_{K_4} = 0$ in a $(4, 6)$ -graph because it contains no four-clique. Similarly $z_{4K_1} = 0$ when the forbidden independent-set size is four. Removing these coordinates means substituting their known value zero in every equation.

Let $\mathbf{u} = (u_1, \dots, u_N)^\top$ be the remaining profile, four-set, and endpoint-degree variables, in the order used by the certificate. Let there be m counting equations. Put their coefficients in the m by N rational matrix M , and their right-hand sides in $\mathbf{b} \in \mathbb{Q}^m$. Thus the k th equation is $\sum_{j=1}^N M_{kj} u_j = b_k$, and the whole system is

$$M\mathbf{u} = \mathbf{b}, \quad \mathbf{u} \geq 0.$$

A coefficient vector $\mathbf{c} \in \mathbb{Q}^N$ specifies the quantity we wish to bound: $\mathbf{c}^\top \mathbf{u} = \sum_j c_j u_j$. For example, $\mathbf{c}$ can be the coefficients of $\mathcal{E}(a)$ for a lower edge bound, or of $-\mathcal{E}(a)$ for an upper edge bound.

Choose a rational multiplier y_k for the k th equation, and collect these into $\mathbf{y} \in \mathbb{Q}^m$. Their weighted sum has coefficient $\sum_k y_k M_{kj}$ on u_j . Subtract this sum from the objective and use the equations:

$$\mathbf{c}^\top \mathbf{u} = \mathbf{y}^\top \mathbf{b} + \sum_{j=1}^N \left(c_j - \sum_{k=1}^m y_k M_{kj} \right) u_j. \quad (\text{I.2})$$

If every coefficient in parentheses is nonnegative, every term in the last sum is nonnegative. The objective is then at least $\mathbf{y}^\top \mathbf{b}$. In matrix notation the coefficient test is $M^\top \mathbf{y} \leq \mathbf{c}$, coordinate by coordinate. The multipliers y_k may have either sign because they multiply *equalities*. Only the remainders multiplying the nonnegative variables must have the stated sign.

The saved vectors $\mathbf{y}$ are called *dual certificates*. For each one, the verifier reconstructs $M, \mathbf{b}, \mathbf{c}$ from the mathematical model, checks all the remainder coefficients, and computes $\mathbf{y}^\top \mathbf{b}$ exactly. The files store the family and optional fixed-edge parameters in `model`, the objective vector in `objective`, the multiplier vector in `dual`, and the resulting rational bound in `exact_lower_bound`. The matrix and right-hand side are reconstructed from the model parameters and the counting definitions. In the file format, the fields `s`, `t`, and `n` encode the parameters s, r, n of $\mathcal{R}(s, r, n)$, respectively.

The stored triple-identity row is twice the equation displayed above, so that its coefficients are integral. Multiplying an equation by two and dividing its multiplier by two leaves the weighted sum unchanged. This correspondence allows the displayed equation and the stored certificate row to be checked against each other.

I.6 Obtaining all required local profile bounds

We can now explain the passage from the small catalogues to the 38–45-vertex local graphs. For example, let $X \in \mathcal{R}(4, 6, 38)$. Its neighborhoods are in $\mathcal{R}(3, 6, d_v)$ and its nonneighborhoods in $\mathcal{R}(4, 5, 37 - d_v)$. The bounds $R(3, 6) = 18$ and $R(4, 5) = 25$ give $13 \leq d_v \leq 17$. The neighborhood profiles are therefore supplied by complete catalogues on 13 through 17 vertices. The nonneighborhoods have orders 20 through 24. At order 24 the catalogue is complete; at orders 20 through 23, complete extreme-edge catalogues are combined with the outer polygons described next.

For the other three local families the degree restrictions are obtained in the same way. The complete list of orders needed on the two sides is

ambient family	order of A_v	order of B_v
$\mathcal{R}(4, 6, 38)$	13–17	20–24
$\mathcal{R}(4, 6, 39)$	14–17	21–24
$\mathcal{R}(5, 5, 44)$	19–24	19–24
$\mathcal{R}(5, 5, 45)$	20–24	20–24.

In the last two lines, A_v is a $(4, 5)$ -graph and B_v is a $(5, 4)$ -graph. Complementing B_v makes it a $(4, 5)$ -graph, so the same catalogue and polygon inputs suffice. For any graph Y of order m , the exact profile transformation is

$$\begin{aligned}
e(\overline{Y}) &= \binom{m}{2} - e(Y), \\
t(\overline{Y}) &= \binom{m}{3} - (m-2)e(Y) + q(Y), \\
q(\overline{Y}) &= 2\binom{m}{3} - (m-2)e(Y) + t(Y).
\end{aligned}$$

These formulas follow by interchanging edges and nonedges in the four possible induced triple types. They are affine, so they also transform containing polygons and convex combinations correctly.

For each order $19 \leq m \leq 23$ and every edge count in the interval of Section I.3, we need a polygon containing all profiles of $\mathcal{R}(4, 5, m)$ at that edge count. There are 149 such edge classes. For 26 of them the complete extreme-edge catalogues give the exact convex hull. For each of the other 123 classes, a smaller counting system is built using neighborhoods in $(3, 5)$ -graphs and nonneighborhoods in $(4, 4)$ -graphs. Both kinds have complete catalogue inputs of smaller order. Certificates of the form (I.2) then give upper and lower bounds on each of

$$t, \quad q, \quad t + q, \quad t - q, \quad t + 2q, \quad 2t + q, \quad t - 2q, \quad 2t - q.$$

Their values on a graph are integers, so the exact rational bounds can be rounded in the justified direction. Intersecting these half-planes gives the required polygon. The $123 \cdot 8 \cdot 2 = 1968$ bound certificates are included. Each uses the smaller catalogue inputs just listed, so its premises are available before the resulting polygon is used in a larger counting system.

The polygons are recorded in `computed_v3_45_19.json` through `computed_v3_45_23.json`. A polygon audit checks that the stored vertices describe the stated intersection. A separate certificate audit proves that every bounding half-plane contains the required graph profiles. Together these checks establish that the stored polygon contains every profile in its edge class.

I.6.1 The four edge bounds

Applying these counting systems gives the four statements

$$\begin{aligned} Y \in \mathcal{R}(4, 6, 38) &\implies e(Y) \leq 314, \\ Y \in \mathcal{R}(4, 6, 39) &\implies e(Y) \leq 324, \\ Y \in \mathcal{R}(5, 5, 44) &\implies e(Y) \geq 432, \\ Y \in \mathcal{R}(5, 5, 45) &\implies e(Y) \geq 460. \end{aligned} \tag{I.3}$$

Each statement follows from the counting system of Section I.4 using the smaller catalogue and polygon inputs above. The certificates are as follows.

1. `edge_46_38_max.json` gives $-e(Y) \geq -26125/83$. Since $26125/83 < 315$ and $e(Y)$ is an integer, $e(Y) \leq 314$.
2. `edge_46_39_max.json` first gives $e(Y) \leq 325$. For the remaining case $e(Y) = 325$, the files `exclude_4639e325_t_gt537.json` and `exclude_4639e325_t_lt538.json` give $t(Y) \geq 1612/3$ and $t(Y) \leq 1612/3$. A triangle count cannot be $1612/3$, so this edge count is excluded and $e(Y) \leq 324$.
3. `edge_55_44_min.json` gives $e(Y) \geq 95414/221 > 431$, hence $e(Y) \geq 432$.
4. `edge_55_45_min.json` gives $e(Y) \geq 12137627230191/26402663302 > 459$, hence $e(Y) \geq 460$.

For a nonempty family, the standard notation $e(s, r, n)$ and $E(s, r, n)$ denotes its minimum and maximum edge count. The universal statements above also make sense if a family is empty.

Return now to the hypothetical $X \in \mathcal{R}(5, 6, 84)$. The following table summarizes the bounds just proved, using the local family corresponding to each possible degree:

d_v	upper bound on $e(A_v)$	lower bound on $e(B_v)$
38	314	460
39	324	432

Because these are the only two degrees, the table can also be written

$$e(A_v) \leq 10d_v - 66, \quad e(B_v) \geq 1524 - 28d_v. \tag{I.4}$$

These affine formulas simply interpolate the two entries in each column. They alone do not contradict (I.1). The lower bounds for its summands are $460 - 314 - 152 = -6$ and $432 - 324 - 117 = -9$. To obtain more information we must retain t and q , not just edges.

I.6.2 Five inequalities retaining triangles and adjacent edges

For every vertex of $X \in \mathcal{R}(5, 6, 84)$ we prove the following additional inequalities, labelled A and B₁ through B₄:

$$\begin{aligned} \text{A : } & 770d_v - 269e(A_v) - 6t(A_v) + 12q(A_v) + 5940 \leq 0, \\ \text{B}_1 : & -732d_v - 76e(B_v) - t(B_v) + 2q(B_v) + 49239 \leq 0, \\ \text{B}_2 : & -701d_v - 75e(B_v) - t(B_v) + 2q(B_v) + 47596 \leq 0, \\ \text{B}_3 : & -483d_v - 35e(B_v) - 2t(B_v) + q(B_v) + 29877 \leq 0, \\ \text{B}_4 : & -403d_v - 37e(B_v) - t(B_v) + q(B_v) + 26296 \leq 0. \end{aligned}$$

In every line d_v is the degree of the original vertex in the 84-vertex graph; the order of B_v is $83 - d_v$.

The correspondence with the electronic files is

statement	A	B ₁	B ₂	B ₃	B ₄
file identifier	203	595	603	658	681.

These integers locate the five inequalities in `local_profile_facets.json`. The file `exact_candidate_used_facets.json` lists their coefficients and keys. The final certificate uses these five entries from the stored collection of local inequalities.

To prove a line, fix the local family and its integer edge count, build the counting system of Section I.4, and use a dual certificate as in Section I.5. The complete ranges which must be covered are

local family	$\mathcal{R}(4, 6, 38)$	$\mathcal{R}(4, 6, 39)$	$\mathcal{R}(5, 5, 44)$	$\mathcal{R}(5, 5, 45)$
$e_{\min}$	267	288	432	460
$e_{\max}$	314	324	514	530.

The endpoints are themselves certified. The file `edge_certification_summary.json` indexes the eight endpoint bounds and the two triangle-count exclusions used above; the full specification gives their exact values and filenames.

There are $48 + 37 = 85$ edge cases for statement A and $83 + 71 = 154$ for each statement B. Thus $85 + 4 \cdot 154 = 701$ certificates prove the five lines. For example, statement A at $d_v = 38$ and $e(A_v) = 314$ requires

$$-6t(A_v) + 12q(A_v) \leq 49266.$$

The file `facet203_46_n38_e314.json` proves $6t - 12q \geq -541956/11$, hence $-6t + 12q < 49272$. The expression $-6t + 12q$ is a multiple of six, so it is at most 49266. The counting equations hold for every graph in the fixed edge class, so the resulting inequality holds throughout that class.

For a general line with coefficients $\gamma_d, \gamma_e, \gamma_t, \gamma_q, \gamma_0$, the same argument bounds $\gamma_t t + \gamma_q q$ at each fixed edge count. The γ 's denote the fixed coefficients in the chosen line. The entire required local chain consists of the 1968 prerequisite polygon bounds, 10 endpoint/exclusion certificates, and 701 final edge cases: 2679 certificates. The checker `replay_required_local_certificates.py` verifies both their arithmetic and their coverage. It also checks that their objectives and integer rounding thresholds match the claimed inequalities.

I.7 A distribution on seven-vertex graphs

From now on X again denotes the hypothetical 84-vertex counterexample. Let $\mathcal{H}$ contain one representative of each isomorphism type of seven-vertex graph having no five-clique and no independent six-set. There are 979 such types. For each $H \in \mathcal{H}$, define

$$p_H = \frac{\#\{W \subseteq V(X) : |W| = 7, X[W] \cong H\}}{\binom{84}{7}}.$$

This is the probability of seeing type H when a seven-element vertex set is chosen uniformly. Therefore $p_H \geq 0$ and $\sum_{H \in \mathcal{H}} p_H = 1$. Every seven-set in X is represented because forbidden configurations in a subset would also be forbidden configurations in X .

The supplied code checks this set of types exhaustively. A labelled seven-vertex graph is determined by its 21 possible edges. The checker tests all 2^{21} adjacency choices and compares the allowed labelled counts with the orbit sizes of the 979 representatives, separately by edge count. Here an orbit consists of the distinct labelled graphs obtained by permuting a representative's seven vertices.

Our aim is now to express valid facts about X as linear equations and inequalities in the numbers p_H . A *coefficient row* is a function $U : \mathcal{H} \rightarrow \mathbb{Q}$: its entry $U(H)$ is the coefficient of p_H . Saying that this row is valid means $\sum_H p_H U(H) \leq 0$ for every possible X . This interpretation will be used for every row in the final certificate.

I.7.1 How a local fact becomes a coefficient row

A *rooted graph* has specified distinguished vertices, called roots. A *flag* is a small induced graph together with the roles of its roots. For example, one root may represent the vertex whose degree is being measured, or two roots may specify an edge. Isomorphisms must preserve these roles. A *placement* is a choice of actual vertices inducing the specified flag. Root labels and any explicitly unordered groups are counted consistently; the specification states the conventions of each row family.

Here is the simplest sampling calculation. Fix a placement of a six-vertex flag with root v , and let r_0 be the number of neighbors of v among the other five vertices. There are 78 vertices outside the flag. Choose one of them uniformly and let χ be 1 if it is adjacent to v , and 0 otherwise. Then

$$\mathbb{E}(\chi) = \frac{d_v - r_0}{78}, \quad \mathbb{E}(r_0 + 78\chi) = d_v.$$

The expectation here is over that one uniformly sampled exterior vertex, with the flag placement fixed. Thus $r_0 + 78\chi - 39$ is an estimate of $d_v - 39$ with exactly the right average. Its value can be computed using only the resulting seven-vertex graph. Since $d_v \leq 39$, summing this estimate over all placements of the flag and averaging over seven-sets gives a valid nonpositive coefficient row.

The general rule is equally finite. A fixed f -vertex flag leaves $84 - f$ exterior vertices; a seven-set containing it selects $7 - f$ of them. A specified configuration using m distinct exterior vertices is included with probability

$$\frac{\binom{7-f}{m}}{\binom{84-f}{m}}, \quad 0 \leq m \leq 7 - f.$$

Multiply its observed count by the reciprocal to obtain an estimate whose average equals the full count. Products of counts require care: when two factors use the same exterior vertex, that contribution has $m = 1$, not $m = 2$. All these coincidences are included.

For precision, let $I(X, \phi) \leq 0$ be the original inequality for a placement ϕ of a specified f -vertex flag. Let $\hat{I}(H, \phi)$ be its estimate on a seven-vertex graph, and put $U(H) = \sum_{\phi} \hat{I}(H, \phi)$, summing over placements in H . Interchanging the sums over seven-sets and placements gives

$$\sum_H p_H U(H) = \frac{\binom{84-f}{7-f}}{\binom{84}{7}} \sum_{\phi \text{ in } X} I(X, \phi) \leq 0.$$

The factor is positive and is the same for every placement. This establishes the row's validity after averaging over all placements. Equalities are treated in the same way. This passage is called *lifting* the local relation; it has no asymptotic error.

I.7.2 The local relations used in these rows

The inputs include the degree bounds, the degree identity $(d_v - 38)(d_v - 39) = 0$, the local edge bounds (I.4), and the five local inequalities. Further rows describe small common neighborhoods. To define them, take disjoint vertex sets C, D , with C a clique and D independent, and put

$$\mathcal{C}(C, D) = \{x \in V(X) \setminus (C \cup D) : x \sim c \text{ for all } c \in C, \quad x \not\sim d \text{ for all } d \in D\}.$$

Here $x \sim y$ means that x, y are adjacent. This *cell* is a vertex set; counts of edges and triangles in it refer to the induced graph $X[\mathcal{C}(C, D)]$. Its size is an integer. A clique in the cell extends by C , and an independent set in it extends by D . This gives a smaller Ramsey family and therefore bounds on the cell.

For example, the common neighborhood of the endpoints of an edge is a $(3, 6)$ -graph, so it has at most 17 vertices. It has at least 13: the neighbors of one endpoint, apart from the other endpoint and their common neighbors, form a $(4, 5)$ -graph and have at most 24 vertices. Since the endpoint has degree at least 38, the common neighborhood has size at least $38 - 1 - 24 = 13$. The complete $(3, 6)$ catalogues on orders 13 through 17 therefore cover this cell exactly.

The rows use upper capacities, edge/profile inequalities for these cells, and elementary integer inequalities. For instance, if c is an integer cell size and k is an integer, then $(c - k)(c - k - 1) \geq 0$: no integer lies strictly between k and $k + 1$. The quantities $d_v - 38$ and $39 - d_v$ are either zero or one, so their products also give nonnegative expressions. Each is expanded into counts to which the sampling rule applies.

One further input uses the seven graphs in the complete $\mathcal{R}(3, 6, 17)$ catalogue. Their degrees are all 4 or 5. For a triangle with vertices u, v, w , let

$$c = |N_X(u) \cap N_X(v)|, \quad \tau = |N_X(u) \cap N_X(v) \cap N_X(w)|.$$

The vertex w belongs to that common-neighborhood graph. If $c = 17$, its degree there is $\tau \in \{4, 5\}$. If $c = 16$, its nonneighbors there form a $(3, 5)$ -graph of order at most 13, giving $\tau \geq 2$. For $c \leq 15$, $\tau \geq 0$ suffices. Consequently

$$\tau \geq 2c - 30, \quad (16 - c)(\tau - 4)(\tau - 5) \geq 0.$$

For the second inequality, at $c = 17$ the last two factors have product zero; for $c \leq 16$ they are consecutive integer factors with nonnegative product. These facts produce additional valid rows by the same exact sampling procedure. The specification lists all row families and their formulas; the source audits check that the stored arrays implement those formulas.

I.8 Why sums of squares give further inequalities

We need one additional kind of valid row. Its underlying fact is simply that a sum of squares is nonnegative. We describe how to express that fact using seven-vertex graphs, including overlaps exactly.

Fix a labelled root graph σ with ℓ vertices, and an extension size $k \geq 0$ such that $\ell + 2k \leq 7$. A root placement ρ is an injection of those ℓ labelled vertices into $V(X)$ which induces σ . Write R_ρ for its image. Let $\mathcal{F}_{\sigma, k}$ be the finite collection of possible rooted isomorphism types on $\ell + k$ vertices, excluding a five-clique and an independent six-set. The roots are fixed by isomorphisms; the k other vertices may be permuted.

For each type $i \in \mathcal{F}_{\sigma, k}$ define

$$f_i(\rho) = \#\{S \subseteq V(X) \setminus R_\rho : |S| = k, \quad (X[R_\rho \cup S], \rho) \text{ has type } i\}.$$

This is an integer count of extensions of a fixed root placement. Collect these counts into a column vector $f(\rho)$. For any real coefficient vector $z = (z_i)_{i \in \mathcal{F}_{\sigma,k}}$,

$$\sum_{\rho} \left(\sum_i z_i f_i(\rho) \right)^2 \geq 0.$$

Equivalently, the matrix $\sum_{\rho} f(\rho)f(\rho)^{\top}$ is *positive semidefinite*: a symmetric matrix M has that property when $z^{\top} M z \geq 0$ for every real vector z . This explains both the matrix terminology and the inequality it encodes.

To expand an entry of this matrix, choose a root placement and an ordered pair (S, T) of k -element extension sets. The sets may intersect, and $S = T$ is allowed. Their union with the roots has $\ell + |S \cup T| \leq 7$ vertices, and is contained in exactly

$$D(S, T) = \binom{84 - \ell - |S \cup T|}{7 - \ell - |S \cup T|}$$

seven-sets of X . This positive integer tells us how often the same pair would otherwise be counted.

For a seven-vertex graph H , form a matrix M_H with rows and columns indexed by $\mathcal{F}_{\sigma,k}$. Enumerate root placements in H and all ordered extension pairs (S, T) outside the roots. If S has type i and T has type j , add $1/D(S, T)$ to entry (i, j) . The denominator uses the target order 84, even though this small calculation takes place in H . Summing over all seven-sets of X counts each extension pair with total weight exactly one. Hence

$$\sum_{\substack{W \subseteq V(X) \\ |W|=7}} M_{X[W]} = \sum_{\rho} f(\rho)f(\rho)^{\top}.$$

Dividing by $\binom{84}{7}$ shows that $\sum_H p_H M_H$ is positive semidefinite. In particular, for any chosen z ,

$$\sum_H p_H (-z^{\top} M_H z) \leq 0.$$

This has exactly the coefficient-row form needed in Section 1.7.

We use the pairs $(\ell, k) = (0, 3), (1, 3), (2, 2), (3, 2)$. There are respectively one, one, two and four root graph types, so eight matrices of this kind. For each root type choose one positive integer L clearing every denominator in its matrices, and set $I_H = L M_H$. The data supply 448 integer vectors z distributed among these eight types. Each supplies the valid row $H \mapsto -z^{\top} I_H z$. The enumeration includes all intersections of the extension sets, as required by the displayed matrix identity.

I.9 The final rational sum and its contradiction

All the rows now have a defined mathematical meaning. There are 4876 rows from local counting relations and 448 from the square construction. Index their normalized coefficient functions by $U_i : \mathcal{H} \rightarrow \mathbb{Q}$, $0 \leq i < 5324$. There are also 89 homogeneous equality rows $E_j : \mathcal{H} \rightarrow \mathbb{Q}$, $0 \leq j < 89$. Thus a hypothetical counterexample would give

$$\sum_H p_H U_i(H) \leq 0, \quad \sum_H p_H E_j(H) = 0.$$

A homogeneous equality has right-hand side zero. The equation $\sum_H p_H = 1$ is kept separate; it is not one of the E_j rows.

For an integer coefficient row $(a_H)_H$, normalization divides it by $\max\{1, \max_H |a_H|\}$. This is a positive number, so validity and sign are unchanged. The saved multipliers refer to these normalized rows, and the verifier uses that same normalization.

The file `exact_separation_certificate.json` supplies rational weights $\lambda_i \geq 0$ for the inequalities and rational weights μ_j of either sign for the equalities. They have common denominator 10^{12} . For every $H \in \mathcal{H}$, form the number

$$C(H) = \sum_i \lambda_i U_i(H) + \sum_j \mu_j E_j(H).$$

This is just a weighted sum of coefficients in one column of the stored arrays. The exact arithmetic check proves

$$C(H) \geq \delta \quad \text{for every } H \in \mathcal{H}, \quad \delta = \frac{3146981978575629437}{10^{24}} > 0.$$

The check obtains a conservative lower bound by rounding each scaled summand down using integer division, including negative summands. It therefore proves the displayed inequality without a floating-point tolerance. The actual coefficients $C(H)$ may exceed this common lower bound.

Only 903 multipliers are nonzero: 816 for inequalities, of which 194 are square rows, and 87 for homogeneous equalities. These weights give two incompatible conclusions. First, because the inequality weights are nonnegative and all the rows are valid,

$$\sum_H p_H C(H) = \sum_i \lambda_i \sum_H p_H U_i(H) + \sum_j \mu_j \sum_H p_H E_j(H) \leq 0.$$

Second, because the p_H are nonnegative and sum to one,

$$\sum_H p_H C(H) \geq \delta \sum_H p_H = \delta > 0.$$

This contradiction excludes X and proves the theorem.

The complete square calculations have been recomputed with arbitrary-precision integers. A second implementation uses integer contractions with a separately checked overflow bound. Both reproduce the separating calculation.

I.10 What an independent reader should check

There are two complementary parts to checking this proof. The mathematical part establishes why an arbitrary counterexample would satisfy the equations and inequalities. The computational part checks that the supplied rational numbers actually combine those relations as claimed. Neither part replaces the other.

I.10.1 Reading the mathematics

The chain to be checked is as follows.

1. Confirm the cited external bounds and the precise scope of each catalogue in Section I.3.
2. Check the graph-to-variable construction in Section I.4. Every actual neighborhood profile must lie in its designated polygon; its convex representation must preserve the local counts. Verify the displayed double counts.
3. Check the weighted-equation principle (I.2) and the exact integer-rounding steps. The worked explanation develops the 314-edge bound in detail, reducing its certificate to ten local inequalities and one identity that can be followed on paper.

4. Check that each local fact used globally becomes a valid row by the finite sampling rule, and that the square rows include every possible overlap. The detailed specification gives all row formulas and their index conventions for comparison with the programs.
5. Check the two final averaging inequalities in Section I.9. Once their premises have been established, this last step is an ordinary finite contradiction.

The two companion documents are titled *Reading the exact certificate: the method and a worked local example* and *Detailed mathematical specification*. The first is for following a concrete calculation; the second is for checking all definitions and reconstructing the computations. The certificate files supply the numerical coefficients used in those computations.

I.10.2 Replaying the supplied calculations

The complete electronic distribution is `verification.tar.xz`. Extracting it creates a directory named `verification`. In that directory, `README.txt` gives installation instructions and `FILE_INDEX.tsv` locates each basename cited in this paper. Keep the extracted directory structure intact so that the programs can locate their inputs.

For the worked local example, run the following command with the terminal's current directory set to the extracted `verification` directory:

```
python3 check_local_example.py --root .
```

This requires only the Python standard library. A successful run reports

```
"worked_example_passed": true
"edge_upper_bound": 314
```

It checks the explicit local inequalities on the supplied profiles and polygon vertices and checks their relation to the saved certificate. It does not by itself prove the prerequisite polygon bounds or catalogue completeness.

For the main certificate and its computational dependencies, install the packages specified in `requirements.txt`, following the `README`, and run from that same directory:

```
python3 verify_package.py
```

The launcher first checks file integrity, then runs the 18 core checks, the source-to-matrix audit, and the worked-example check. Every stage must exit successfully and the final report must contain `"all_passed": true`. A success message from only one intermediate stage is insufficient.

For the additional exhaustive profile-extraction replay, install a C++11 compiler and `zlib` as specified in the `README`, and run

```
python3 verify_package.py --full
```

This also runs the supplementary sampling checks and the separate consequence below. The extraction replay processes all 58 supplied raw catalogue files using the original C++ extractor. Independent NetworkX profile checks cover selected records. Catalogue completeness is the external input stated in Section I.3.

The local audit `replay_required_local_certificates.py` checks all 2679 required certificates and their dependencies. The polygon and coverage audits check that no needed degree or edge class has been omitted. The package audit `audit_packaged_model.py` checks the link between stored global rows and their source arrays, including the five selected local inequalities. The final arithmetic is replayed by `replay_global_certificate.py` and `audit_global_exact_certificate.py`. Together these checks connect the mathematical definitions, the stored coefficients, and the final weighted sum.

A compact distribution omits the 58 raw catalogues and supplies `restore_catalogues.py` to restore their exact recorded bytes. Its `START_HERE.txt` gives the required extra step. The full distribution requires no such restoration. Missing inputs, hash mismatches, a nonzero exit status, or failed assertions mean that the reproduction has not succeeded. Do not run optimized Python (`-O` or `-OO`), since the checks use assertions. No optimization solver is required for any of these replay commands.

A separate consequence, not used above

A further audited consequence of the 2018 $R(5, 5) \leq 48$ computation [I.5, Sections 2 and 6] is $e(5, 5, 45) \geq 462$. The proof above uses only the bound 460; this additional result is not one of its premises. Its separate derivation is summarized below.

Let Y be a $(5, 5)$ -graph of order 44 or 45, and let D_{24} be the set of its vertices of degree 24. The cited gluing result and the complete $(4, 5, 24)$ catalogue imply that $Y[D_{24}]$ has maximum degree at most eight. Here gluing means combining two neighborhoods along their common induced graph. In the cited computation, cores smaller than 12 common neighbors either yield no graph or cannot be extended by even one vertex. The subsequent catalogue calculation, using the degree profiles, $e(4, 5, 23) = 101$, and the three 22-vertex $(4, 5)$ -graphs with 88 edges, proves $|D_{24}| \leq 21$. The full case argument is in `degree24_reduction.txt`; the independent census code is `audit_degree24.py`.

To see why this gives the edge bound, take $X \in \mathcal{R}(5, 5, 45)$. Both X and its complement have maximum degree at most 24 because $R(4, 5) = 25$. Hence every degree in X is at least $44 - 24 = 20$. A degree-20 vertex in X has degree 24 in its complement, so there are at most 21 such vertices. Every other vertex has degree at least 21. Therefore

$$2e(X) \geq 21 \cdot 20 + 24 \cdot 21 = 924, \quad e(X) \geq 462.$$

The nontrivial census statement $|D_{24}| \leq 21$ belongs to the separate argument just cited, not to this last handshaking calculation.

Acknowledgment

The author was supported by the NRD grant KKP 138270.

References

- [I.1] S. P. Radziszowski, *Small Ramsey Numbers*, Electronic Journal of Combinatorics, Dynamic Survey DS1, revision 18, 24 April 2026, especially Table Ib, p. 7. <https://doi.org/10.37236/21>; <https://www.cs.rit.edu/~spr/ElJC/ejcram18.pdf>
- [I.2] V. Angelstveit and B. D. McKay, $R(5, 5) \leq 46$, Journal of Graph Theory **112** (2026), 198–208. Theorem 1.1; Appendix A, Table 1. <https://doi.org/10.1002/jgt.70029>; <https://arxiv.org/html/2409.15709v2>
- [I.3] B. D. McKay and S. P. Radziszowski, *Subgraph counting identities and Ramsey numbers*, J. Combinatorial Theory, Ser. B **69** (1997), 193–209. <https://users.cecs.anu.edu.au/~bdm/papers/r55.pdf>
- [I.4] B. D. McKay, *Ramsey Graphs*, complete graph catalogues. <https://users.cecs.anu.edu.au/~bdm/data/ramsey.html>
- [I.5] V. Angelstveit and B. D. McKay, $R(5, 5) \leq 48$. Journal of Graph Theory **89** (2018), 5–13. <https://arxiv.org/abs/1703.08768>

- [I.6] J. Goedgebeur and S. P. Radziszowski, *New Computational Upper Bounds for Ramsey Numbers $R(3, k)$* , Electronic Journal of Combinatorics **20**(1) (2013), P30. Theorem 2. <https://doi.org/10.37236/2824>.
- [I.7] *Electronic supplement to the present paper: exact local Ramsey-profile certificates* (2026). Mathematical description and provenance: `LOCAL_CERTIFICATES.txt`. Counting model: `recursive_profiles_v4.py`. Certificate index: `edge_certification_summary.json`. Independent verifier: `replay_required_local_certificates.py`.

Part II

Reading the exact certificate for $R(5, 6) \leq 84$: the method and a worked local example

Explanatory supplement

29 October 2026

Abstract

This supplement develops a worked example from the computer-assisted proof of $R(5, 6) \leq 84$. We prove that a graph on 38 vertices containing neither a clique of size four nor an independent set of size six has at most 314 edges. The proof starts with ten inequalities for edge and triangle counts and pairs of incident edges in neighborhoods and nonneighborhoods. Summing these inequalities and applying counting identities cancels all quantities except the total number of edges. We give the complete calculation, identify the finite checks that justify the ten inequalities, and show how the same calculation is recorded by rational equation multipliers in the verification files. We then explain how further valid counting inequalities are combined with rational weights to rule out a counterexample on 84 vertices. The final sections give commands for checking the worked example and replaying the complete certificate.

II.1 Notation and the information retained from a graph

Write $\mathcal{R}(s, r, n)$ for the graphs on n vertices containing neither a clique of size s nor an independent set of size r . All graphs are finite and simple: edges are unordered pairs of distinct vertices. A clique is a set of pairwise adjacent vertices; an independent set is a set with no edges between its vertices. Two graphs have the same isomorphism type if a bijection of their vertex sets preserves edges and nonedges. The symbol K_m denotes the complete graph on m vertices. For a graph X , write $V(X)$ and $E(X)$ for its vertex and edge sets and put $n = |V(X)|$ and $e(X) = |E(X)|$. The number $t(X)$ is the number of unordered three-subsets inducing a triangle. Write $N_X(v)$ for the neighbor set of v , $d_v = d_X(v) = |N_X(v)|$, and $X[S]$ for the subgraph induced by $S \subseteq V(X)$. Unqualified vertex sums run over $V(X)$. Put

$$e = e(X), \quad t = t(X), \quad w = w(X) = \sum_{v \in V(X)} \binom{d_v}{2}, \quad q = q(X) = w - t.$$

Here w counts unordered pairs of distinct incident edges. An induced three-vertex path contributes one to w , whereas a triangle contributes three. Consequently

$$q = \#\{\text{induced three-vertex paths}\} + 2t. \quad (\text{II.1})$$

For example, K_3 has $(e, t, q) = (3, 1, 2)$, a three-vertex path has $(2, 0, 1)$, and a matching of two edges has $(2, 0, 0)$.

For $0 \leq k \leq 3$, let $T_k(X)$ count the three-subsets of $V(X)$ inducing exactly k edges. These four counts can be recovered from the profile (e, t, q) :

$$\begin{aligned} T_0(X) &= \binom{n}{3} - (n-2)e + q, \\ T_1(X) &= (n-2)e - 2q + t, \\ T_2(X) &= q - 2t, \\ T_3(X) &= t. \end{aligned} \quad (\text{II.2})$$

Indeed, $T_3 = t$ and $T_2 = q - 2t$ follow from (II.1). Counting incidences between edges and triples gives $T_1 + 2T_2 + 3T_3 = (n-2)e$, and the four counts sum to $\binom{n}{3}$.

For each vertex v , let

$$A_v = X[N_X(v)], \quad B_v = X[V(X) \setminus (N_X(v) \cup \{v\})].$$

If $X \in \mathcal{R}(s, r, n)$ and $d_v = d$, then

$$A_v \in \mathcal{R}(s-1, r, d), \quad B_v \in \mathcal{R}(s, r-1, n-1-d). \quad (\text{II.3})$$

The forbidden clique in the first assertion would extend by v ; the forbidden independent set in the second would extend by v .

A *profile* of a graph is the tuple (n, e, t, q) . Many nonisomorphic graphs can have the same profile. A profile file lists the distinct tuples obtained from a graph catalogue. Its header records both the number of graphs processed and the number of distinct profiles. An inequality depending only on these four statistics can therefore be checked once for each distinct profile.

II.2 What the worked example establishes

We first prove that every $X \in \mathcal{R}(4, 6, 38)$ has at most 314 edges. In the proof of $R(5, 6) \leq 84$, such a graph occurs as the neighborhood of a degree-38 vertex in a hypothetical graph on 84

vertices. After deriving the 314-edge bound, we explain how it enters the counting argument for that hypothetical graph.

The proof of the 314 bound has two tasks. First, prove ten inequalities for the smaller graphs A_v and B_v . Second, add those inequalities and use counting identities to cancel all unknown quantities except $e(X)$. The cancellation is written out in Section II.6. The input checks for the ten inequalities are described in Sections II.4 and II.7.

A *certificate* in this discussion is a recorded choice of rational coefficients for such an addition of equations and inequalities. The certificate specifies a weighted sum of valid counting statements. Its verification consists of proving those statements and checking that the specified sum has the claimed coefficients and sign.

The counting proofs below establish that every graph in the relevant family satisfies the equations. The weighted sum for the worked example can be checked on paper. The remaining finite calculation substitutes profiles into the ten local inequalities; Section II.15 gives the command that performs these substitutions, followed by the command for the complete 84-vertex proof.

Only the smaller graph families used as local inputs are enumerated. The families $\mathcal{R}(4, 6, 38)$ and $\mathcal{R}(5, 6, 84)$ are treated through necessary equations and inequalities for their subgraph counts. These conditions give the edge bound in the first family and a contradiction in the second.

II.3 The first triangle double count

We begin with a counting identity used at both the local and global levels. Each triangle is an edge in the neighborhood of each of its three vertices, so

$$\sum_v e(A_v) = 3t(X). \quad (\text{II.4})$$

For the nonneighborhoods, fix an edge ab . It appears in B_v precisely when v is distinct from and nonadjacent to both a and b . The number of such vertices is

$$n - d_a - d_b + |N_X(a) \cap N_X(b)|.$$

Summing over edges gives

$$\sum_v e(B_v) = ne(X) - \sum_v d_v^2 + 3t(X). \quad (\text{II.5})$$

In the last term, each triangle is counted once at each of its three edges. Subtracting (II.4) and using $\sum_v d_v = 2e(X)$ yields

$$\sum_v \left(e(B_v) - e(A_v) - d_v \left(\frac{n}{2} - d_v \right) \right) = 0. \quad (\text{II.6})$$

This is the three-vertex counting identity used in the certificate model; see McKay and Radziszowski [II.1] and the formulation in Angeltveit and McKay [II.2].

For a hypothetical $X \in \mathcal{R}(5, 6, 84)$, the external bounds $R(4, 6) \leq 40$ and $R(5, 5) \leq 46$ imply

$$d_v \leq 39, \quad 83 - d_v \leq 45, \quad \text{hence } d_v \in \{38, 39\}.$$

At order 84, (II.6) is

$$\sum_v (e(B_v) - e(A_v) - d_v(42 - d_v)) = 0.$$

The subtracted quantity is 152 at degree 38 and 117 at degree 39. The following table summarizes four computational bounds used in the main proof. Each entry bounds the number of edges of

every graph in its corresponding local family:

d_v	$e(A_v)$ at most	$e(B_v)$ at least
38	314	460
39	324	432.

They alone do not give a contradiction: they make the corresponding summands at least -6 and -9 . The full proof therefore retains further local profile information and uses it in a larger counting model. The next sections explain the first entry, 314, completely at the level of its counting argument and its specified finite inputs.

II.4 The worked example: its claim and local inputs

Proposition II.4.1. *Subject to the catalogue and certified profile inputs stated below, every $X \in \mathcal{R}(4, 6, 38)$ satisfies $e(X) \leq 314$.*

The inputs have two sources. Catalogue completeness is supplied by the cited mathematical results. Containment in the outer profile polygons is established by rational certificates in the verification package. Both kinds of input provide inequalities for the smaller graphs that can occur around a vertex of X .

A complete catalogue of $\mathcal{R}(s, r, m)$ lists a representative of every isomorphism type in that family. Thus every graph with the specified order and forbidden subgraphs is isomorphic to a listed graph. The catalogues used here are supplied by McKay [II.3]; their precise families are given below. A statement verified on every listed graph applies to the entire family by this completeness property.

Fix such a graph X . Since $R(3, 6) = 18$ and $R(4, 5) = 25$, equation (II.3) gives

$$13 \leq d_v \leq 17, \quad A_v \in \mathcal{R}(3, 6, d_v), \quad B_v \in \mathcal{R}(4, 5, 37 - d_v).$$

Thus A_v is triangle-free, while the order of B_v is between 20 and 24. Put

d	13	14	15	16	17
α_d	346	280	179	80	-48
$\beta_d = 3d - 38$	1	4	7	10	13
K_d	184	143	131	111	114.

(II.7)

The ten local assertions needed for the proof are

$$\alpha_d + (3d - 38)e(A_v) - 3q(A_v) \geq 0, \tag{II.8}$$

$$K_d - 3e(B_v) + 3t(B_v) \geq 0. \tag{II.9}$$

The subscript d specifies which of the five rows of constants is used. For example, the $d = 13$ assertions concern all possible neighborhoods of order 13 and all possible nonneighborhoods of order 24 in X . There are five assertions on triangle-free graphs and five on $(4, 5)$ -graphs. Proving each assertion for every graph in its family allows it to be applied at every vertex of X , whatever X may be.

For (II.8), the input consists of the complete $\mathcal{R}(3, 6, d)$ catalogues for $13 \leq d \leq 17$, supplied by McKay [II.3]. Their extracted distinct profiles occur in the files with basenames `r36_13.g6.gz.profiles` through `r36_16.g6.gz.profiles`, and `r36_17.g6.profiles`. The degree 13 case, for instance, is the concrete inequality

$$3q(Y) \leq 346 + e(Y) \quad (Y \in \mathcal{R}(3, 6, 13)).$$

Checking it amounts to substituting every distinct catalogue profile.

For (II.9), order 24 uses the complete catalogue represented by `r45_24.g6.profiles`. Orders 20–23 use the outer profile polygons stored in `computed_v3_45_20.json` through `computed_v3_45_23.json`. These polygons contain all possible (t, q) pairs at their specified edge counts. A polygon may also contain points that are not profiles of any graph. An inequality valid throughout the polygon still applies to every graph profile contained in it.

Precisely, for any order m , clique parameter s , independent-set parameter r , and integer $0 \leq \eta \leq \binom{m}{2}$, define

$$\mathcal{S}_{s,r,m,\eta} = \{(t(Y), q(Y)) : Y \in \mathcal{R}(s, r, m), e(Y) = \eta\}.$$

A profile polygon $P_{s,r,m,\eta} \subseteq \mathbb{R}^2$ is a bounded convex region containing this set, with coordinates denoted (τ, κ) . In this plane, an inequality such as $\alpha\tau + \beta\kappa \leq c$ defines a half-plane, and finitely many such inequalities bound the region. All coefficients and vertices used here are rational. A segment or point is allowed. An empty polygon is allowed only if the corresponding graph family with edge count η has been excluded. For a complete catalogue we can take the convex hull of its actual (t, q) pairs at edge count η . Otherwise the certified inequalities define an outer polygon. Write $V_{s,r,m,\eta} = \text{vert } P_{s,r,m,\eta}$ for its finite vertex set. In this example the neighborhood inputs are $P_{3,6,d,\eta}$ and the nonneighborhood inputs are $P_{4,5,37-d,\eta}$, for $13 \leq d \leq 17$. A polygon vertex may have noninteger coordinates and need not be the profile of any graph. Every actual profile is a convex combination of these vertices at its own, unchanged, integer edge count. This is why linear inequalities checked on the polygon's vertices hold for all actual profiles it contains.

Explicitly, a convex combination of points $p_1, \dots, p_k$ is $\lambda_1 p_1 + \dots + \lambda_k p_k$, with every $\lambda_i \geq 0$ and $\lambda_1 + \dots + \lambda_k = 1$. For an affine function L , its value at this combination is $\sum_i \lambda_i L(p_i)$. If all $L(p_i)$ are nonnegative, so is the value at the actual profile. Thus checking the polygon vertices proves the inequality for every graph whose profile lies in the polygon. The edge count η remains fixed during the combination; there is a separate polygon for each allowed edge count.

II.5 A second identity, counted around triangles

A second counting identity eliminates the profile terms that remain after applying (II.6).

Lemma II.5.1. *For every graph X on n vertices,*

$$\sum_v (d_v e(A_v) - q(A_v) + t(B_v)) = (n + 3)t(X). \quad (\text{II.10})$$

Proof. For $0 \leq i \leq 3$, let b_i count pairs (T, z) in which T is a triangle, $z \notin T$, and z has exactly i neighbors in T . Every triangle has $n - 3$ possible external vertices, so

$$b_0 + b_1 + b_2 + b_3 = (n - 3)t(X). \quad (\text{II.11})$$

First,

$$\sum_v t(B_v) = b_0. \quad (\text{II.12})$$

Indeed, a triangle in B_v is precisely a triangle with an external vertex v having no neighbors in it.

Next, each triangle abc contributes $d_a + d_b + d_c$ to $\sum_v d_v e(A_v)$. Its three internal edges contribute six to this degree sum; its external vertices contribute their numbers of neighbors in the triangle. Hence

$$\sum_v d_v e(A_v) = 6t(X) + b_1 + 2b_2 + 3b_3. \quad (\text{II.13})$$

Finally,

$$\sum_v q(A_v) = b_2 + 2b_3. \quad (\text{II.14})$$

One way to see this last equality is to inspect its possible four-vertex configurations. An induced K_4 minus an edge has two universal vertices, meaning vertices adjacent to all three others. In either universal vertex's neighborhood the other three vertices form an induced path; it therefore contributes two to the left side. It also contains two choices of triangle with an external vertex adjacent to exactly two triangle vertices, and contributes two to b_2 . A K_4 contributes $q(K_3) = 2$ at each of its four vertices, giving eight; it contributes four to b_3 , also giving eight on the right. No other induced four-vertex type contributes to either side.

Combining (II.12)–(II.14), the left side of (II.10) becomes $6t + b_0 + b_1 + b_2 + b_3$. Equation (II.11) completes the proof. $\square$

For the worked example $n = 38$, the right side is $41t(X)$. Although X is K_4 -free in that application, the identity itself is universal.

II.6 The complete weighted counting argument

Define the local nonnegative expressions

$$L_A(v) = \alpha_{d_v} + (3d_v - 38)e(A_v) - 3q(A_v), \quad L_B(v) = K_{d_v} - 3e(B_v) + 3t(B_v).$$

Their nonnegativity is exactly (II.8) and (II.9). The constants in (II.7) satisfy

$$\alpha_d + K_d = 1375 - 26d - 3d^2 \quad (13 \leq d \leq 17). \quad (\text{II.15})$$

Checking the five rows of the table verifies this polynomial expression for the combined constant term. It allows the sum of local inequalities to be evaluated using the vertex degrees:

$$\begin{aligned} \sum_v (L_A(v) + L_B(v)) &= \sum_v (1375 - 26d_v - 3d_v^2) \\ &\quad + 3 \sum_v (d_v e(A_v) - q(A_v) + t(B_v)) \\ &\quad - 38 \sum_v e(A_v) - 3 \sum_v e(B_v) \\ &= 38 \cdot 1375 - 52e - 3 \sum_v d_v^2 \\ &\quad + 3 \cdot 41t - 38 \cdot 3t - 3 \left(38e - \sum_v d_v^2 + 3t \right) \\ &= 52250 - 166e. \end{aligned}$$

The second equality uses, in order, the handshaking identity, Lemma II.5.1, (II.4), and (II.5). Both the degree-square terms and the triangle terms cancel.

The resulting identity is the certificate for the edge bound:

$$\boxed{52250 - 166e(X) = \sum_v (L_A(v) + L_B(v)) \geq 0.} \quad (\text{II.16})$$

It follows that

$$e(X) \leq \frac{52250}{166} = \frac{26125}{83} = 314 + \frac{63}{83} < 315.$$

Since $e(X)$ is an integer, $e(X) \leq 314$, proving Proposition II.4.1.

This calculation reduces the edge bound to the ten local inequalities (II.8) and (II.9). The next section lists the finite profile sets on which they are checked and the exact extremal values obtained by the checker.

II.7 Checking the ten local inequalities

For the A side, the following table records the complete profile lists used, the number of graphs from which they were extracted, and the minimum of $L_A = \alpha_d + (3d - 38)e - 3q$ on those lists.

d	Graphs	Distinct profiles	min L_A	One minimizing (e, t, q)
13	275086	138	0	(32, 0, 126)
14	263520	118	0	(35, 0, 140)
15	64732	68	0	(37, 0, 146)
16	2576	31	0	(40, 0, 160)
17	7	3	0	(42, 0, 166)

The degree 13 minimizing example gives $346 + 32 - 3 \cdot 126 = 0$. The degree 17 example gives $-48 + 13 \cdot 42 - 3 \cdot 166 = 0$. These substitutions illustrate the arithmetic, but the universal statement requires checking every profile in the relevant list.

For the B side, L_B depends only on $e - t$, not on q . Thus it is enough to maximize $e - t$ on the appropriate profile set. At order 24 the set is the catalogue's distinct profiles; at orders 20–23 it is the union of the vertices of the certified polygons at each edge count. The results are

d	$ B_v $	Profile points	max($e - t$)	min L_B	Minimizing (e, t, q)
13	24	1488	−2	190	(118, 120, 924)
14	23	184	21	80	(105, 84, 771)
15	22	235	31	38	(94, 63, 649)
16	21	268	37	0	(83, 46, 528)
17	20	299	38	0	(75, 37, 453)

For example, at $d = 15$ the largest possible $e - t$ in the outer profile set is 31, giving $L_B \geq 131 - 3 \cdot 31 = 38$. At $d = 16$, the corresponding calculation is $111 - 3 \cdot 37 = 0$.

An affine function on a polygon achieves its minimum at a vertex. Consequently, checking polygon vertices is sufficient even if the polygon contains many additional integer points or unrealizable profiles. When actual profiles are used directly, completeness of the underlying catalogue establishes that every graph profile has been checked.

The program `check_local_example.py` performs these substitutions using exact rational arithmetic. It also checks the polynomial identity (II.15) and the agreement with the saved equation multipliers. The program computes the minima from the supplied profile sets.

II.8 Where the profile restrictions come from

The complete triangle-free catalogues and the complete order 24 $(4, 5)$ catalogue are external inputs [II.3]. The supplied graph6 files encode the graphs. The extractor `catalogue_profiles.cpp` computes their exact profiles. The distinct-profile sets can be regenerated exhaustively with `regenerate_catalogue_profiles.py` and compared with the supplied sets. The independent catalogue audit additionally checks graph counts and sampled profile computations by a different graph decoder. Sampling is not an exhaustive independent recomputation, and none of these checks re-proves catalogue completeness.

The order 20–23 $(4, 5)$ profile polygons are derived from still smaller graphs. For a fixed order m and edge count η , the unknown graph has neighborhoods in $\mathcal{R}(3, 5, d)$ and nonneighborhoods in $\mathcal{R}(4, 4, m - 1 - d)$. Their complete smaller catalogues supply admissible profiles: the $(3, 5)$ catalogues have order at most 13 and the $(4, 4)$ catalogues have order at most 17. Counting triples and four-vertex configurations imposes equations on how many local profiles can occur. We retain these equations and nonnegativity, but allow real weights and do not require them to arise from a graph. This enlarged system is called a *linear relaxation*. Every graph gives a solution of the system, although some solutions may have no corresponding graph. A bound valid for every solution therefore holds for every graph. Section II.9 constructs this assignment explicitly.

To obtain a polygon, we prove lower and upper bounds on eight linear combinations of t and q . The coefficient pairs are

$$(1, 0), (0, 1), (1, 1), (1, -1), (1, 2), (2, 1), (1, -2), (2, -1).$$

For a direction (a, b) , the quantity $at + bq$ is integral. If an exact rational certificate shows $at + bq > k - 1$, where k is an integer, then $at + bq \geq k$. Upper bounds are treated by changing the sign. Intersecting these integer half-planes produces an outer polygon. For example, $(a, b) = (1, 0)$ bounds the number of triangles, whereas $(a, b) = (1, -1)$ bounds $t - q$. Every graph profile satisfies all these bounds and therefore lies in their half-plane intersection.

Each defining half-plane is justified by an exact rational certificate. The polygon audit then reconstructs their intersection over the rational numbers and compares its vertices with the stored vertex list. These checks establish both that every graph satisfies the defining inequalities and that the stored vertices describe their full intersection. The files also contain `raw` numerical estimates used to select the integer cutoffs; verification uses the exact certificates at those cutoffs.

Across orders 19–23, the full dependency chain uses 1968 exact certificates for the defining half-planes. Twenty-six additional edge classes use complete extreme-edge catalogues. These certificates form the shared set of prerequisites for the local bounds used in the complete proof. Their reconstruction and verification are specified in the companion mathematical specification.

The verified minimum and maximum edge counts for $(4, 5)$ graphs of orders 19–24 are, respectively,

$$[57, 92], [68, 100], [77, 107], [88, 114], [101, 122], [116, 132].$$

Their published source is Table 1 in Appendix A of [II.2]. They delimit the edge classes that must be covered. The coverage audit checks that each edge count in these intervals has either a complete catalogue or a certified containing polygon.

II.9 Profile variables and the saved equation multipliers

The file `edge_46_38_max.json` stores the same numerical bound as (II.16), but in coordinates adapted to a general counting model. This section identifies those coordinates and all nonzero multipliers, making the relation explicit.

For $X \in \mathcal{R}(4, 6, 38)$, the numbers of vertices with each possible neighborhood profile sum to 38, and their degree-weighted sum is $2e(X)$. To express these counts using the available containing polygons, write each actual profile as a convex combination of polygon vertices. Aggregating those coefficients over the vertices of X gives nonnegative profile weights. They preserve the two sums just described and all other statistics that are affine in a local profile.

Put $D = \{13, 14, 15, 16, 17\}$. The polygon vertices in Section II.4 determine finite index sets

$$\begin{aligned}\mathcal{I}_A &= \{(d, \eta, \tau, \kappa) : d \in D, \quad \eta \in \{0, \dots, \binom{d}{2}\}, \quad (\tau, \kappa) \in V_{3,6,d,\eta}\}, \\ \mathcal{I}_B &= \{(h, \eta, \tau, \kappa) : h = 37 - d, \quad d \in D, \quad \eta \in \{0, \dots, \binom{h}{2}\}, \quad (\tau, \kappa) \in V_{4,5,h,\eta}\}.\end{aligned}$$

There is one nonnegative real variable $a_{d,\eta,\tau,\kappa}$ per index in $\mathcal{I}_A$ and one $b_{h,\eta,\tau,\kappa}$ per index in $\mathcal{I}_B$. For brevity put $p = (\eta, \tau, \kappa)$ and write $a_{d,p}$ and $b_{h,p}$ for these same variables. Thus the first index always denotes the order of the local graph: d for a neighborhood and $h = 37 - d$ for a nonneighborhood. Each sum over p below is restricted to the indices belonging to its indicated order and side.

Here is the assignment supplied by an actual $X \in \mathcal{R}(4, 6, 38)$. For each v of degree d , choose nonnegative coefficients $\lambda_{v,p}^A$ with sum one, supported only on $p = (e(A_v), \tau, \kappa)$ with $(\tau, \kappa) \in V_{3,6,d,e(A_v)}$, such that

$$(t(A_v), q(A_v)) = \sum_p \lambda_{v,p}^A (\tau, \kappa).$$

They exist because the actual pair lies in that polygon. Similarly choose nonnegative coefficients $\lambda_{v,p}^B$ with sum one, supported on $p = (e(B_v), \tau, \kappa)$ with $(\tau, \kappa) \in V_{4,5,37-d,e(B_v)}$, whose convex combination of (τ, κ) equals $(t(B_v), q(B_v))$. Define

$$a_{d,p} = \sum_{v:d_v=d} \lambda_{v,p}^A, \quad b_{h,p} = \sum_{v:d_v=37-h} \lambda_{v,p}^B,$$

where missing coefficients are zero. These variables are aggregate convex-combination weights; they do not count vertices whose local graph has the polygon-vertex profile p . They can be fractional and their representation can be nonunique, even for one fixed graph. Nevertheless,

$$m_d := \sum_p a_{d,p} = \sum_p b_{37-d,p} = \#\{v : d_v = d\}, \quad \sum_{d,p} a_{d,p} = 38.$$

The symbol m_d denotes this common degree-class weight; only for an assignment supplied by a graph must it be an integer.

For a concrete example, the catalogue `r36_13.g6.gz.profiles` has exactly three profiles with 16 edges, whose (t, q) coordinates are

$$(0, 25), \quad (0, 26), \quad (0, 28).$$

Their convex hull is the segment with endpoints $(0, 25)$ and $(0, 28)$. If a vertex of X has neighborhood profile $(13, 16, 0, 26)$, then

$$(0, 26) = \frac{2}{3}(0, 25) + \frac{1}{3}(0, 28)$$

assigns weight $2/3$ to $a_{13,16,0,25}$ and weight $1/3$ to $a_{13,16,0,28}$. These contributions total one vertex, preserve its 16 local edges and zero local triangles, and give weighted q value $(2/3)25 + (1/3)28 = 26$.

The example uses a profile in the smaller catalogue and shows how any occurrence of that profile would be represented. The aggregated variables are consequently allowed to be fractional, even when they arise from a graph.

The linear functions of the profile weights representing ambient statistics are

$$\mathcal{E} = \frac{1}{2} \sum_{d,p} d a_{d,p}, \quad \mathcal{T} = \frac{1}{3} \sum_{d,p} \eta a_{d,p}, \quad \mathcal{W} = \sum_{d,p} \binom{d}{2} a_{d,p}, \quad \mathcal{Q} = \mathcal{W} - \mathcal{T}.$$

On the graph assignment these equal $e(X), t(X), w(X), q(X)$. Likewise a local expression affine in (η, τ, κ) at fixed order is evaluated by its weighted sum. For example, $\sum_v T_k(A_v)$ means $\sum_{d,p} T_k(d,p) a_{d,p}$, where $T_k(d,p)$ is obtained from (II.2) by substituting $(n, e, t, q) = (d, \eta, \tau, \kappa)$; the analogous expression for B uses (h, p) and $b_{h,p}$. Convex combinations preserve these affine expressions, so these weighted sums equal the graph counts whenever the variables come from a graph.

Add a variable $z_F \geq 0$ for each of the eleven unlabelled four-vertex graph types F , and $j_{d,f} \geq 0$ for each $d, f \in D$ with $d \leq f$. For a graph assignment, z_F counts four-subsets inducing F and $j_{d,f}$ counts unordered edges with endpoint degrees d, f . The relaxation permits real values but imposes the counting equations. One coordinate is known to be zero: $z_{K_4} = 0$, because the ambient graph $X \in \mathcal{R}(4, 6, 38)$ contains no complete four-vertex subgraph. Counting the ends of edges incident with degree- d vertices gives one equation for each degree class:

$$2j_{d,d} + \sum_{f < d} j_{f,d} + \sum_{f > d} j_{d,f} = d m_d \quad (d \in D).$$

Counting unordered paths on four distinct vertices by their central edge gives another equation; the paths are not required to be induced:

$$\sum_{d \leq f} (d-1)(f-1)j_{d,f} - 3\mathcal{T} = \sum_F \text{path}_4(F) z_F,$$

where $\text{path}_4(F)$ is the number of such paths in F . All other rows appear in Section II.10.

The following elementary example illustrates how a weighted sum of equations proves a bound. Let u_1, u_2, u_3 be nonnegative numbers satisfying

$$u_1 + u_2 = 5, \quad u_2 + u_3 = 4.$$

Multiply the first equation by 2 and the second by 1, then add:

$$2u_1 + 3u_2 + u_3 = 14.$$

To bound the slightly different expression $3u_1 + 3u_2 + u_3$, write

$$3u_1 + 3u_2 + u_3 = 14 + u_1 \geq 14.$$

The equation multipliers $(2, 1)$ supply a certificate for the lower bound 14: subtracting the weighted equations from the desired expression leaves the nonnegative quantity u_1 . The same calculation applies to every nonnegative solution of the two equations. The local graph certificate uses this operation with its profile weights and counting equations.

There are 107 neighborhood profile variables, 1084 nonneighborhood profile variables, 15 degree-pair variables, and 11 four-vertex variables. Let $\mathbf{u} = (u_0, \dots, u_{1216})^T$ list them in the order used by the supplied program. There are 25 counting equations. Write the i th equation as

$$\sum_{j=0}^{1216} m_{ij} u_j = b_i \quad (0 \leq i < 25).$$

The matrix $M = (m_{ij})$ is the rectangular table of their coefficients, and $\mathbf{b} = (b_0, \dots, b_{24})^T$ is the vector of right-hand sides. Thus $M\mathbf{u} = \mathbf{b}$ abbreviates these 25 equations; $\mathbf{u} \geq 0$ abbreviates the 1217 nonnegativity conditions. The additional condition $z_{K_4} = 0$ is retained.

We seek an upper bound on $e(X)$, so we seek a lower bound on its negative, $-\mathcal{E}$. Define c_j to be the coefficient of u_j in this linear expression; then $-\mathcal{E} = \sum_j c_j u_j$. Choose rational multipliers $y_0, \dots, y_{24}$ for the counting equations. Their weighted sum is

$$\sum_j \left(\sum_i y_i m_{ij} \right) u_j = \sum_i y_i b_i.$$

Subtracting this from $\sum_j c_j u_j$ leaves coefficient $c_j - \sum_i y_i m_{ij}$ on u_j . If each such coefficient is nonnegative, the target is at least the constant $\sum_i y_i b_i$. The sign of the coefficient on z_{K_4} is immaterial because that coordinate is zero. It is therefore sufficient to check the signs on all remaining coordinates.

With $\mathbf{c} = (c_j)^T$, $\mathbf{y} = (y_i)^T$, and superscript T denoting transpose, the preceding calculation can be abbreviated as

$$\mathbf{c}^T \mathbf{u} = \mathbf{y}^T \mathbf{b} + (\mathbf{c} - M^T \mathbf{y})^T \mathbf{u}. \quad (\text{II.17})$$

The vector $\mathbf{c} - M^T \mathbf{y}$ is called the *residual*. The checker verifies its coordinate signs and the exact value of $\mathbf{y}^T \mathbf{b}$. It also reconstructs the equations and the objective so that the calculation is tied to the stated graph problem. The list of equation multipliers $\mathbf{y}$ is stored in the JSON field `dua1`. In the following row table, e, t, w abbreviate $\mathcal{E}, \mathcal{T}, \mathcal{W}$; graph-style sums of local counts abbreviate the affine weighted sums just defined.

The certificate contains 25 equation multipliers, 17 of them nonzero. The degree-pair equations and the path equation have zero multiplier in this example. Thus they contribute nothing to this particular weighted sum, although they are useful for other certificates.

After the weighted sum is expanded, the coefficient of $a_{d,p}$ is

$$\frac{\alpha_d + (3d - 38)\eta - 3\kappa}{166} - \frac{4101}{220780} \tau.$$

All the neighborhood polygon vertices come from triangle-free catalogues, so $\tau = 0$. The coefficient of $b_{37-d,p}$ is $(K_d - 3\eta + 3\tau)/166$. Every degree-pair coefficient is zero. The four-vertex coefficients are zero except that of K_4 , which is $4101/55195$ and multiplies a zero variable. Equation (II.17) is therefore precisely (II.16) divided by 166.

II.10 The multipliers and their normalization

For completeness, the entire multiplier vector is tabulated below in the exact row order used by `recursive_profiles_v4.py`. Here $U_k(F)$ is the number of universal vertices of F whose deletion leaves a triple with k edges; $Z_k(F)$ is the corresponding number of isolated vertices (vertices of degree zero). Quantities such as e, t, w in the equations are linear functions of the profile weights.

The rows arise from the following incidence counts. Each four-subset has one induced type, giving row 7. Each edge belongs to $\binom{36}{2}$ four-subsets, and each triangle or pair of incident edges belongs to 35 four-subsets, giving rows 8–10. For row 11, a triple in A_v together with v is a four-set in which v is universal. Counting these pairs first by v gives the left sum, and first by the induced four-vertex type gives the right sum. The same argument applies to each triple edge count and, for B_v , to an isolated fourth vertex. Each row therefore holds for every graph in the ambient family.

Row	Equation, written with the displayed left side	Multiplier
0	$\sum a = 38$	$-6902/1577$
1	$\sum_p a_{13,p} - \sum_p b_{24,p} = 0$	$5014/55195$
2	Matching weights at degree 14	$-11/380$
3	Matching weights at degree 15	$47/3154$
4	Matching weights at degree 16	0
5	Matching weights at degree 17	$66/581$
6	Twice the left side of (II.6), with $n = 38$	$3/332$
7	$\sum_F z_F = \binom{38}{4}$	$-111/55195$
8	$2 \sum_F e(F) z_F - 2 \binom{36}{2} e = 0$	$111/220780$
9	$3 \sum_F t(F) z_F - 3 \cdot 35 t = 0$	$-1/166$
10	$\sum_F w(F) z_F - 35 w = 0$	$-111/220780$
11	$\sum_v T_0(A_v) - \sum_F U_0(F) z_F = 0$	$-111/220780$
12	Same equation with T_1, U_1	$-4101/220780$
13	Same equation with T_2, U_2	$-4101/220780$
14	Same equation with T_3, U_3	0
15	$\sum_v T_0(B_v) - \sum_F Z_0(F) z_F = 0$	$-111/220780$
16	Same equation with T_1, Z_1	$-111/220780$
17	Same equation with T_2, Z_2	$-111/220780$
18	Same equation with T_3, Z_3	$-4101/220780$
19–23	Degree-stub equations for $d = 13, \dots, 17$	0
24	Four-vertex path identity	0

Row 6 uses the program’s normalization: twice the left side of equation (II.6); its multiplier $3/332$ therefore has the same effect as multiplying the simplified equation by $3/166$. Rows 8 and 9 similarly retain the program’s factors 2 and 3, so that the multipliers in this table can be compared directly with the JSON. Only rows 0 and 7 have nonzero right sides. Their contribution is

$$-\frac{6902}{1577} 38 - \frac{111}{55195} \binom{38}{4} = -\frac{26125}{83}.$$

This is the field `exact_lower_bound` in the JSON file. Because the objective is $-e$, a lower bound on the objective is an upper bound on e . The field `threshold` is -315 : the certified objective is strictly greater than that threshold, which gives the integral conclusion $e \leq 314$.

The multiplier signs are unrestricted because the multiplied rows are equalities. The signs of the resulting residual coefficients, not the signs of these equality multipliers, justify the inequality. This distinction is useful when inspecting a certificate with many negative entries.

II.11 What the remaining local computations add

The same mechanism proves the other endpoint bounds used in the main paper. For a $(4, 6)$ -graph, the smaller inputs are again $(3, 6)$ neighborhoods and $(4, 5)$ nonneighborhoods. For a $(5, 5)$ -graph, they are $(4, 5)$ neighborhoods and $(5, 4)$ nonneighborhoods. Complementation exchanges edges and nonedges and turns a $(5, 4)$ -graph into a $(4, 5)$ -graph of the same order. Thus the same $(4, 5)$ input catalogues and polygons provide bounds for both sides.

To see how the counts are obtained, write $\bar{Y}$ for the complement of a graph Y on m vertices. From the triple counts in (II.2),

$$\begin{aligned} e(\bar{Y}) &= \binom{m}{2} - e(Y), \\ t(\bar{Y}) &= \binom{m}{3} - (m-2)e(Y) + q(Y), \\ q(\bar{Y}) &= 2\binom{m}{3} - (m-2)e(Y) + t(Y). \end{aligned}$$

The second line is $T_0(Y)$, and the third is $T_1(Y) + 2T_0(Y)$. These are affine transformations of the profiles, so they also preserve the validity of containing polygons under complementation.

Some endpoint bounds have an immediate integrality step. For example,

$$e(Y) \geq \frac{95414}{221} > 431 \implies e(Y) \geq 432 \quad (Y \in \mathcal{R}(5, 5, 44)).$$

The corresponding file is `edge_55_44_min.json`. For order 45, `edge_55_45_min.json` gives

$$e(Y) \geq \frac{12137627230191}{26402663302} > 459, \quad \text{hence } e(Y) \geq 460.$$

These bounds are obtained by checking the rational equation multipliers in the two files using exact arithmetic.

The order 39 $(4, 6)$ bound has a useful extra step. The initial certificate proves $e(Y) \leq 325$. Fixing $e(Y) = 325$, two further certificates give

$$t(Y) \geq \frac{1612}{3}, \quad -t(Y) \geq -\frac{1612}{3}.$$

Thus $t(Y) = 1612/3$, impossible for an integer triangle count. This excludes the endpoint and yields $e(Y) \leq 324$. The files `exclude_4639e325_t_gt537.json` and `exclude_4639e325_t_lt538.json` encode the two sides. Their filenames refer to the neighboring integer thresholds; the exact certified value is the rational number displayed above.

To describe the five remaining local inequalities, return to a hypothetical $X \in \mathcal{R}(5, 6, 84)$ and take A_v, B_v to be its neighborhood and nonneighborhood graphs. The inequalities have the form

$$\gamma_d d + \gamma_e e(Y) + \gamma_t t(Y) + \gamma_q q(Y) + \gamma_0 \leq 0,$$

for $Y = A_v$ or B_v in the 84-vertex graph, where $d = d_X(v)$ and $(\gamma_d, \gamma_e, \gamma_t, \gamma_q, \gamma_0)$ is the fixed coefficient vector of the chosen inequality. The five vectors are recorded in `exact_candidate_used_facets.json`. The identifiers 203, 595, 603, 658 and 681 are lookup keys for these five stored inequalities; they identify entries in that file. Each inequality is proved in every possible edge class of its local family by a weighted sum of counting equations. The inputs are the smaller profile sets already described. In particular, the proof requires no catalogue of all 38-, 39-, 44- or 45-vertex graphs. For example, identifier 203 means

$$770d - 269e(Y) - 6t(Y) + 12q(Y) + 5940 \leq 0 \quad (Y = A_v).$$

At $d = 38$ and $e(Y) = 314$, the statement becomes $-6t(Y) + 12q(Y) \leq 49266$. The file `facet203_46_n38_e314.json` supplies equation multipliers proving

$$6t(Y) - 12q(Y) \geq -\frac{541956}{11} > -49272.$$

Since $-6t(Y) + 12q(Y)$ is a multiple of 6, this proves the required bound 49266. It proves it for every graph in the edge class, by the same graph-to-equations implication as in the worked example.

For identifier 203, the edge classes are $267, \dots, 314$ at order 38 and $288, \dots, 324$ at order 39: 85 cases. Each of the other four inequalities concerns nonneighborhoods, with edge classes $432, \dots, 514$ at order 44 and $460, \dots, 530$ at order 45: 154 cases per inequality. Thus there are $85 + 4 \cdot 154 = 701$ checks of rational equation multipliers. The endpoint bounds justify excluding all other edge counts. The verifier reconstructs this entire case list so that an absent case cannot silently disappear from the proof.

The complete local dependency chain consists of 1968 prerequisite half-plane certificates, ten endpoint and exclusion certificates, and 701 certificates for these five further inequalities: 2679 in total. The verifier `replay_required_local_certificates.py` checks that entire chain, including the list of required cases and the arithmetic of each certificate.

II.12 Why the global proof uses seven-vertex graphs

Suppose again that $X \in \mathcal{R}(5, 6, 84)$ exists. Every seven-vertex induced subgraph belongs to one of 979 allowed isomorphism types: those with neither a K_5 nor an independent six-set. Let $\mathcal{H}$ be this finite set and put

$$p_H = \frac{\#\{W \subseteq V(X) : |W| = 7, X[W] \cong H\}}{\binom{84}{7}}.$$

Then $p_H \geq 0$ and $\sum_H p_H = 1$. Equivalently, p_H is the probability that a uniformly chosen seven-subset induces type H .

A rooted graph has specified distinguished vertices, called roots; an isomorphism of rooted graphs must fix each specified root label. A flag specifies the induced rooted isomorphism type on a chosen set containing the roots. Its indicator is 1 if the chosen vertices have that type and 0 otherwise. Thus multiplying an inequality by a flag indicator restricts the count to those root configurations.

Many local counting expressions can be expanded as linear functions of the p_H . For instance, one can multiply a valid inequality by the indicator of a prescribed rooted graph and average over roots. The multiplier is nonnegative, so the averaged inequality is valid. Lifting the five local inequalities in this way retains information about how neighborhoods sit inside the ambient graph.

For example, fix a six-vertex flag with a distinguished root v , and let r_0 be the number of its other five vertices adjacent to v . There are 78 vertices outside this flag. Choose one uniformly, and let χ be 1 if it is adjacent to v , and 0 otherwise. The average of $r_0 + 78\chi$ is exactly d_v , because the chance that $\chi = 1$ is $(d_v - r_0)/78$. The quantity $r_0 + 78\chi$ can be read from the induced seven-vertex graph, although d_v cannot. Thus the universally valid bound $d_v \leq 39$ produces an inequality on averages of seven-vertex statistics. This replacement by a statistic with the same average is what the general count conversion does.

The normalization follows by comparing counts in a seven-set with counts in the full graph. If an expression uses m vertices outside a fixed flag on f vertices, where $0 \leq m \leq 7 - f$, a seven-set contains only $7 - f$ available positions while the full graph has $84 - f$. The corresponding count conversion includes the exact factor

$$\frac{\binom{84-f}{m}}{\binom{7-f}{m}}.$$

When two extension sets are counted together, the conversion is performed separately for each possible size of their intersection. Including these overlap cases makes the conversion an exact identity for a graph on 84 vertices.

The linear rows include the degree restriction $d_v \in \{38, 39\}$, capacities of common neighborhoods, local edge and profile bounds, and elementary constraints on integer cell sizes. Here a mixed cell is

$$\mathcal{C}(C, D) = \{x \in V(X) \setminus (C \cup D) : C \subseteq N_X(x), \quad D \cap N_X(x) = \emptyset\},$$

where the specified root sets C, D are disjoint, C is a clique, and D is independent. The cell size is $c = |\mathcal{C}(C, D)|$. For any integer c and integer k ,

$$(c - k)(c - k - 1) \geq 0.$$

This inequality holds for each choice of roots and hence also after averaging over root configurations. Other rows use the fact that $d_v - 38$ and $39 - d_v$ take only the values zero and one. For example, when C is the two endpoints of an edge and D is empty, the cell is their common neighborhood. A triangle in that cell would form a K_5 with the two roots; an independent six-set there would

also be forbidden in X . It is therefore a $(3,6)$ -graph, so its size is at most 17. The general definition treats mixed adjacency and nonadjacency requirements in the same explicit way.

The 979-type list is itself audited: all 2^{21} labelled graphs on seven vertices are enumerated, forbidden patterns are excluded, and counts are compared with the orbit sizes of the representatives. This verifies a finite enumeration used internally by the proof. It is logically separate from the completeness of the external Ramsey graph catalogues.

II.13 The positive-semidefinite inequalities are sums of squares

The final certificate contains inequalities that are most compactly expressed with positive-semidefinite matrices. Their validity comes from an elementary square-counting observation.

Fix a labelled root graph σ on $\{1, \dots, \ell\}$ and an integer $k \geq 0$ with $\ell + 2k \leq 7$. A root tuple is an injection $\rho : \{1, \dots, \ell\} \rightarrow V(X)$ inducing σ . Let $\mathcal{F}_{\sigma,k}$ be the finite set of rooted isomorphism types on $\ell + k$ vertices whose labelled roots induce σ and which contain neither a K_5 nor an independent six-set. Isomorphisms fix the root labels and may permute the k nonroot vertices. For each $i \in \mathcal{F}_{\sigma,k}$ define

$$f_i(\rho) = \#\{S \subseteq V(X) \setminus \text{im } \rho : |S| = k, (X[\text{im } \rho \cup S], \rho) \text{ has type } i\}.$$

Write $f(\rho)$ for the column vector with these coordinates. For any real vector $z = (z_i)_{i \in \mathcal{F}_{\sigma,k}}$,

$$\sum_{\rho} (z^T f(\rho))^2 \geq 0.$$

The sum runs over all root injections of type σ , including the unique empty injection when $\ell = 0$. Thus $\sum_{\rho} f(\rho)f(\rho)^T$ is positive semidefinite. Expanding a square counts ordered pairs of extension sets, including equal or overlapping sets. For instance, choosing coefficients 1 and -2 on two extension types gives $(f_1(\rho) - 2f_2(\rho))^2 \geq 0$. Its cross term $-4f_1(\rho)f_2(\rho)$ counts choices of one extension of each type with the indicated coefficient. The full matrix records all such products at once. Positive semidefinite means that every choice of real coefficients yields a nonnegative quadratic expression; here that property is proved by the displayed squares.

For each $H \in \mathcal{H}$, define a matrix M_H with rows and columns indexed by $\mathcal{F}_{\sigma,k}$. Its (i, j) entry is a sum over root injections ρ into H and ordered pairs of k -subsets $S, T \subseteq V(H) \setminus \text{im } \rho$ having types i, j . Give each such configuration weight

$$\frac{1}{\binom{84-\ell-|S \cup T|}{7-\ell-|S \cup T|}}.$$

Its denominator is the number of seven-subsets of $V(X)$ containing that configuration's roots and extension vertices. Thus summing over seven-sets counts each full-graph configuration exactly once:

$$\sum_{\substack{W \subseteq V(X) \\ |W|=7}} M_{X[W]} = \sum_{\rho} f(\rho)f(\rho)^T, \quad \sum_{H \in \mathcal{H}} p_H M_H \succeq 0.$$

The latter matrix is the former divided by $\binom{84}{7}$. All pairs (S, T) , including $S = T$, enter the definition of M_H .

For each fixed pair (σ, k) , choose a positive common denominator L and put $I_H = LM_H$ for every $H \in \mathcal{H}$. The same positive factor throughout the block makes every I_H integral and preserves the sign. Every supplied integer vector z gives the valid row

$$\sum_{H \in \mathcal{H}} p_H (-z^T I_H z) \leq 0.$$

The final sum uses 194 such rows, drawn from eight blocks with $(\ell, k) = (0, 3), (1, 3), (2, 2), (3, 2)$: respectively one, one, two, and four root isomorphism types. Each root type has a fixed labelling; all injections of that labelled type are counted. The entries can be recomputed by integer arithmetic, and the displayed sum of squares proves the sign.

The accompanying files `rational_psd_vectors.json` and `moment_integer_verified.npz` specify the vectors and integer matrices. The array `rational_psd_numerators.npy` records their contractions. The independent verifiers recompute these contractions and compare them with the saved array.

II.14 The final exact separation

For $0 \leq i < 5324$, let $U_i : \mathcal{H} \rightarrow \mathbb{Q}$ be an inequality row: 4876 linear counting rows and 448 square rows. For $0 \leq j < 89$, let $E_j : \mathcal{H} \rightarrow \mathbb{Q}$ be a homogeneous equality row. An integer row $(a_H)_{H \in \mathcal{H}}$ is normalized by division by $d = \max\{1, \max_H |a_H|\}$. The normalization equation $\sum_H p_H = 1$ is separate and is not homogeneous. Validity means

$$\sum_H p_H U_i(H) \leq 0, \quad \sum_H p_H E_j(H) = 0.$$

An inequality row is therefore a list of 979 rational coefficients, one per seven-vertex type. Its validity concerns the weighted average of those coefficients with the frequencies supplied by an actual 84-vertex graph; individual coefficients can have either sign. The equality rows have zero right-hand side, which is the meaning of “homogeneous” here. The final file `exact_separation_certificate.json` supplies nonnegative rational weights λ_i and arbitrary rational weights μ_j . It uses 816 inequality rows, of which 194 are the square rows just discussed, and 87 equality rows. Thus there are 903 nonzero weights.

For every allowed seven-vertex type, define

$$C(H) = \sum_i \lambda_i U_i(H) + \sum_j \mu_j E_j(H).$$

The exact computation establishes

$$C(H) \geq \delta := \frac{3146981978575629437}{10^{24}} > 0 \quad (H \in \mathcal{H}). \quad (\text{II.18})$$

Once validity of the rows and (II.18) are established, the contradiction itself is short:

$$0 \geq \sum_H p_H C(H) \geq \delta \sum_H p_H = \delta > 0.$$

There can therefore be no graph $X \in \mathcal{R}(5, 6, 84)$.

The stored weights have denominator 10^{12} . Each row also has its own exact normalization denominator. To avoid a potentially large least common multiple, the verifier bounds each summand downward on a common grid of denominator 10^{24} . If a summand is $wa/(10^{12}d)$, where w is the signed integer numerator of its multiplier, a is the integer row coefficient at the chosen H , and $d > 0$ is that row’s normalization denominator, it contributes the lower bound

$$\frac{1}{10^{24}} \left\lfloor \frac{wa 10^{12}}{d} \right\rfloor.$$

This formula is valid for negative summands as well as positive ones. Integer floor division must be used, rather than truncation toward zero. Summing the floors gives a lower bound, so a strictly positive resulting numerator proves (II.18) without any floating-point tolerance.

Taking the minimum of these computed lower bounds over the 979 types gives the reported numerator. Thus δ is a common strictly positive lower bound for all coefficients $C(H)$. The use of integer arithmetic establishes its positivity exactly.

II.15 A practical verification route

A reader can check the local argument and then replay the complete computation. The two tasks have different conclusions. This section gives the commands in their execution order; the names of internal programs below explain what those commands invoke.

Prepare one verification directory

The accompanying file `verification.tar.xz` contains a directory named `verification`. In a terminal whose current directory contains that archive, run

```
tar -xJf verification.tar.xz
cd verification
```

The commands below are run from this extracted directory. It contains `verify_package.py`, `check_local_example.py`, `README.txt` and the subdirectories holding the data. If those files have already been extracted, open a terminal in their `verification` directory instead of extracting another copy. The complete archive includes all raw catalogue inputs. In the explicitly labelled compact archive, `START_HERE.txt` explains how to restore the 58 omitted catalogue files with their required hashes; that restoration must precede the complete replay.

Check the local example

First read the definitions of e, t, q, A_v, B_v , the ten inequalities in Section II.4, and the cancellation in Section II.6. The exact paper calculation to check is

$$\sum_v (L_A(v) + L_B(v)) = 52250 - 166e(X).$$

To check its finite profile substitutions, run

```
python3 check_local_example.py --root .
```

Here the dot denotes the extracted `verification` directory, which is the terminal's current directory. This command uses only the Python standard library. Its output must contain

```
"worked_example_passed": true
"edge_upper_bound": 314
```

It also prints the exact minima listed in Section II.7. A successful run checks those substitutions, the constant identity, and the correspondence with the saved equation multipliers. It does not yet verify the proofs of all input polygon bounds or the final 84-vertex contradiction. The arithmetic uses exact fractions. Decimal polygon coordinates in the stored files are recovered as small rationals, with reduced denominators 1, 2 or 3, and checked against their decimal encodings. The separate exact polygon audit reconstructs the vertices from their defining half-planes.

Replay the whole computation

The tested interpreter is Python 3.13.9. The package pins NumPy, SciPy and NetworkX versions in its requirements file. With Python 3.13 installed, create an isolated environment and install them:

```
python3.13 -m venv .venv
. .venv/bin/activate
python -m pip install -r proof_84/requirements.txt
```

These are terminal commands for macOS or Linux. The package README also gives Windows instructions. Installation requires access to a package index or previously downloaded packages. The verification commands themselves run without network access. The full extraction check additionally needs a C++11 compiler and zlib headers and library; the README gives the platform-specific installation commands.

Once these dependencies are installed, run

```
python verify_package.py --full
```

Run Python normally, without `-O`, `-OO`, or `PYTHONOPTIMIZE`: some proof checks are assertions. Every invoked stage must finish successfully, and the final summary must contain `all_passed: true`. If a stage fails or the command stops before the final summary, the full replay has not succeeded. The launcher writes reports and identifies the stage that failed.

This command invokes the 18-check suite and all supplementary checks. Together they verify the following parts of the argument:

1. The 58 supplied catalogue files produce the claimed distinct profiles. The supplied C++ extractor is rerun exhaustively; independent sampled calculations also check profile extraction. Neither operation proves the external catalogue completeness theorem anew.
2. The recursive polygons contain the intended profile possibilities, and all required degrees and edge classes are covered. The 2679 local certificates have the stated exact multiplier sums and residual signs.
3. The 979 allowed seven-vertex types form a complete list, the used coefficient rows agree with their counting definitions, and the square rows come from the finite square-counting construction.
4. The saved final multipliers produce a strictly positive coefficient on every allowed seven-vertex type, with the lower bound in (II.18).

The principal internal local checker is `replay_required_local_certificates.py`; the two final replay implementations are `replay_global_certificate.py` and `audit_global_exact_certificate.py`. The supplied file index locates these basenames within the preserved archive structure.

What remains mathematical input and review

The initial Ramsey bounds and completeness of the cited catalogues remain external mathematical inputs. In particular, $R(4, 6) \leq 40$ is recorded in the survey [II.4] as a result of Angelteit and McKay, attributed there to personal communications. The package does not supply a new proof of that bound. A reader should also check that the displayed counting identities apply to the stated families and that convex weights preserve the expressions used. The companion specification provides the complete definitions for reconstructing all remaining row families. Software success confirms the specified finite arithmetic; the preceding explanations identify why that arithmetic is relevant to the theorem.

II.16 Mathematical review and finite verification

The proof has a mathematical part and a finite arithmetic part. The mathematical part establishes that every graph in the specified Ramsey family yields variables satisfying the stated conditions. This requires the catalogue completeness results, the graph-to-weight construction, the counting identities, the treatment of root vertices, and the finite sampling factors. The arithmetic part checks the required profile cases and the supplied weighted sums using exact rational or integer operations. Both parts are needed to infer the theorem from a successful verification run.

For the worked example, the identities in Sections II.5 and II.6 can be checked on paper. The ten local inequalities have their coefficients printed explicitly; their validity is checked on the finite profile sets in Section II.7. The same method extends to the complete proof, whose 2679 local certificates and 903 nonzero final weights are supplied in machine-readable files. The companion mathematical specification defines every row family and gives the procedures for reconstructing and checking those files.

The file hashes identify the precise inputs used in a verification run. The mathematical meaning of those inputs comes from the constructions above, while the exact checkers establish their claimed arithmetic properties. A reproducible verification therefore records both the input hashes and the successful outputs of the relevant checks.

The worked local certificate also illustrates a useful simplification: its 1217-variable representation reduces, after collecting terms and applying the triangle-incidence identity, to a sum of two nonnegative expressions per vertex. This explicit reduction connects the stored coefficients with the graph-counting proof of the 314-edge bound.

References

- [II.1] B. D. McKay and S. P. Radziszowski, *Subgraph counting identities and Ramsey numbers*, Journal of Combinatorial Theory, Series B **69** (1997), 193–209. <https://users.cecs.anu.edu.au/~bdm/papers/r55.pdf>
- [II.2] V. Angelteit and B. D. McKay, $R(5, 5) \leq 46$, Journal of Graph Theory **112** (2026), 198–208. <https://doi.org/10.1002/jgt.70029>; <https://arxiv.org/html/2409.15709v2>
- [II.3] B. D. McKay, *Ramsey Graphs*, complete graph catalogues. <https://users.cecs.anu.edu.au/~bdm/data/ramsey.html>
- [II.4] S. P. Radziszowski, *Small Ramsey Numbers*, Electronic Journal of Combinatorics, Dynamic Survey DS1, revision 18, 24 April 2026, Table Ib. <https://www.cs.rit.edu/~spr/ElJC/ejcram18.pdf>

Part III

Detailed mathematical specification of the certificate for

$$R(5, 6) \leq 84$$

Verification recipes and representative calculations

Supplement to the accompanying manuscript

29 October 2026

Abstract

This supplement specifies the finite calculations in the accompanying proof that every graph on 84 vertices contains a clique of order 5 or an independent set of order 6. Assuming a counterexample, we count configurations in the graph and derive equations and inequalities that these counts must satisfy. Explicit rational weights combine these constraints into a contradiction. The weights on inequalities are nonnegative, so adding the weighted inequalities preserves their direction; equations may be multiplied by signed weights. The verification checks these combinations using exact arithmetic. This document defines the quantities being counted, derives the constraints, gives representative calculations, and provides the commands and expected outputs for checking the supplied numerical files. Published Ramsey bounds and the completeness of the cited small-graph catalogues are stated external inputs.

III.1 Notation and external inputs

All graphs are finite, undirected, simple graphs. For a graph X , write $V(X)$ and $E(X)$ for its vertex and edge sets, $|X| = |V(X)|$ for its order, $N_X(v) = \{u \in V(X) : \{u, v\} \in E(X)\}$ for a vertex neighborhood, and $d_X(v) = |N_X(v)|$ for its degree. For $S \subseteq V(X)$, $X[S]$ is the induced subgraph on S . The complement $\bar{X}$ has the same vertices and exactly the missing pairs as edges. The clique number $\omega(X)$ and independence number $\alpha(X)$ are the largest orders of a complete and an edgeless induced subgraph, respectively. Subscripts on N and d may be omitted when the ambient graph is specified.

For positive integers s, r and an integer $n \geq 0$, let

$$\mathcal{R}(s, r, n) = \{X : |X| = n, \omega(X) < s, \alpha(X) < r\}.$$

Graphs in this family are considered up to isomorphism whenever they are enumerated. The Ramsey number $R(s, r)$ is the least n for which the family is empty. Define the following graph statistics:

$$\begin{aligned} e(X) &= |E(X)|, \\ t(X) &= |\{S \subseteq V(X) : |S| = 3, X[S] \cong K_3\}|, \\ w(X) &= \sum_{v \in V(X)} \binom{d_X(v)}{2}, & q(X) &= w(X) - t(X). \end{aligned}$$

Thus w counts a center and an unordered pair of its neighbors, including pairs that are adjacent. An induced three-vertex path contributes 1 to q and a triangle contributes 2; all other three-vertex induced types contribute zero. The profile of X is the four-tuple $(|X|, e(X), t(X), q(X))$. Induced-copy counts count vertex subsets, not labelled embeddings. Other counts, such as paths that need not be induced, are specified separately.

The symbols K_m, P_m, C_m denote the complete graph, path, and cycle on m vertices; $K_{1,3}$ is the three-leaf star, $+$ denotes disjoint union, and $K_4 - e$ means K_4 with one edge removed. We use $\mathbf{1}_{\mathcal{A}} = 1$ if a condition $\mathcal{A}$ holds and 0 otherwise. Empty sums are zero, empty products are one, and $\binom{S}{k}$ denotes the set of k -element subsets of a finite set S . For nonnegative integer m , $\binom{m}{k} = 0$ when $k < 0$ or $k > m$. For vectors and matrices, superscript T denotes transpose and $M \succeq 0$ means $z^T M z \geq 0$ for every real vector z of the appropriate dimension. All expectations below are with respect to the explicitly specified uniform finite sample.

The starting Ramsey facts are

$$\begin{aligned} R(3, 5) &= 14, & R(3, 6) &= 18, & R(4, 4) &= 18, & R(4, 5) &= 25, \\ R(4, 6) &\leq 40, & R(5, 5) &\leq 46. \end{aligned}$$

The last bound is Theorem 1.1 of Angeltveit–McKay [III.2]. The $R(4, 6)$ bound is recorded in Radziszowski’s survey [III.1, Table Ib], which attributes it to Angeltveit and McKay with personal communications from 2020 and 2024. The survey is the source used here for that bound.

A *complete catalogue* for a graph family is a finite list containing a representative of every isomorphism class in that family. Relabelling a graph does not change the counts e, t, q , so one representative per class is sufficient for these calculations. The catalogues used here are Brendan McKay’s published collections [III.4] for the following small families:

Family	Which graphs the supplied files cover
$\mathcal{R}(3, 5, n)$	Supplied orders up to 13
$\mathcal{R}(3, 6, n)$	Every order 13, 14, 15, 16, 17
$\mathcal{R}(4, 4, n)$	Every order from 5 through 17
$\mathcal{R}(4, 5, 24)$	All 352366 isomorphism classes
$\mathcal{R}(4, 5, n), 19 \leq n \leq 23$	The fixed-edge classes listed below

For example, $\mathcal{R}(3, 6, 17)$ means triangle-free graphs on 17 vertices with no independent set of six vertices; its catalogue contains seven graphs. The complete extreme-edge classes have the following integer edge counts:

Order n	Edge counts with complete supplied catalogues
19	57, 58, 90, 91, 92
20	68, 69, 98, 99, 100
21	77, 78, 106, 107
22	88, 89, 113, 114
23	101, 102, 103, 104, 119, 120, 121, 122

Each entry is a complete catalogue only for its stated order and edge count. For instance, **r4523.101.g6** contains the $(4, 5)$ -graphs on 23 vertices with exactly 101 edges. The 26 classes in the table are precisely the catalogue cases used in the recursive polygon construction.

For the $(4, 6)$ families on 38 or 39 vertices, the $(5, 5)$ families on 44 or 45 vertices, and the hypothetical $(5, 6)$ family on 84 vertices, the proof uses counting equations and information about smaller graphs. These larger families are treated through numerical constraints satisfied by every member, without enumerating their graphs. Where a complete catalogue is unavailable, we use a proved containing region for the possible profiles, defined in Section III.4.

Catalogue completeness is an external mathematical input from the cited source. The supplied programs check that the archived records and derived profiles agree, and that the record counts match the catalogue descriptions. The completeness assumption concerns the catalogue enumeration itself. A further established input, used for lower-order $(3, 6)$ outer profiles, is the small-order edge formula in [III.5, Theorem 2]; its specialization is stated in Section III.17.2.

The edge intervals used for $\mathcal{R}(4, 5, n)$ are

n	19	20	21	22	23	24
$e_{\min}$	57	68	77	88	101	116
$e_{\max}$	92	100	107	114	122	132

These lower and upper edge bounds are recorded in [III.2, Appendix A, Table 1]. They describe intervals containing every possible edge count. For orders 19 through 23, the 26 listed catalogue classes are treated directly and every other integer edge count in these intervals receives a containing polygon justified by further counting arguments. This supplies profile information for every edge class in the intervals.

III.1.1 Complement profiles

For a graph of order n , the exact complement transformation is

$$e(\overline{X}) = \binom{n}{2} - e(X), \quad (\text{III.1})$$

$$t(\overline{X}) = \binom{n}{3} - (n-2)e(X) + q(X), \quad (\text{III.2})$$

$$q(\overline{X}) = 2\binom{n}{3} - (n-2)e(X) + t(X). \quad (\text{III.3})$$

This permits a supplied catalogue to serve both $\mathcal{R}(s, r, n)$ and $\mathcal{R}(r, s, n)$. The transformation is affine in the profile coordinates; it therefore commutes with taking convex combinations.

III.2 Proof structure and verification

The theorem asserts that $\mathcal{R}(5, 6, 84)$ is empty. A hypothetical member would have 84 vertices, no five pairwise adjacent vertices, and no six pairwise nonadjacent vertices. The proof proceeds

by assigning numerical counts to such a graph and showing that those counts would satisfy contradictory inequalities.

A *counting model* in this document is a list of numerical variables and equations or inequalities that every graph in a stated family must satisfy. For example, the number of vertices of each degree is nonnegative, and the sum of those numbers is the order of the graph. A *relaxation* allows real values for its variables and imposes only the listed conditions. It may therefore allow numerical assignments that no graph can realize. Infeasibility of these weaker conditions therefore implies that the graph family is empty.

A *certificate* is a supplied list of rational coefficients for combining already justified equations and inequalities. Their weighted sum gives the required bound or contradiction. Verification consists of checking the supplied coefficients by exact addition, multiplication, and comparison. Section III.8 first explains this weighted-sum argument without matrix notation and then specifies its matrix representation.

The proof has two main counting models. The first is applied to graphs on 19–23, 38, 39, 44, or 45 vertices to establish local bounds. Its variables describe neighborhoods, nonneighborhoods, and four-vertex subsets. The second is applied to the hypothetical 84-vertex graph. Its variables are the proportions of seven-vertex subsets inducing each of the 979 allowed types. The local bounds supply some of the inequalities for that second model. A final weighted sum rules out every possible vector of those proportions.

III.2.1 A route through the mathematics and the programs

For mathematical review, first check the definitions and the triple count in the next section. Then follow the construction of the local variables and the equations they satisfy: this establishes that every graph gives an allowed numerical assignment. Next check the weighted-sum principle and the representative local calculation. The remaining sections derive the seven-vertex averages, the squared expressions, and the final contradiction. The separate *Explanation of the method* gives a longer worked derivation of one local bound, $e(X) \leq 314$ for $X \in \mathcal{R}(4, 6, 38)$.

For computational reproduction, use the commands in Section III.27. They check the supplied numerical coefficients and their links to the counting formulas. The default command runs the main certificate checks; the full command also re-extracts profiles from every raw catalogue record. The output is evidence that the submitted data satisfy the specified finite arithmetic tests. The mathematical review establishes why those tests imply statements about graphs. Both parts are needed.

A reader writing another verifier can use the displayed equations as the specification and the saved coefficients as input. In that case the essential comparisons are equality of rational numbers and completeness of the required cases. The stored ordering of variables and rows specifies how positions in the numerical arrays correspond to the definitions. A different ordering is valid if every dependent coefficient is permuted consistently.

III.2.2 Locating the files

The complete electronic verification archive is named `verification.tar.xz`. Extracting it creates a folder named `verification`, which contains `README.txt`, `verify_package.py`, and the subfolders used by the programs. The file `FILE_INDEX.tsv` maps each basename cited here to its location within that extracted folder. This document cites files by basename; the execution recipes give the required relative paths.

A JSON certificate is a text file containing labelled fields, such as a list of rational multipliers. A saved audit report records the outcome of a previous program run. The certificate is input to the verifier; the report records the checks performed and their outcomes.

III.3 The first counting reduction

Suppose $X \in \mathcal{R}(5, 6, 84)$. For each vertex v , set

$$A_v = X[N(v)], \quad B_v = X[V(X) \setminus (N(v) \cup \{v\})], \quad d_v = d_X(v).$$

A four-clique in A_v , together with v , would be a forbidden five-clique in X . An independent six-set in A_v would already be forbidden in X . Thus $A_v \in \mathcal{R}(4, 6, d_v)$. Similarly, B_v has no five-clique, and an independent five-set in B_v would extend with v to an independent six-set; hence $B_v \in \mathcal{R}(5, 5, 83 - d_v)$. The bounds $R(4, 6) \leq 40$ and $R(5, 5) \leq 46$ give $d_v \leq 39$ and $83 - d_v \leq 45$, respectively. Therefore

$$d_v \in \{38, 39\}. \quad (\text{III.4})$$

If x vertices have degree 39, handshaking gives $2e(X) = 3192 + x$. In particular x is even. The local cases are

d_v	A_v	B_v
38	$\mathcal{R}(4, 6, 38)$	$\mathcal{R}(5, 5, 45)$
39	$\mathcal{R}(4, 6, 39)$	$\mathcal{R}(5, 5, 44)$

The next identity holds for any graph X of order n . In this general identity, $A_v = X[N_X(v)]$, $B_v = X[V(X) \setminus (N_X(v) \cup \{v\})]$, and $d_v = d_X(v)$. Double counting yields

$$\sum_v \left[e(B_v) - e(A_v) - d_v \left(\frac{n}{2} - d_v \right) \right] = 0. \quad (\text{III.5})$$

To prove it, first count a triangle together with one of its vertices. The other two vertices form an edge in the chosen vertex's neighborhood, and each triangle is counted three times. Hence $\sum_v e(A_v) = 3t(X)$. For an edge uw , the number of vertices adjacent to neither endpoint is

$$n - d_u - d_w + |N(u) \cap N(w)|.$$

Summing this expression over all edges counts each edge of each B_v exactly once. The sum of $d_u + d_w$ over all edges uw is $\sum_v d_v^2$, because each vertex occurs as an endpoint of d_v edges. The common-neighbor terms count each triangle three times. Therefore

$$\sum_v e(B_v) = ne(X) - \sum_v d_v^2 + 3t(X).$$

Subtracting the neighborhood sum and using $\sum_v d_v = 2e(X)$ proves (III.5). This is the $m = 2$ instance of the general counting identity in [III.3], also stated in [III.2].

For $n = 84$, the contribution is $d_v(42 - d_v)$: it equals 152 at degree 38 and 117 at degree 39. These constants will combine with local edge bounds. The identity alone does not distinguish all possible local profiles, so the proof retains additional counts of triangles and wedges.

III.4 Variables of the local profile model

Fix a family $\mathcal{R}(s, r, n)$ and take X to be an arbitrary member. The construction in this section applies to each local family needed in the proof, including $(4, 6, 38)$ and $(5, 5, 45)$.

The local model records neighborhood profiles using containing polygons in the (t, q) -coordinates. The vertices of these polygons provide a finite set of coordinates for the model. A possible neighborhood profile can be a convex combination of several polygon vertices; its contribution is therefore split among them. The resulting variables are nonnegative weights, not necessarily integers. The construction below states exactly what the polygons contain, what labels the variables, and how every actual graph supplies values for them.

III.4.1 Actual profile sets and their outer polygons

For a family $\mathcal{R}(s, r, m)$ and an integer $0 \leq \eta \leq \binom{m}{2}$, define the finite set

$$\mathcal{S}_{s,r,m,\eta} = \{(t(Y), q(Y)) : Y \in \mathcal{R}(s, r, m), e(Y) = \eta\} \subseteq \mathbb{Z}_{\geq 0}^2. \quad (\text{III.6})$$

The coordinate η is held fixed in this definition. In particular, profiles with different edge counts are not combined in a single polygon.

For each required parameter tuple, the input supplies a bounded rational convex polygon $\mathcal{P}_{s,r,m,\eta} \subseteq \mathbb{R}^2$ with

$$\mathcal{S}_{s,r,m,\eta} \subseteq \mathcal{P}_{s,r,m,\eta}. \quad (\text{III.7})$$

Here “polygon” includes a line segment or a single point. An empty polygon is permitted only for a class independently justified to contain no graphs. Write

$$\mathcal{V}_{s,r,m,\eta} = \text{vert}(\mathcal{P}_{s,r,m,\eta}), \quad \mathcal{P}_{s,r,m,\eta} = \text{conv}(\mathcal{V}_{s,r,m,\eta}).$$

The finite set $\text{vert}(P)$ consists of the extreme points of P . A convex combination of points $v_1, \dots, v_k$ is a sum $\theta_1 v_1 + \dots + \theta_k v_k$ with $\theta_i \geq 0$ and $\sum_i \theta_i = 1$; the set $\text{conv}(V)$ consists of all such combinations of points in V . We set $\text{conv}(\emptyset) = \emptyset$. All elements of $\mathcal{V}$ have rational coordinates, denoted (τ, κ) ; these are candidate triangle and q coordinates, respectively.

There are two constructions of $\mathcal{P}$. If a complete catalogue is available for the edge class, take the convex hull of its extracted (t, q) pairs. In that case $\mathcal{P} = \text{conv}(\mathcal{S})$. Otherwise take the intersection of finitely many certified half-planes

$$L_\ell \leq \alpha_\ell \tau + \beta_\ell \kappa \leq U_\ell \quad (\ell = 1, \dots, m_0),$$

where m_0 is the number of support directions and all displayed coefficients are rational. Each support is proved for the fixed family, order, and edge count. Section III.9 specifies the recursive instances, directions, and files. An extreme point of such an outer polygon need not be the profile of any graph. Inclusion (III.7) ensures that every graph profile has a convex representation using the polygon vertices.

III.4.2 The finite coordinate sets

Fix the ambient local family $\mathcal{R}(s, r, n)$, with $s, r \geq 2$. Let $\mathcal{D} \subseteq \{0, \dots, n-1\}$ be the supplied set of admissible degrees, meaning that every vertex degree of every graph in this family lies in $\mathcal{D}$. Write $\mathcal{D}^B = \{n-1-d : d \in \mathcal{D}\}$. For every $d \in \mathcal{D}$, a neighborhood belongs to $\mathcal{R}(s-1, r, d)$ and a nonneighborhood belongs to $\mathcal{R}(s, r-1, n-1-d)$. Define

$$\begin{aligned} \mathcal{I}_A = \{ & (d, \eta, \tau, \kappa) : d \in \mathcal{D}, \quad \eta \in \mathbb{Z}, \quad 0 \leq \eta \leq \binom{d}{2}, \\ & (\tau, \kappa) \in \mathcal{V}_{s-1,r,d,\eta} \}, \end{aligned} \quad (\text{III.8})$$

$$\begin{aligned} \mathcal{I}_B = \{ & (h, \eta, \tau, \kappa) : h \in \mathcal{D}^B, \quad \eta \in \mathbb{Z}, \quad 0 \leq \eta \leq \binom{h}{2}, \\ & (\tau, \kappa) \in \mathcal{V}_{s,r-1,h,\eta} \}. \end{aligned} \quad (\text{III.9})$$

The four subscripts therefore have separate roles: local order, exact local edge count, polygon triangle coordinate, and polygon q coordinate. Empty vertex sets supply no coordinates. Each edge class must have either a justified covering polygon or a proof of emptiness. The full degree ranges and edge-class coverage are checked in Section III.9.

The profile variables of the linear program are one independent real variable for each element of these finite sets:

$$a_{d,\eta,\tau,\kappa} \geq 0 \quad ((d, \eta, \tau, \kappa) \in \mathcal{I}_A), \quad b_{h,\eta,\tau,\kappa} \geq 0 \quad ((h, \eta, \tau, \kappa) \in \mathcal{I}_B).$$

The variable $a_{d,\eta,\tau,\kappa}$ is the total weight assigned to the specified neighborhood profile vertex, and $b_{h,\eta,\tau,\kappa}$ is the analogous nonneighborhood weight. The following construction gives this statement a precise meaning for an actual graph.

III.4.3 From a graph to nonnegative weights

Let $X \in \mathcal{R}(s, r, n)$, and for $v \in V(X)$ put

$$d_v = d_X(v), \quad h_v = n - 1 - d_v, \quad A_v = X[N_X(v)], \quad B_v = X[V(X) \setminus (N_X(v) \cup \{v\})],$$

$$\eta_v^A = e(A_v), \quad \eta_v^B = e(B_v).$$

By (III.7), choose nonnegative numbers $\lambda_{v,\tau,\kappa}^A$, indexed by $(\tau, \kappa) \in \mathcal{V}_{s-1,r,d_v,\eta_v^A}$, such that

$$\sum_{\tau,\kappa} \lambda_{v,\tau,\kappa}^A = 1, \quad \sum_{\tau,\kappa} \lambda_{v,\tau,\kappa}^A (\tau, \kappa) = (t(A_v), q(A_v)). \quad (\text{III.10})$$

Choose $\lambda_{v,\tau,\kappa}^B$ in the same way using $\mathcal{V}_{s,r-1,h_v,\eta_v^B}$ and $(t(B_v), q(B_v))$. All sums in (III.10) run over the displayed vertex set. Extend each λ by zero outside that set. Now define

$$a_{d,\eta,\tau,\kappa} = \sum_{\substack{v \in V(X): d_v=d \\ \eta_v^A=\eta}} \lambda_{v,\tau,\kappa}^A, \quad (\text{III.11})$$

$$b_{h,\eta,\tau,\kappa} = \sum_{\substack{v \in V(X): h_v=h \\ \eta_v^B=\eta}} \lambda_{v,\tau,\kappa}^B. \quad (\text{III.12})$$

These formulas also define zero values when there are no vertices in the indicated degree and edge class. A graph need not determine unique weights: convex representations can be nonunique. The required counting identities hold for every such choice.

Define the degree-class masses

$$m_d^A = \sum_{\substack{(d',\eta,\tau,\kappa) \in \mathcal{I}_A \\ d'=d}} a_{d',\eta,\tau,\kappa}, \quad m_h^B = \sum_{\substack{(h',\eta,\tau,\kappa) \in \mathcal{I}_B \\ h'=h}} b_{h',\eta,\tau,\kappa}.$$

For the graph-derived weights they equal $|\{v : d_v = d\}|$ and $|\{v : h_v = h\}|$. Consequently the relaxation includes

$$\sum_{d \in \mathcal{D}} m_d^A = n, \quad m_d^A = m_{n-1-d}^B \quad (d \in \mathcal{D}). \quad (\text{III.13})$$

Arbitrary feasible weights of the relaxation, including these masses, are allowed to be fractional. Integrality is used only for statistics known to come from a graph, never assumed for the profile weights themselves.

Lemma III.4.1 (Preservation of local affine counts). *Let F be a real-valued function on the index set whose dependence on (τ, κ) , for each fixed (d, η) , extends to an affine function on $\mathbb{R}^2$. The graph-derived weights satisfy*

$$\sum_{(d,\eta,\tau,\kappa) \in \mathcal{I}_A} F(d, \eta, \tau, \kappa) a_{d,\eta,\tau,\kappa} = \sum_{v \in V(X)} F(d_v, \eta_v^A, t(A_v), q(A_v)).$$

The analogous identity holds on the B side with h_v and η_v^B .

Proof. Substitute (III.11), interchange the finite sums, and apply (III.10) separately for each vertex. The constant term is preserved because its weights sum to one; the two linear terms are preserved by the coordinate equation. The order and edge count stay fixed throughout each convex combination. $\square$

III.4.4 A catalogue example of fractional weights

The supplied file `r36_13.g6.gz.profiles` has, at edge count 16, exactly the pairs

$$\mathcal{S}_{3,6,13,16} = \{(0, 25), (0, 26), (0, 28)\}.$$

Its polygon is the segment with extreme points $(0, 25)$ and $(0, 28)$. The middle catalogue profile is represented by

$$(0, 26) = \frac{2}{3}(0, 25) + \frac{1}{3}(0, 28).$$

If a vertex of an ambient $(4, 6)$ -graph has this neighborhood profile, that single vertex contributes $2/3$ to $a_{13,16,0,25}$ and $1/3$ to $a_{13,16,0,28}$. Its total contribution to mass is 1, to local edges is 16, to local triangles is 0, and to local q is 26. There is no coordinate $a_{13,16,0,26}$ in the extreme-point model. The fractions are the coefficients representing this single catalogue profile as a convex combination of polygon vertices. The construction applies whenever such a neighborhood occurs.

III.4.5 Ambient statistics as linear functionals

For any profile weight vector a , define

$$\mathcal{E}(a) = \frac{1}{2} \sum_{\mathcal{I}_A} d a_{d,\eta,\tau,\kappa}, \quad (\text{III.14})$$

$$\mathcal{T}(a) = \frac{1}{3} \sum_{\mathcal{I}_A} \eta a_{d,\eta,\tau,\kappa}, \quad (\text{III.15})$$

$$\mathcal{W}(a) = \sum_{\mathcal{I}_A} \binom{d}{2} a_{d,\eta,\tau,\kappa}, \quad (\text{III.16})$$

$$\mathcal{Q}(a) = \mathcal{W}(a) - \mathcal{T}(a). \quad (\text{III.17})$$

Here $\sum_{\mathcal{I}_A}$ means a sum over every four-tuple in (III.8). For weights constructed from X , these functionals equal $e(X), t(X), w(X), q(X)$, respectively. Handshaking proves the first identity; each triangle contributes one edge to each of its three neighborhoods, proving the second; the degree-class masses prove the third. These functionals are defined even at a relaxed point that corresponds to no graph.

The implementation names their coefficient vectors `ec`, `tc`, `wc`, and `qc`. The metadata key `model.t` denotes the independence parameter r ; the vector `tc` and profile coordinate `tr` denote triangles. If the ambient edge count is fixed at the integer e_0 , add $\mathcal{E}(a) = e_0$. The neighborhood edge counts η still range over their allowed values.

III.5 Four-vertex variables and bulk equations

Let $\mathcal{F}_4$ contain one representative of each of the eleven isomorphism classes of graphs on four vertices. Introduce a real variable $z_F \geq 0$ for every $F \in \mathcal{F}_4$. The value assigned by an actual graph X is

$$z_F(X) = |\{S \in \binom{V(X)}{4} : X[S] \cong F\}|.$$

An arbitrary relaxed z_F need not be an integer. Define $u(F) = |\{v \in V(F) : d_F(v) = 3\}|$ and $i(F) = |\{v \in V(F) : d_F(v) = 0\}|$, the numbers of universal and isolated vertices. The remaining entries in the following table are $e(F), t(F), w(F)$ as defined in Section III.1. Every sum over F in the local model ranges over $\mathcal{F}_4$.

F	e	t	w	u	i
$4K_1$	0	0	0	0	4
$K_2 + 2K_1$	1	0	0	0	2
$2K_2$	2	0	0	0	0
$P_3 + K_1$	2	0	1	0	1
$K_3 + K_1$	3	1	3	0	1
$K_{1,3}$	3	0	3	1	0
P_4	3	0	2	0	0
C_4	4	0	4	0	0
Paw	4	1	5	1	0
$K_4 - e$	5	2	8	2	0
K_4	6	4	12	4	0

The paw is a triangle with a pendant vertex. The code uses the NetworkX graph-atlas ordering; another ordering is acceptable if every corresponding column and objective is permuted consistently.

These variables connect local information at different vertices: the same four-set may be seen from several roots, but must receive one consistent induced type. The following four equations count all four-sets and the edges, triangles, or wedges contained in them:

$$\sum_F z_F = \binom{n}{4}, \quad (\text{III.18})$$

$$\sum_F e(F) z_F = \binom{n-2}{2} \mathcal{E}(a), \quad (\text{III.19})$$

$$\sum_F t(F) z_F = (n-3) \mathcal{T}(a), \quad (\text{III.20})$$

$$\sum_F w(F) z_F = (n-3) \mathcal{W}(a). \quad (\text{III.21})$$

An edge lies in $\binom{n-2}{2}$ four-sets; a triangle or wedge occupies three vertices and lies in $n-3$ four-sets. The counted edge, triangle, or wedge need not be induced within the four-set.

Some four-vertex counts are known to be zero before any numerical calculation. If $s \leq 4$, a graph in $\mathcal{R}(s, r, n)$ cannot contain a K_4 , so $z_{K_4} = 0$. If $r \leq 4$, it cannot contain an independent four-set, so $z_{4K_1} = 0$. These are the coordinates called *fixed to zero* in the certificate description. For example, the model for $\mathcal{R}(4, 6, 38)$ has $z_{K_4} = 0$ because a four-clique is forbidden. Thus the family definition determines these zero coordinates. All other retained coordinates are constrained to be nonnegative. Their role in the weighted-sum check is explained in Section III.8.

III.6 Triple types and rooted incidence equations

For an integer order m and coordinates (η, τ, κ) , define the four-component polynomial vector

$$T(m, \eta, \tau, \kappa) = \left(\binom{m}{3} - (m-2)\eta + \kappa, \right. \\ \left. (m-2)\eta - 2\kappa + \tau, \kappa - 2\tau, \tau \right). \quad (\text{III.22})$$

Write T_k for its component indexed by $k \in \{0, 1, 2, 3\}$ in the displayed order. When (m, η, τ, κ) is a graph profile, T_k is the number of its induced three-sets with exactly k edges. Indeed, these counts sum to $\binom{m}{3}$, their edge incidence sum is $(m-2)\eta$, and $T_2 + 3T_3 = w = \kappa + \tau$, while $T_3 = \tau$. These relations give (III.22). The polynomial also defines the coefficients at rational polygon vertices.

For $k \in \{0, 1, 2, 3\}$ and $F \in \mathcal{F}_4$, let $U_k(F)$ count degree-3 vertices of F whose deletion leaves k edges, and let $Z_k(F)$ count degree-0 vertices with that property. The rooted incidence rows are

$$\sum_{\mathcal{I}_A} T_k(d, \eta, \tau, \kappa) a_{d, \eta, \tau, \kappa} = \sum_F U_k(F) z_F, \quad (\text{III.23})$$

$$\sum_{\mathcal{I}_B} T_k(h, \eta, \tau, \kappa) b_{h, \eta, \tau, \kappa} = \sum_F Z_k(F) z_F. \quad (\text{III.24})$$

These eight equations count a triple together with a universal or isolated fourth vertex. To obtain their coefficients, observe that removing a universal vertex deletes exactly three edges, whereas removing an isolated vertex deletes none. Therefore

$$U_k(F) = u(F) \mathbf{1}_{e(F)-3=k}, \quad Z_k(F) = i(F) \mathbf{1}_{e(F)=k}.$$

For example, removing the universal vertex of a paw leaves one edge, so it contributes to U_1 . The isolated vertex of $K_3 + K_1$ contributes to Z_3 .

The local model also includes the separate triple identity (III.5), in the explicit linear form

$$\sum_{\mathcal{I}_B} \eta b_{h, \eta, \tau, \kappa} - \sum_{\mathcal{I}_A} \left[\eta + d \left(\frac{n}{2} - d \right) \right] a_{d, \eta, \tau, \kappa} = 0. \quad (\text{III.25})$$

This identity is included as its own row in the model. Lemma III.4.1 proves that the graph-derived weights preserve every local count in these equations.

The rooted equations require the neighborhood and nonneighborhood triple patterns to fit into a common four-vertex distribution. They express compatibility between local counts at different vertices.

III.7 Joint degrees, disjoint edges, and paths

The smaller model called version 3 contains profile matching, the optional fixed-edge row, the triple identity, four bulk identities, and eight rooted identities. Version 4 additionally uses the constraints in this section.

When $e(X) = e_0$ is fixed, count pairs of disjoint edges:

$$\mathcal{W}(a) + \sum_F \left(\binom{e(F)}{2} - w(F) \right) z_F = \binom{e_0}{2}. \quad (\text{III.26})$$

With e_0 fixed, the right-hand side is constant and this is a linear equation in the model variables. It is included only in models with a prescribed edge count.

For each $d, f \in \mathcal{D}$ with $d \leq f$, introduce a real variable $j_{d, f} \geq 0$. At the point assigned by a graph, $j_{d, f}$ counts unordered edges whose endpoint degrees, in increasing order, are (d, f) . In particular, $j_{d, d}$ counts each edge within a degree class once. Count the ends of edges incident with vertices of degree d . The left-hand side below counts d ends per vertex in that degree class; the right-hand side groups the same ends according to the degree of the other endpoint. This gives, for every $d \in \mathcal{D}$,

$$dm_d^A = 2j_{d, d} + \sum_{\substack{f \in \mathcal{D} \\ f < d}} j_{f, d} + \sum_{\substack{f \in \mathcal{D} \\ f > d}} j_{d, f}. \quad (\text{III.27})$$

The diagonal factor is 2 because each edge within a class has two ends in that class.

Let $p(Y)$ count unoriented paths with three edges and four distinct vertices in a graph Y , with additional edges permitted among their vertices. Equivalently, it is half the number of ordered injective four-tuples (v_1, v_2, v_3, v_4) with all three consecutive pairs adjacent. The path row is

$$\sum_F p(F) z_F = \sum_{\substack{d, f \in \mathcal{D} \\ d \leq f}} (d-1)(f-1) j_{d, f} - 3\mathcal{T}(a). \quad (\text{III.28})$$

Choose the middle edge uv . There are $(d_u - 1)(d_v - 1)$ choices of outer endpoints before requiring them to be distinct. Each common neighbor gives one invalid coincidence, and these total $3t(X)$ over all edges. On a graph-derived point the left-hand side is $p(X)$, and the coefficients can be computed from

$$p(F) = \sum_{uv \in E(F)} (d_F(u) - 1)(d_F(v) - 1) - 3t(F).$$

This provides another linear equation.

Put $D_0 = |\mathcal{D}|$ and let $F_0 = 1$ when the ambient edge count is fixed, and $F_0 = 0$ otherwise. Version 3 has $D_0 + 14 + F_0$ rows and version 4 has $2D_0 + 15 + 2F_0$. The closure auditor checks these row counts and verifies that the version 3 matrix contains precisely its specified variables and equations.

Proposition III.7.1 (Every graph gives a feasible local point). *Suppose the admissible degree and profile coverage conditions above hold. Every $X \in \mathcal{R}(s, r, n)$ gives a feasible point of version 3 and version 4, and of their fixed-edge variants when $e(X) = e_0$.*

Proof. Choose the weights by (III.10)–(III.12); take $z_F = z_F(X)$ and the actual joint-degree edge counts $j_{d,f}$. Nonnegativity and degree matching follow from the definitions. The bulk equations count subsets containing a fixed edge, triangle, or wedge. Lemma III.4.1 converts the rooted incidence counts and (III.5) into (III.23)–(III.25). Stub matching counts ends of edges. The disjoint-edge equation partitions edge pairs into those meeting at a vertex and those on four vertices, and the path equation follows from its middle-edge count. Forbidden four-vertex variables vanish. Finally $\mathcal{E}(a) = e(X)$, so the optional fixed-edge row also holds. The model imposes necessary conditions for membership in the graph family. $\square$

III.8 Checking a bound by a weighted sum

The variables of the local model are the neighborhood weights $a_{d,\eta,\tau,\kappa}$, nonneighborhood weights $b_{h,\eta,\tau,\kappa}$, four-vertex counts z_F , and, in version 4, edge counts $j_{d,f}$ by endpoint degree. Every actual graph supplies nonnegative values satisfying the equations, as Proposition III.7.1 proves. A certificate combines these equations to bound an edge or triangle count.

III.8.1 The principle before matrix notation

Suppose, as a small algebraic example, that nonnegative real numbers u_1, u_2, u_3 satisfy

$$u_1 + u_2 = 5, \quad u_2 + u_3 = 4.$$

Multiply the first equation by 2 and the second by 1, and add:

$$2u_1 + 3u_2 + u_3 = 14.$$

The quantity $3u_1 + 3u_2 + u_3$ is therefore at least 14, since

$$3u_1 + 3u_2 + u_3 = 14 + u_1 \geq 14.$$

The pair of multipliers $(2, 1)$ is a certificate for the bound. Verification consists of adding the two weighted equations and checking that the remaining coefficient is nonnegative.

The Ramsey certificates use exactly this argument with more variables and equations. For example, $e(X)$ is the linear expression $\mathcal{E}(a)$ defined above, and an upper bound on it is obtained by proving a lower bound on $-\mathcal{E}(a)$. A claimed upper bound is thus ultimately an identity expressing $-\mathcal{E}(a)$ as a constant plus a nonnegative combination of the model variables. The certificate supplies the equation multipliers needed to obtain that identity.

III.8.2 How the equations become a matrix

Choose the saved ordering of the variables and place their values in a column vector $\mathbf{u}$. Let N_{var} be its length and m_{eq} the number of equations. In each equation move all variable terms to the left and leave the constant on the right. Define $M_{\ell j}$ to be the coefficient of the j th variable in the ℓ th equation, and $\mathbf{b}_\ell$ to be that equation's right-hand side. Thus

$$M \in \mathbb{Q}^{m_{\text{eq}} \times N_{\text{var}}}, \quad \mathbf{b} \in \mathbb{Q}^{m_{\text{eq}}}, \quad M\mathbf{u} = \mathbf{b}, \quad \mathbf{u} \geq 0.$$

The matrix records the coefficients of the counting equations. For instance, the mass equation has coefficient 1 at every neighborhood-weight coordinate, coefficient 0 at all other coordinates, and right-hand side n . The entries of $\mathbf{b}$ are constants; the nonneighborhood weights are among the coordinates of $\mathbf{u}$.

Let $\mathbf{c} \in \mathbb{Q}^{N_{\text{var}}}$ record the coefficients of the quantity to be bounded; this quantity is $\mathbf{c}^T \mathbf{u}$. It is called the *objective*. For an edge lower bound its coefficients are those of $\mathcal{E}(a)$: $d/2$ at the neighborhood coordinate (d, η, τ, κ) and zero at the other coordinates. For an edge upper bound the signs of those coefficients are reversed. A triangle objective uses $\eta/3$ instead.

Now let $\mathbf{y} \in \mathbb{Q}^{m_{\text{eq}}}$ give one multiplier per equation. Multiplying the ℓ th equation by $\mathbf{y}_\ell$ and summing gives

$$\sum_{j=1}^{N_{\text{var}}} \left(\sum_{\ell=1}^{m_{\text{eq}}} M_{\ell j} \mathbf{y}_\ell \right) \mathbf{u}_j = \sum_{\ell=1}^{m_{\text{eq}}} \mathbf{y}_\ell \mathbf{b}_\ell.$$

The inner sum is the j th coordinate of $M^T \mathbf{y}$. It is the sum of the contributions to one variable from all the weighted equations.

III.8.3 What signs must be checked

Let J_0 be the set of variable positions occupied by counts already known to be zero, such as z_{K_4} in a K_4 -free family. Let J_+ be the other positions. The certificate is valid if

$$\mathbf{c}_j - (M^T \mathbf{y})_j \geq 0 \quad (j \in J_+).$$

Subtracting the weighted equation sum from the objective proves the explicit identity

$$\mathbf{c}^T \mathbf{u} - \mathbf{y}^T \mathbf{b} = \sum_{j \in J_+} \left(\mathbf{c}_j - (M^T \mathbf{y})_j \right) \mathbf{u}_j \geq 0. \quad (\text{III.29})$$

Every term on the right is nonnegative. A coordinate in J_0 contributes zero regardless of its coefficient, so it is omitted from this sign comparison. The equation multipliers may be negative because they multiply equalities. By contrast, the coefficients remaining on the nonnegative variables must have the displayed sign.

The multiplier vector $\mathbf{y}$ is called a *dual certificate*. Its validity follows directly from (III.29). Verification has two parts: proving that the rows of M are valid counting equations, and checking that the supplied $\mathbf{y}$ gives nonnegative residual coefficients.

III.8.4 Matching the displayed equations to the stored rows

The mathematical triple identity is equation (III.25). For integer coefficient storage, the implementation records twice that row. It also records twice the fixed-edge equation, twice the four-set edge-incidence equation, and three times the four-set triangle-incidence equation; the other displayed rows retain their stated scale. Multiplying an equation by a nonzero constant leaves its solutions unchanged, but changes the multiplier needed in a certificate. Accordingly, the verifier uses the stored row scales and the stored multipliers together. To use the displayed equations, multiply each stored certificate multiplier by the corresponding row's storage factor.

For example, the mathematical fixed-edge equation is $\mathcal{E}(a) = e_0$. Its stored version has coefficient d at each neighborhood variable and right-hand side $2e_0$. The worked-method document shows the analogous conversion explicitly for the triple row in the 314-edge example.

III.8.5 Files, exact arithmetic, and the check to perform

A local JSON certificate records the family parameters, the objective coefficients, the equation multipliers, the rounding threshold, and the exact resulting lower bound. Rational entries are strings such as "3/7", with numerator and denominator separated by a slash. The fields named **objective** and **dual** contain **c** and **y**, respectively. The model parameters tell the verifier which profile sets and counting equations to reconstruct.

The verifier first reconstructs the objective from the mathematical claim and compares it with the stored objective. It then calculates every residual coefficient in (III.29), checks its sign, and computes $\mathbf{y}^T \mathbf{b}$. Finally it checks that this rational bound is strong enough for the stated integer conclusion. The saved field **max_dual_violation** records a diagnostic result; the verifier recomputes the residuals from the multiplier vector.

Integer arithmetic can perform all these steps. Let positive integers L, D clear denominators in $M, \mathbf{b}$ and in $\mathbf{y}, \mathbf{c}$, respectively, and define

$$A = LM, \quad B = L\mathbf{b}, \quad Y = D\mathbf{y}, \quad C = LD\mathbf{c}.$$

Here A is an integer matrix and B, Y, C are integer vectors used only for this arithmetic representation. The sign check is $(A^T Y)_j - C_j \leq 0$ for $j \in J_+$, and the resulting bound is $Y^T B / (LD)$. The replay uses arbitrary-precision integers, so accumulation does not wrap at a fixed machine word size. The common matrix denominators occurring in the required chain are 1 and 6.

The program `replay_required_local_certificates.py` performs these checks for every required local claim. Its report names the checked certificates and records their hashes. The derivations above justify the model variables and rows. The program verifies each resulting rational weighted sum by the exact residual and threshold tests.

III.9 Recursive profile coverage and exact polygons

The weighted-sum principle is first used to supply the smaller local information needed by later models. For example, a neighborhood of a vertex in a $(4, 6)$ -graph is a $(3, 6)$ -graph, while its nonneighborhood is a $(4, 5)$ -graph. Complete $(3, 6)$ catalogues are available at the required orders, but not every required $(4, 5)$ edge class has a complete catalogue. The following construction supplies proved containing polygons for the latter classes.

The $\mathcal{R}(4, 5, n)$ inputs for $19 \leq n \leq 23$ consist of 149 edge classes. Of these, 26 use complete catalogue hulls; the remaining 123 use certified support polygons. They are stored in `computed_v3_45_19.json` through `computed_v3_45_23.json`.

For each of the 123 edge classes without a complete catalogue, a smaller counting model gives both a lower and an upper bound on $at + bq$. Each such linear bound is called a support inequality. The eight fixed coefficient pairs (a, b) are:

$$(1, 0), (0, 1), (1, 1), (1, -1), (1, 2), (2, 1), (1, -2), (2, -1).$$

Thus there are $123 \cdot 8 \cdot 2 = 1968$ support certificates. Since t, q are integers for a graph, $at + bq$ is integral for an integer direction (a, b) . To prove $at + bq \leq U$, a certificate gives

$$-(at + bq) > -U - 1.$$

The strict inequality and integrality imply the desired bound. Equality with $-U - 1$ would not justify it.

To reconstruct a polygon, intersect pairs of nonparallel rational boundary lines, retain intersections satisfying all half-planes, and compute their exact convex hull. Orientation tests use

$$(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1).$$

Segments and singletons are handled without perturbation. The largest denominator of the reconstructed rational polygon vertices is 3. The JSON coordinates include decimal approximations to thirds; they are not all exact terminating decimals. The polygon auditor recovers rational coordinates with denominator at most 10^6 and checks a discrepancy below 10^{-10} , then independently verifies equality with the exact half-plane or catalogue hull. The local replay recovers matrix coefficients with denominator at most 6 and checks a discrepancy below 10^{-8} . These closeness tests identify the intended rational data; the final residual inequalities and polygon comparisons are performed exactly.

The script `audit_recursive_polygons.py` compares stored vertices with the exact intersection of the support half-planes, or with the exact catalogue hull as appropriate. The dual replay establishes the support inequalities used in that intersection.

The models include the following complete degree ranges:

Family	Degrees
$\mathcal{R}(4, 5, n)$, $19 \leq n \leq 23$	$n - 18, \dots, 13$
$\mathcal{R}(4, 6, 38)$	$13, \dots, 17$
$\mathcal{R}(4, 6, 39)$	$14, \dots, 17$
$\mathcal{R}(5, 5, 44)$	$19, \dots, 24$
$\mathcal{R}(5, 5, 45)$	$20, \dots, 24$

For example, in a $(4, 6, 38)$ graph a neighborhood has no triangle and no independent six-set, so its order is at most 17; the nonneighborhood is a $(4, 5)$ graph of order at most 24, so the degree is at least $38 - 1 - 24 = 13$. The other ranges follow in the same way from the displayed small Ramsey bounds. The closure audit checks that the program includes each degree in these ranges and every integer edge class allowed for its two local families. Every degree and edge class in the required ranges must be covered before the model can be used.

III.10 Endpoints, integrality, and the five local facets

The endpoint chain gives

Family	Lower edge bound	Upper edge bound
$\mathcal{R}(4, 6, 38)$	267	314
$\mathcal{R}(4, 6, 39)$	288	324
$\mathcal{R}(5, 5, 44)$	432	514
$\mathcal{R}(5, 5, 45)$	460	530

The following certificates prove the endpoints of these intervals. The lower certificates `edge_46_38_min.json` and `edge_46_39_min.json` give, respectively,

$$e \geq \frac{43377}{163} > 266, \quad e \geq \frac{16965}{59} > 287.$$

Thus their integer lower endpoints are 267 and 288. The upper certificates `edge_55_44_max.json` and `edge_55_45_max.json` give

$$-e \geq -\frac{113652}{221} > -515, \quad -e \geq -\frac{73710}{139} > -531,$$

so their integer upper endpoints are 514 and 530.

The file `edge_46_38_max.json` certifies $-e \geq -26125/83$. Since $83 \cdot 314 = 26062 < 26125 < 26145 = 83 \cdot 315$, this gives $e \leq 314$. Its model has 25 rows and 1217 variables; 17 dual multipliers are nonzero.

The order-44 file `edge_55_44_min.json` gives $e \geq 95414/221 > 431$, so $e \geq 432$. For order 45, `edge_55_45_min.json` gives

$$e \geq \frac{12137627230191}{26402663302} > 459,$$

so $e \geq 460$. The rational bounds are compared with integer thresholds by exact cross multiplication.

For $\mathcal{R}(4, 6, 39)$, `edge_46_39_max.json` gives $-e \geq -325$, hence $e \leq 325$. With $e = 325$, two further certificates give

$$t \geq 1612/3, \quad -t \geq -1612/3.$$

Thus $t = 1612/3$, impossible for a graph. This excludes the last edge count and proves $e \leq 324$. The files are `exclude_4639e325_t_gt537.json` and `exclude_4639e325_t_lt538.json`.

Apply these bounds to the neighborhoods A_v and nonneighborhoods B_v of a hypothetical 84-vertex graph X . The relevant four edge bounds alone do not contradict (III.5): at degree 38, its summand is bounded below by $460 - 314 - 152 = -6$, and at degree 39 by $432 - 324 - 117 = -9$. An argument using only those four bounds therefore stops short of a contradiction. The proof also retains relationships among local edges, triangles, and wedges.

The next table states five such relationships. In each row Y means the local graph named in the first column. The quantity d_v is the degree of its root v in the 84-vertex graph X ; in particular, when $Y = B_v$, its order is $83 - d_v$. The five numbers $\gamma_d, \gamma_e, \gamma_t, \gamma_q, \gamma_0$ are fixed coefficients of the inequality

$$\gamma_d d_v + \gamma_e e(Y) + \gamma_t t(Y) + \gamma_q q(Y) + \gamma_0 \leq 0.$$

Each table row is a universal assertion for the indicated neighborhood or nonneighborhood family: it must hold at every vertex of any hypothetical X .

Y	File ID	γ_d	γ_e	γ_t	γ_q	γ_0
A_v	203	770	-269	-6	12	5940
B_v	595	-732	-76	-1	2	49239
B_v	603	-701	-75	-1	2	47596
B_v	658	-483	-35	-2	1	29877
B_v	681	-403	-37	-1	1	26296

For example, the first row spells out

$$770d_v - 269e(A_v) - 6t(A_v) + 12q(A_v) + 5940 \leq 0,$$

and the second spells out

$$-732d_v - 76e(B_v) - t(B_v) + 2q(B_v) + 49239 \leq 0.$$

The file IDs 203, 595, 603, 658, 681 identify these five coefficient tuples in the saved data. The metadata file `exact_candidate_used_facets.json` lists the IDs, their coefficient tuples, and which side they concern. The list in `local_profile_facets.json` is indexed by putting the neighborhood list before the nonneighborhood list, with indexing starting at zero. The data files call these inequalities *facets*. Each is used as a valid supporting inequality; the proof requires validity on all relevant profiles.

To prove each inequality for graphs on 38, 39, 44, or 45 vertices, fix the order of Y and an integer value $e(Y) = e_0$ from the proved intervals. Construct the local counting model for that family with this fixed edge count. A rational weighted sum of its equations then bounds the remaining expression in $t(Y)$ and $q(Y)$. Since every possible Y supplies a feasible assignment, that single calculation proves the inequality for every graph in the chosen order and edge class. The next paragraph gives one such calculation explicitly.

For example, facet 203 at $d_v = 38, e = 314$ becomes $-6t + 12q \leq 49266$. The file `facet203_46_n38_e314.json` proves

$$6t - 12q \geq -541956/11 > -49272.$$

The strict gap is $36/11$, since $49272 \cdot 11 - 541956 = 36$. Because $-6t + 12q$ is a multiple of 6, it is at most 49266.

In general, abbreviate $t = t(Y)$ and $q = q(Y)$ for this calculation. At fixed $e(Y) = e_0$, put $K = -\gamma_d d_v - \gamma_e e_0 - \gamma_0$ and $g = \gcd(|\gamma_t|, |\gamma_q|)$. The next possible value of $\gamma_t t + \gamma_q q$ above K is $g(\lfloor K/g \rfloor + 1)$. Its exclusion is the strict dual threshold. The proved edge intervals determine the complete set of calculations. The neighborhood inequality has $48 + 37 = 85$ cases, for the 48 possible edge counts at order 38 and the 37 at order 39. Each of the four nonneighborhood inequalities has $83 + 71 = 154$ cases, for orders 44 and 45. Hence the five inequalities require $85 + 4 \cdot 154 = 701$ fixed-edge certificates. Including the 10 endpoint/exclusion certificates and 1968 prerequisite support certificates gives 2679 required local certificates in total.

To check the table computationally, the reader runs `replay_required_local_certificates.py` as specified in Section III.27. That program determines the required cases from the intervals and coefficient tuples, reconstructs each counting matrix and objective, and verifies its exact weighted-sum certificate. Its successful output must account for all 701 cases and their prerequisites. Each JSON certificate covers one fixed-edge case.

III.11 Seven-vertex probabilities and completeness

Let $\mathcal{H}$ contain one representative of each isomorphism class of seven-vertex graphs containing neither K_5 nor an independent six-set. Each representative has the fixed labelled vertex set $\{0, \dots, 6\}$ for computations. It has 979 members. In the remaining counting arguments, let X be a hypothetical member of $\mathcal{R}(5, 6, 84)$. For each $H \in \mathcal{H}$, define

$$p_H = \frac{\#\{W \subseteq V(X) : |W| = 7, X[W] \cong H\}}{\binom{84}{7}}.$$

Then $p_H \geq 0$ and $\sum_H p_H = 1$. Every induced seven-set statistic is linear in this probability vector. All coefficient arrays use the same representative order, stored in `profile_cache.npz`.

The set of seven-vertex types is enumerated directly as follows. The probabilities p_H are unknown because X is hypothetical. The coefficient of any stated seven-vertex statistic can be evaluated on each of the 979 representatives. An inequality for its average is therefore a linear inequality in the unknown p_H . Proving that no nonnegative probabilities with sum one can satisfy all those inequalities rules out X .

The graph atlas supplies the representative list, and exhaustive labelled enumeration checks its completeness. Encode a labelled seven-vertex graph by its 21 adjacency bits, enumerate all $2^{21} = 2097152$ masks, and directly test all five-subsets and six-subsets. Exactly 2062423 labelled graphs survive.

For each saved representative, compute its minimum adjacency mask (the integer encoding defined below) over all $7!$ permutations. Verify these canonical masks are distinct, and count each representative's automorphisms. Here $\text{Aut}(H)$ is the group of permutations of $V(H)$ preserving adjacency. Its orbit has $7!/|\text{Aut}(H)|$ labelled graphs. The sums of orbit sizes, separately by edge count, must equal the exhaustive labelled counts. This proves both nonduplication and completeness of the 979 types.

The script `audit_seven_graph_completeness.py` performs this full finite enumeration. It verifies completeness of the seven-vertex type set within the supplied suite.

All dependent arrays use this representative ordering. Hashes and index maps check their consistency. A reimplementing using another order can compare coefficients by graph type.

III.12 Exact finite sampling

The local inequalities concern quantities such as the degree of a root or the number of edges in a common neighborhood. A seven-set usually contains only part of that information. We therefore assign weights to the portions it contains so that averaging over all seven-sets recovers the full quantity exactly. The sampling formulas below describe these exact averages.

A *root* is a vertex that has been distinguished and whose role must be preserved when configurations are compared. A *flag* is a finite graph together with specified distinguished vertices (roots), whose roles are either an ordered tuple or distinguished unordered groups. Its *type* is its isomorphism class under maps preserving these roles. Vertices not designated as roots form its tail. The flag order f counts both roots and tail vertices. A placement of a fixed labelled flag F into a graph Y is an injective map $\phi : V(F) \rightarrow V(Y)$ inducing all edges and nonedges of F and carrying its distinguished roles to the chosen roots. Denote the finite set of such maps by $\text{Emb}(F, Y)$. Taking orbits under automorphisms of F preserving the roles gives the unlabelled placements used by several builders; the conventions are specified in the next section.

Fix a placement of a flag on $f \leq 7$ vertices in a graph of order $N = 84$, and let W be its vertex image. An *exterior* vertex means a vertex in $V(X) \setminus W$. A seven-set containing the flag samples $7 - f$ vertices without replacement from the remaining $N - f$. For $0 \leq m \leq 7 - f$, a

specified configuration using m distinct exterior vertices is included with probability

$$\frac{\binom{N-f-m}{7-f-m}}{\binom{N-f}{7-f}} = \frac{\binom{7-f}{m}}{\binom{N-f}{m}}.$$

Thus its observed count is multiplied by

$$W_{f,m} = \frac{\binom{N-f}{m}}{\binom{7-f}{m}} \quad (\text{III.30})$$

to obtain an *unbiased estimate*: a quantity whose average over all choices of the sampled vertices is exactly the full count being studied. Here “conditional on the flag” means that its vertices and induced adjacencies are held fixed while the exterior vertices vary.

For example, a fixed six-vertex flag has 78 exterior vertices. A seven-set containing it selects one of those 78 vertices uniformly. If exactly c of the exterior vertices lie in a particular cell, its membership indicator is 1 for c choices and 0 for the others. Multiplying that indicator by 78 gives average c . For a pair or triple of exterior vertices the same argument gives the binomial factors above.

Lemma III.12.1. *For a sum of weighted configurations using at most $7 - f$ distinct exterior vertices, multiply each observed configuration by (III.30), classified by its number of distinct exterior vertices. The conditional expectation is exactly the original statistic.*

Proof. The inclusion probability cancels the factor separately for each configuration. Sum these equalities by linearity of expectation. $\square$

The word “distinct” is essential for products. In $d_x d_y$, one exterior vertex may contribute to both degrees; that term uses one vertex, not two. Repeated positions in a tuple must be handled before assigning a sampling weight.

More explicitly, suppose $I(X, \phi) \leq 0$ for every placement $\phi \in \text{Emb}(F, X)$, and let $\hat{I}(H, \phi)$ be an estimator formed by the lemma. Its conditional expectation over seven-sets containing the image of ϕ is $I(X, \phi)$. Define the coefficient of a seven-vertex representative by

$$U_F(H) = \sum_{\phi \in \text{Emb}(F, H)} \hat{I}(H, \phi).$$

Interchanging the finite sums over placements and seven-sets proves

$$\sum_{H \in \mathcal{H}} p_H U_F(H) = \frac{\binom{84-f}{7-f}}{\binom{84}{7}} \sum_{\phi \in \text{Emb}(F, X)} I(X, \phi) \leq 0. \quad (\text{III.31})$$

Indeed, each fixed placement belongs to $\binom{84-f}{7-f}$ seven-sets. Equalities lift in the same manner. Counting unlabelled placements divides both sides by the same positive role-preserving automorphism count. Integer scaling and normalization are subsequently recorded in the row metadata. This operation is called *lifting* a local inequality to a global coefficient row.

For flags of sizes 6, 5, 4, the exterior population sizes are 78, 79, 80. These explain the numerical constants throughout the builders. They are computed for the target order 84.

III.13 Root conventions and canonical masks

Ordered roots are fixed pointwise under isomorphism. For an unordered clique group C and independent group D , permutations within each group are allowed, but the groups are not exchanged. Unrooted flag vertices may be permuted among themselves. These conventions are implemented by `canon` and `group_canon`.

For any ambient graph Y , define $a_{uv} = \mathbf{1}_{\{u,v\} \in E(Y)}$ for distinct vertices u, v , and $a_{uu} = 0$. Given an ordered list $(v_0, \dots, v_{f-1})$ of distinct vertices of Y , encode its induced adjacency by

$$\text{mask}(Y; v_0, \dots, v_{f-1}) = \sum_{0 \leq i < j < f} a_{v_i v_j} 2^{\iota_f(i,j)}, \quad \iota_f(i,j) = \frac{i(2f-i-1)}{2} + j - i - 1.$$

Thus pairs are in lexicographic order, starting with exponent 0 for $(0, 1)$. A graph with its specified vertex order has this mask; root masks use the induced ordered root list. Canonicalization takes the least mask over the allowed permutations. The resulting integer identifies the rooted isomorphism type.

The builders generally enumerate unordered root groups and canonicalize. Several auditors deliberately use a different representation: enumerate all ordered injections inducing the fixed representative mask, then divide by the number of automorphisms preserving the distinguished groups. This division is checked to be exact. It reproduces the same placements and tests root multiplicities independently of the original enumeration.

The Gram construction, by contrast, sums over ordered root tuples and ordered pairs of unordered extension sets. The stored row scales and final multipliers correspond to these counting conventions.

Key files therefore form part of the computational interface. A zero row may be omitted and row order may follow insertion order. An independent implementation should match semantic keys or preserve the saved index ordering exactly.

III.14 Base degree and common-neighborhood rows

Condition on a six-vertex flag rooted at v . If r_0 counts its neighbors within the flag and a indicates adjacency to the single exterior sample, then $\hat{d} = r_0 + 78a$ is unbiased for d_v . The two degree inequalities give the rows $\hat{d} - 39$ and $38 - \hat{d}$.

An edge has at most 17 common neighbors, since its common neighborhood is a $(3, 6)$ -graph. A nonedge has at most 24 common nonneighbors, since they form a $(5, 4)$ -graph. A triangle has at most 5 common neighbors, and an independent triple has at most 13 common nonneighbors. If r_0 counts the appropriate cell inside a six-vertex flag and χ marks the exterior vertex, the estimator is $r_0 + 78\chi$. Subtracting the capacity produces a nonpositive row.

III.14.1 Degree-square equalities

The binary degree condition gives $(d_v - 38)(d_v - 39) = 0$. Condition on a five-vertex flag. If r_0 counts root neighbors inside it and a, b are adjacency indicators for the two exterior samples, twice an unbiased estimator is

$$2 \cdot 79 \cdot 78ab + (2r_0 - 76)79(a + b) + 2(r_0 - 38)(r_0 - 39). \quad (\text{III.32})$$

Indeed, let m_{ext} be the number of exterior neighbors, so $d_v = r_0 + m_{\text{ext}}$. Then $\mathbb{E}(a+b) = 2m_{\text{ext}}/79$ and $\mathbb{E}(ab) = m_{\text{ext}}(m_{\text{ext}} - 1)/(79 \cdot 78)$. The expectation of (III.32) is

$$2m_{\text{ext}}(m_{\text{ext}} - 1) + 2(2r_0 - 76)m_{\text{ext}} + 2(r_0 - 38)(r_0 - 39) = 2(d_v - 38)(d_v - 39).$$

The rows are homogeneous equalities. Probability normalization is stored separately and is not treated as another zero-right-hand-side equation.

The builder is `finite_profiles.py`. The auditor `audit_used_base_rows.py` reconstructs the required rows by labelled injections and weighted subset counts. For degree squares it uses the equivalent expression $2\binom{d}{2} - 76d + 1482$, confirming the polynomial through a different expansion.

III.15 Mixed common-neighborhood cells

Let $C, D \subseteq V(X)$ be disjoint vertex sets, where C is a clique and D is independent. We seek vertices adjacent to every member of C and nonadjacent to every member of D . Their set, called a *mixed cell*, is

$$\mathcal{C}(C, D) = \{x \in V(X) \setminus (C \cup D) : x \sim c \ \forall c \in C, \quad x \not\sim d \ \forall d \in D\}.$$

Here $x \sim c$ means $\{x, c\} \in E(X)$, and $x \not\sim d$ means $\{x, d\} \notin E(X)$. Write $p = |C|$ and $h = |D|$, with $0 \leq p \leq 4$ and $0 \leq h \leq 5$. The induced cell graph $X[\mathcal{C}(C, D)]$ belongs to $\mathcal{R}(5 - p, 6 - h, |\mathcal{C}(C, D)|)$. A forbidden clique inside it extends with C , and a forbidden independent set extends with D .

The sizes of these cells are bounded by small Ramsey numbers. The following table records the maximum allowed size used for each displayed pair (p, h) ; the indices p and h are the two root-group sizes:

(p, h)	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(4,1)
$ \mathcal{C}(C, D) \leq$	24	17	8	3	13	8	5	4	3	0

Each entry is $\text{cap}(p, h) := R(5 - p, 6 - h) - 1$, using standard small Ramsey numbers or the immediate formulas for a parameter 1 or 2.

For a six-vertex flag containing the root groups, let r_0 count cell vertices already in its tail. For the exterior sample z , set

$$\chi(z) = \prod_{c \in C} a_{cz} \prod_{d \in D} (1 - a_{dz}).$$

The row contribution is

$$r_0 + 78\chi(z) - \text{cap}(p, h).$$

The roots are excluded from the cell. This is especially important if C is empty: an independent root is nonadjacent to itself under the zero diagonal convention, so omitting the exclusion would count roots incorrectly.

The builder is `mixed_cells.py`; `audit_used_exact_rows.py` reconstructs selected rows with ordered injections and group-preserving automorphism factors. It derives the capacities from the corresponding Ramsey bounds.

III.16 Cell integrality and binary degree products

For integers c, k ,

$$(c - k)(c - k - 1) \geq 0. \tag{III.33}$$

Here c is the actual integer size of a mixed cell and k is any chosen integer threshold. Condition on a five-vertex flag containing its roots. Let r_0 count cell vertices in that flag, let u, v be the two distinct exterior samples, and let $a = \mathbf{1}_{u \in \mathcal{C}(C, D)}$, $b = \mathbf{1}_{v \in \mathcal{C}(C, D)}$. Twice an unbiased estimate of (III.33) is

$$2 \cdot 79 \cdot 78ab + 2(r_0 - k)79(a + b) + 2(r_0 - k)(r_0 - k - 1).$$

The stored row is its negative, since every global inequality uses the nonpositive convention. The final certificate uses 131 rows of this type.

The auditor constructs estimates of c and $\binom{c}{2}$ by counting subsets and then applies

$$(c - k)(c - k - 1) = 2 \binom{c}{2} - 2kc + k(k + 1).$$

This checks the expanded formula and its sample factors without reusing the original expansion.

III.16.1 Products of degrees

Put $\varepsilon_v = d_v - 38 \in \{0, 1\}$. For two ordered roots x, y , the three quantities

$$\varepsilon_x \varepsilon_y, \quad (1 - \varepsilon_x)(1 - \varepsilon_y), \quad \varepsilon_x(1 - \varepsilon_y)$$

are nonnegative. Reversing the root order supplies the complementary mixed product.

Condition on a five-vertex flag containing the two ordered roots x, y . Let u, v be the two distinct exterior sample vertices. Put $r_x = |N_X(x) \cap W|$ and $r_y = |N_X(y) \cap W|$, where W is the flag vertex set, and define $x_u = a_{xu}$, $x_v = a_{xv}$, $y_u = a_{yu}$, $y_v = a_{yv}$. Twice the degree estimates are

$$D_x = 2r_x + 79(x_u + x_v), \quad D_y = 2r_y + 79(y_u + y_v).$$

Twice an unbiased estimate of $d_x d_y$ is

$$\begin{aligned} P_{xy} = & 2r_x r_y + 79(r_x(y_u + y_v) + r_y(x_u + x_v)) \\ & + 79(x_u y_u + x_v y_v) + 79 \cdot 78(x_u y_v + x_v y_u). \end{aligned} \quad (\text{III.34})$$

The terms in which the same sample serves both degrees have coefficient 79, not $79 \cdot 78$.

Define $H_x = D_x - 76$, $H_y = D_y - 76$, and

$$H_{xy} = P_{xy} - 38(D_x + D_y) + 2 \cdot 38^2.$$

Twice the three nonnegative products are estimated by H_{xy} , $2 - H_x - H_y + H_{xy}$, and $H_x - H_{xy}$. Their negatives give the stored rows; 69 occur in the final sum. The independent auditor enumerates ordered neighbor pairs, permits coincident positions, and chooses weights by the number of distinct exterior vertices.

III.17 Edge and profile inequalities inside cells

Let Y be the induced graph $X[\mathcal{C}(C, D)]$ when edges of X are being counted, and let $Y = \overline{X[\mathcal{C}(C, D)]}$ when nonedges are being counted. We call these the red and blue cases, respectively. Suppose rational constants α, β give a valid support $e(Y) \geq \alpha|Y| + \beta$ for the relevant family. Condition on a five-vertex flag with vertex set W containing the roots. Put $I = \mathcal{C}(C, D) \cap W$, $r_0 = |I|$, and let e_I count pairs from I that are edges in the selected color. For exterior vertices u, v , let χ_u, χ_v be cell indicators and

$$z_u = \chi_u \sum_{w \in I} \epsilon_{uw}, \quad z_v = \chi_v \sum_{w \in I} \epsilon_{vw},$$

where, for distinct u, w , $\epsilon_{uw} = a_{uw}$ in red and $\epsilon_{uw} = 1 - a_{uw}$ in blue. The indicators are $\chi_u = \mathbf{1}_{u \in \mathcal{C}(C, D)}$ and $\chi_v = \mathbf{1}_{v \in \mathcal{C}(C, D)}$. The following C_2 and E_2 are twice unbiased estimates of $|Y|$ and $e(Y)$, respectively:

$$C_2 = 2r_0 + 79(\chi_u + \chi_v), \quad (\text{III.35})$$

$$E_2 = 2e_I + 79(z_u + z_v) + 79 \cdot 78\chi_u \chi_v \epsilon_{uv}. \quad (\text{III.36})$$

The expression $\alpha C_2 + 2\beta - E_2$ has average $2(\alpha|Y| + \beta - e(Y)) \leq 0$, so summing it over flag placements produces a valid global inequality. In E_2 , the term $2e_I$ counts edges already inside the fixed flag; $79(z_u + z_v)$ accounts for edges with one endpoint outside it; and the final term accounts for edges with both endpoints outside it. These three cases explain every sampling factor in the formula.

The final sum uses two rows from `cell_edge_cuts.py`. The auditor checks that these selected supports concern the $(4, 4)$ family and verifies them on every supplied profile point. These two selected rows are the cell-edge supports used in the final weighted sum.

III.17.1 Full cell profiles

Let $\mathcal{A} \subseteq \mathbb{Q}^4$ be a finite set whose convex hull contains the profile $(|Y|, e(Y), t(Y), q(Y))$ of every graph Y in the cell family under consideration. Write (c, e, t, q) for coordinates in this four-dimensional space; thus c is the order of the cell graph. If fixed rational coefficients $\gamma_c, \gamma_e, \gamma_t, \gamma_q, \gamma_0$ satisfy

$$\gamma_c c + \gamma_e e + \gamma_t t + \gamma_q q + \gamma_0 \leq 0 \quad \text{for every } (c, e, t, q) \in \mathcal{A},$$

then the inequality holds on $\text{conv}(\mathcal{A})$ and hence on every actual cell profile. The points in $\mathcal{A}$ need not all be actual profiles. The saved integer normals are checked on every point. Any collection of valid supporting inequalities gives a containing region for the profiles.

The general cell-profile builder uses the following types:

$ C $	$ D $	Counted color	Profile family
2	0	red	(3, 6)
0	3	blue	(3, 5)
2	1	red	(3, 5)
1	2	red or blue	(4, 4)
2	2	red	(3, 4)
1	3	blue	(3, 4)

The blue profiles are those of the complement of the induced cell graph.

Condition on a four-vertex flag and use three exterior samples. Multiplying by 6 gives integer weights

$$6 \frac{\binom{80}{m}}{\binom{3}{m}} = 6, 160, 6320, 492960 \quad (m = 0, 1, 2, 3). \quad (\text{III.37})$$

For each coordinate, enumerate the corresponding subsets of the cell visible in the seven-set: single vertices for c , edges for e , and three-sets with their triangle or q contribution. Give a subset with m vertices outside the flag the corresponding weight in (III.37); give the constant coordinate weight 6. The resulting five-component vector estimates $6(c, e, t, q, 1)$ without bias. A three-set contributes 1 to t if it is a triangle, and contributes 1 or 2 to q if it is an induced path or triangle, respectively. Its scalar product with $(\gamma_c, \gamma_e, \gamma_t, \gamma_q, \gamma_0)$ estimates six times the valid inequality. Summing over flag placements as in (III.31) gives the row.

III.17.2 Small-order outer sets

The (3, 5) and (4, 4) point sets are built from supplied complete catalogues, together with directly enumerable lower orders. The (3, 4) set is extracted by filtering the relevant (3, 5) graphs. For (3, 6) below order 13, the builder uses the following explicit outer regions. For $1 \leq c \leq 12$, each integer edge count satisfies

$$\ell(c) \leq e \leq \min\{\lfloor 5c/2 \rfloor, \lfloor c^2/4 \rfloor\}, \quad \ell(c) = \begin{cases} 0 & c \leq 5, \\ c - 5 & 6 \leq c \leq 10, \\ 3c - 25 & 11 \leq c \leq 12. \end{cases}$$

The lower bounds are the $k = 5$ cases of [III.5, Theorem 2]; this is an additional established mathematical input. A neighborhood in a triangle-free graph is independent, so the independence restriction gives maximum degree 5; handshaking gives the first upper bound. Mantel's theorem gives the second upper bound. At order zero, the single profile is (0, 0, 0, 0).

Triangle-freeness gives $t = 0$ and $q = w$. For $c \geq 1$, define the Euclidean quotients and remainders by $2e = ck + r_0$ with $0 \leq r_0 < c$, and $2e = 5s_0 + r_1$ with $0 \leq r_1 < 5$. Then convexity of $\binom{d}{2}$ on integer degrees gives

$$c \binom{k}{2} + r_0 k \leq q \leq 10s_0 + \binom{r_1}{2}.$$

The lower bound balances the degrees as evenly as possible; the upper bound fills degrees to 5 before using a remainder. Graphicality is not assumed for these extremizing degree lists, so the interval is safely an outer bound. The builder adds both endpoints at each (c, e) and takes their hull. This constructs the containing region from degree bounds at these lower orders.

The packaged bridge audit regenerates the point arrays and links the saved arrays to these rules. The regenerated arrays therefore connect the checked facet coefficients to the stated catalogue and degree bounds.

III.18 Restricted edge cores and critical order 17

For an edge uv , its common neighborhood has at most 17 vertices. It has at least 13, since the neighbors of u other than v that are not adjacent to v form a $(4, 5)$ -graph and hence number at most 24:

$$|N(u) \cap N(v)| \geq d_u - 1 - 24 \geq 13.$$

If $d_u = 39$, this improves to 14. Thus complete $(3, 6)$ catalogues of orders 13 through 17 suffice for the edge core. Restricted profile hulls from these catalogues are stronger than a hull allowing all smaller orders. Nine such lifted rows occur in the final sum.

The files are `restricted_edge_core_points.npy`, `restricted_edge_core_facets.json`, and `restricted_edge_core_profile_cuts.npz`. The auditor verifies facets on points and reconstructs the lifts. It additionally checks `degree_core_defect_cuts.npz`, although none of those rows has a nonzero final weight.

III.18.1 A cubic inequality from the seven critical graphs

The seven graphs in $\mathcal{R}(3, 6, 17)$ have every degree in $\{4, 5\}$. For a triangle uvw with distinguished edge uv , define

$$c = |N(u) \cap N(v)|, \quad \tau = |N(u) \cap N(v) \cap N(w)|.$$

If $c = 17$, then τ is the degree of w in the critical edge core and belongs to $\{4, 5\}$. If $c \leq 16$, both $16 - c$ and the integer polynomial $(\tau - 4)(\tau - 5)$ are nonnegative. Hence

$$(16 - c)(\tau - 4)(\tau - 5) \geq 0. \tag{III.38}$$

The related linear inequality $\tau \geq 2c - 30$ follows from $\tau \geq 0$ when $c \leq 15$. When $c = 16$, consider the nonneighbors of w inside the edge core. They form a $(3, 5)$ -graph, so there are at most 13 of them; hence $16 - 1 - \tau \leq 13$ and $\tau \geq 2$. When $c = 17$, the catalogue gives $\tau \geq 4$. The four selected critical-core rows are cubic.

Condition on a four-vertex flag with vertex set W containing u, v, w . Let $\mathcal{O} = V(X) \setminus W$, and define

$$c_0 = |W \cap N(u) \cap N(v)|, \quad \tau_0 = |W \cap N(u) \cap N(v) \cap N(w)|.$$

For $i \in \mathcal{O}$, set $p_i = \mathbf{1}_{i \in N(u) \cap N(v)}$ and $q_i = \mathbf{1}_{i \in N(u) \cap N(v) \cap N(w)}$. Put $P = \sum_{i \in \mathcal{O}} p_i$ and $Q = \sum_{i \in \mathcal{O}} q_i$, so $c = c_0 + P$, $\tau = \tau_0 + Q$, and $q_i \leq p_i$. Every index in the following sums ranges over $\mathcal{O}$; pairs and triples are ordered and obey the displayed distinctness conditions. The overlap expansions are

$$\begin{aligned} Q^2 &= Q + \sum_{i \neq j} q_i q_j, & PQ &= Q + \sum_{i \neq j} p_i q_j, \\ PQ^2 &= Q + \sum_{i \neq j} p_i q_j + 2 \sum_{i \neq j} q_i q_j + \sum_{i, j, k \text{ distinct}} p_i q_j q_k. \end{aligned}$$

They separate all-equal, one-pair-equal, and all-distinct tuples. Assigning weights from (III.37) gives the exact cubic estimator. The auditor evaluates ordered products directly, including coincidences, to check the original expanded formula independently.

III.19 Lifting the four local edge bounds and five facets

At degrees 38 and 39, the local edge bounds are equivalent to

$$e(A_v) \leq 10d_v - 66, \quad e(B_v) \geq 1524 - 28d_v.$$

For a five-vertex flag rooted at v , let r_0 count neighbors in the flag and let x, y be exterior samples. Put

$$D_2 = 2r_0 + 79(a_{vx} + a_{vy}).$$

Using the preceding edge estimator for either cell, the lifted rows are

$$E_2(A_v) - 10D_2 + 132, \quad -E_2(B_v) - 28D_2 + 3048.$$

The underlying bounds are established by local dual certificates; this construction establishes their finite averaged coefficients.

For a full local facet, choose $Y = A_v$ or $Y = B_v$ as specified by that facet. Condition on a four-vertex flag rooted at v and take three exterior samples. Define its five-component feature vector by the same weighted subset procedure, counting the root's neighbors for the degree coordinate and subsets of Y for the remaining graph statistics. It is three times an unbiased estimate of

$$(1, d_v, e(Y), t(Y), q(Y)).$$

The integer sampling factors are

$$3 \frac{\binom{80}{m}}{\binom{3}{m}} = 3, 80, 3160, 246480.$$

The same degree coordinate d_v is used for both $Y = A_v$ and $Y = B_v$. The feature array `local_profile_features.npy` is contracted with coefficient vector $(\gamma_0, \gamma_d, \gamma_e, \gamma_t, \gamma_q)$.

The exact matrix builder performs the contraction in integer arithmetic. The packaged bridge audit checks that the five used coefficient tuples agree with the certified facet metadata.

The feature auditor reconstructs the array through the general cell estimator at scale 6, checks divisibility by 2, and compares with the scale-3 array. For the nonneighborhood case, it still uses the red root-degree coordinate and only changes the edge, triangle, and q cell coordinates. Thus its degree feature represents d_v on both sides.

III.20 The exact finite Gram identity

The preceding sections obtained inequalities from degrees, cell sizes, and local profiles. We now add inequalities of another elementary kind: the square of any real expression is nonnegative. The expressions to be squared are linear combinations of counts of small configurations extending fixed roots. Expanding a square introduces pairs of such configurations. Their union has at most seven vertices, so its contribution can be expressed in the seven-vertex probabilities.

The matrices below organize this expansion. A *Gram matrix* here is a sum of matrices ff^T formed from count vectors f . For a real coefficient vector z , the corresponding quadratic expression is a sum of squares $(z^T f)^2$ and is nonnegative. The following counting identity expresses the averaged matrix in this form. Fix nonnegative integers ℓ, k with $\ell + 2k \leq 7$. A root type σ is a graph on the ordered vertex set $\{1, \dots, \ell\}$. Let $\mathcal{F}_{\sigma,k}$ be the finite set of isomorphism classes of graphs on $\ell + k$ vertices that have neither K_5 nor an independent six-set, and whose first ℓ distinguished ordered vertices induce σ . Isomorphisms fix every distinguished root and may permute the k remaining vertices. Fix the saved ordering of these classes, with indices $i \in \{1, \dots, m_{\sigma,k}\}$, where $m_{\sigma,k} = |\mathcal{F}_{\sigma,k}|$.

For an ordered tuple $\rho = (v_1, \dots, v_\ell)$ of distinct vertices in a graph Y with neither K_5 nor an independent six-set, let $V(\rho) = \{v_1, \dots, v_\ell\}$. The notation $Y[\rho] = \sigma$ means that $v_i \mapsto i$ identifies the root adjacencies with σ . For $S \in \binom{V(Y) \setminus V(\rho)}{k}$, write $\text{type}_Y(\rho, S)$ for the index in $\mathcal{F}_{\sigma, k}$ of the induced graph on $V(\rho) \cup S$ with this ordered root tuple. For each such ρ in X , define

$$\begin{aligned} f_i(\rho) &= |\{S \in \binom{V(X) \setminus V(\rho)}{k} : \text{type}_X(\rho, S) = i\}|, \\ f(\rho) &= (f_1(\rho), \dots, f_{m_{\sigma, k}}(\rho))^T. \end{aligned}$$

For a fixed root tuple, $f_i(\rho)$ counts the ways to choose k additional vertices of type i . Consequently, $f_i(\rho)f_j(\rho)$ counts ordered pairs of choices of types i and j , including choices that share additional vertices. Gathering these products into a matrix and summing over root tuples gives

$$\sum_{\rho: X[\rho] = \sigma} f(\rho)f(\rho)^T \succeq 0,$$

because for every real vector z its contraction is a sum of squares

$$\sum_{\rho: X[\rho] = \sigma} \left(\sum_i z_i f_i(\rho) \right)^2 \geq 0.$$

To express this on seven-vertex graphs, consider roots and an ordered pair of unordered extension sets S, T . Their union with the roots has $\ell + |S \cup T|$ vertices. It occurs in exactly

$$\binom{84 - \ell - |S \cup T|}{7 - \ell - |S \cup T|}$$

seven-sets. Give it the reciprocal of this number as weight. For a representative $H \in \mathcal{H}$, let $\mathcal{D}_H$ consist of all triples (ρ, S, T) such that ρ is an ordered ℓ -tuple of distinct vertices of H inducing σ , and $S, T \in \binom{V(H) \setminus V(\rho)}{k}$. The pair (S, T) is ordered, and $S \cap T$ may have any size from 0 to k . Define the symmetric $m_{\sigma, k} \times m_{\sigma, k}$ matrix

$$M_H(i, j) = \sum_{(\rho, S, T) \in \mathcal{D}_H} \frac{\mathbf{1}_{\text{type}_H(\rho, S) = i} \mathbf{1}_{\text{type}_H(\rho, T) = j}}{\binom{84 - \ell - |S \cup T|}{7 - \ell - |S \cup T|}}. \quad (\text{III.39})$$

The denominator is positive because $\ell + |S \cup T| \leq 7$. This definition applies to any allowed seven-vertex graph and is invariant under relabelling, so $M_{X[W]}$ denotes the same matrix as the isomorphic representative in $\mathcal{H}$. Summing over all seven-sets of X counts each global pair exactly once:

$$\sum_{W \in \binom{V(X)}{7}} M_{X[W]} = \sum_{\rho: X[\rho] = \sigma} f(\rho)f(\rho)^T.$$

Dividing by $\binom{84}{7}$ proves

$$\sum_H p_H M_H \succeq 0. \quad (\text{III.40})$$

All overlap sizes, including $S = T$, occur in the product of the two configuration counts and hence in (III.40).

Individual matrices M_H need not be positive semidefinite. The statement is about their average under a distribution arising from an 84-vertex graph. This is why a small graph can have a positive coefficient in a row whose global average is necessarily nonpositive.

III.21 Integer blocks, square vectors, and overflow checks

A *block* is the matrix family belonging to one fixed choice of root type σ and extension size k . We use $(\ell, k) = (0, 3), (1, 3), (2, 2), (3, 2)$. There is one root type at each of orders 0 and 1, two at order 2 (an edge and a nonedge), and four at order 3 (zero, one, two, or three edges). Thus there are $1 + 1 + 2 + 4 = 8$ blocks. The dimensions in the table count the possible rooted extension types, so matrices from different blocks need not have the same size. Clearing denominators in (III.39) gives $I_H = LM_H$ with integer entries:

(ℓ, k)	Root mask	Dimension	L
(0, 3)	0	4	6654960
(1, 3)	0	20	246480
(2, 2)	0	20	246480
(2, 2)	1	20	246480
(3, 2)	0	72	6162
(3, 2)	1	72	6162
(3, 2)	3	72	6162
(3, 2)	7	71	6162

Here the dimension is $m_{\sigma,k}$, and L is the least common multiple of the overlap denominators for the indicated block; its value depends on (ℓ, k) . In the saved block names, the field **r** records the root order ℓ . For each saved integer vector $z \in \mathbb{Z}^{m_{\sigma,k}}$ belonging to this block, the row

$$Q_z(H) = -z^T I_H z$$

has nonpositive average by (III.40). Its entries can have either sign.

The file `rational_psd_vectors.json` stores block names, integer vectors, and positive normalization denominators. The file `rational_psd_numerators.npy` stores their rows. There are 448 saved vectors and 194 used vectors. The used vectors have 11048 nonzero entries in total.

The program `rebuild_integer_moments.py` enumerates ordered roots and every ordered pair of extension subsets, assigns rooted type indices from `moment_keys.json`, and adds the integer overlap weight. In check-only mode it compares the result with `moment_integer_verified.npz`.

One independent replay uses arbitrary-precision Python integers for every contraction. Another uses fixed-width arrays only after checking a sufficient bound. If entries of I_H are nonnegative and at most B , then

$$|z^T I_H z| \leq B \left(\sum_i |z_i| \right)^2.$$

The latter quantity must be below 2^{63} before signed 64-bit arithmetic is used. The final separation accumulation also uses exact integers. Both implementations evaluate the contractions in the Gram identity.

III.22 Row normalization and exact source linkage

At this stage each valid inequality has been converted into a list of 979 rational coefficients, one for each seven-vertex type H . We call such a list a *row*. Its statement is that the weighted average of its entries, with weights p_H , is nonpositive. A homogeneous equality row instead has weighted average zero. The word “homogeneous” means that the right-hand side is zero. Probability normalization has right-hand side one, so it must be kept separate.

The numerical files collect these rows in matrices. This section specifies their ordering and scales so that a reader can match the previously justified formulas to the precise rows multiplied by the final certificate. Use the fixed ordering of the 979 representatives $H \in \mathcal{H}$ as column indices.

The file `exact_linear_model.npz` stores an integer matrix U with 4876 rows and an integer matrix E with 90 rows, together with positive integer denominator vectors U_{den} and E_{den} . The indices in the data files are zero-based: $i \in \{0, \dots, 4875\}$ for U and $j \in \{0, \dots, 89\}$ for E . In each case the denominator is the maximum absolute entry in that row, with denominator 1 for a zero row. Positive scaling preserves a homogeneous equality or a nonpositive inequality.

The first 89 rows of E are homogeneous degree-square equalities. The last row, indexed by 89, is all ones with denominator 1 and expresses probability normalization, with right-hand side 1. Define the normalized homogeneous row functions only for $j < 89$ by

$$E_j(H) = \frac{E_{jH}}{E_{\text{den},j}} \quad (0 \leq j < 89).$$

The square data contain 448 ordered entries. For square index $a \in \{0, \dots, 447\}$, let z_a be its saved integer vector, let $Q_a(H) = -z_a^T I_H z_a$ use its specified block, and let $d_a > 0$ be its saved normalization denominator. This denominator is verified to equal $\max_H |Q_a(H)|$. Define all normalized inequality rows by

$$U_i(H) = \begin{cases} U_{iH}/U_{\text{den},i}, & 0 \leq i < 4876, \\ Q_{i-4876}(H)/d_{i-4876}, & 4876 \leq i < 5324. \end{cases}$$

The necessary global constraints, in addition to $p_H \geq 0$ and $\sum_H p_H = 1$, are precisely

$$\sum_{H \in \mathcal{H}} U_i(H) p_H \leq 0 \quad (0 \leq i < 5324), \quad \sum_{H \in \mathcal{H}} E_j(H) p_H = 0 \quad (0 \leq j < 89).$$

The final sum uses 87 homogeneous rows and excludes the normalization row from this zero-right-hand-side system.

The file `exact_linear_model_keys.json` maps each linear row to its source family and source index. For local profile rows, integer coefficients are reconstructed from the facet and feature arrays rather than rounded from a numerical matrix.

The program `audit_packaged_model.py` checks the complete source linkage of all 4876 linear rows and all 89 homogeneous equalities, their normalization denominators, and the five final facet tuples. It also regenerates the cell-profile point arrays from their declared catalogue and outer-bound sources. The resulting point counts are 42, 215, 611, 3638 for families (3, 4), (3, 5), (3, 6), (4, 4) respectively, and 358 for the restricted edge-core set.

This check identifies the rows used in the final weighted sum with the rows reconstructed from the counting formulas.

III.23 The separating sum and its strict margin

The file `exact_separation_certificate.json` supplies the final equation and inequality multipliers. They are rational numbers with common positive denominator $W = 10^{12}$. Complete its two lists with zero values at unlisted indices, thereby defining

$$\lambda_i \in \mathbb{Q}_{\geq 0} \quad (0 \leq i < 5324), \quad \mu_j \in \mathbb{Q} \quad (0 \leq j < 89).$$

The supports $I_* := \{i : \lambda_i \neq 0\}$ and $J_* := \{j : \mu_j \neq 0\}$ have sizes 816 and 87, respectively. The total number of nonzero multipliers is 903. Put

$$C(H) = \sum_{i=0}^{5323} \lambda_i U_i(H) + \sum_{j=0}^{88} \mu_j E_j(H).$$

The average of C under any valid probability vector $(p_H)_{H \in \mathcal{H}}$ is nonpositive. This follows by multiplying each inequality by its nonnegative λ_i , each equality by its signed μ_j , and adding.

Equality multipliers may have either sign because an equality remains true when multiplied by a negative number. Inequality multipliers must be nonnegative to preserve their direction.

The remaining finite calculation evaluates $C(H)$ separately for every one of the 979 columns. It uses the saved rational multipliers and the row coefficients already linked to the counting formulas. The aim is to prove that every one of these 979 numbers is positive. If this succeeds, any nonnegative weighted average with total weight one would also be positive, contradicting the previous paragraph. Thus no search over unknown probability vectors is required during verification.

For a precise description of the integer lower-bound calculation, form the disjoint term index set

$$\mathcal{K} = (\{\text{ineq}\} \times I_*) \cup (\{\text{eq}\} \times J_*).$$

For a term $k \in \mathcal{K}$, denote its integer multiplier numerator by w_k , its integer row numerator at H by a_{kH} , and its positive integer row denominator by d_k . These are obtained from the corresponding linear row or square row just defined. Thus its contribution to $C(H)$ is $w_k a_{kH} / (W d_k)$. With the additional positive scale $F = 10^{12}$,

$$\frac{w_k a_{kH}}{W d_k} \geq \frac{\lfloor F w_k a_{kH} / d_k \rfloor}{W F}.$$

Floor division is required also for negative numerators. Truncating toward zero would give an upper rather than a lower approximation for negative terms.

For each $H \in \mathcal{H}$, compute the integer

$$L_H = \sum_{k \in \mathcal{K}} \left\lfloor \frac{F w_k a_{kH}}{d_k} \right\rfloor, \quad C(H) \geq \frac{L_H}{W F}.$$

The minimum is

$$\min_H L_H = 3146981978575629437 > 0. \quad (\text{III.41})$$

Thus $C(H) \geq \delta$ for every H , where

$$\delta = \frac{3146981978575629437}{10^{24}} > 0.$$

The weighted constraints give the first inequality below, the columnwise positive bounds give the second, and probability normalization gives the equality:

$$0 \geq \sum_H p_H C(H) \geq \delta \sum_H p_H = \delta > 0.$$

This proves $\mathcal{R}(5, 6, 84) = \emptyset$, subject to the stated external inputs.

Any strictly positive lower bound on every coefficient yields this contradiction. Downward rounding preserves the required lower-bound direction.

III.24 Which rows have nonzero final weights

The following table counts the rows with nonzero weight in the final certificate.

Inequality family	Used rows
Base degree and capacity rows	315
Mixed-cell capacities	29
Cell edge supports	2
Cell integrality	131
Binary degree products	69
General cell profiles	31
Restricted edge-core profiles	9
Critical-core cubic inequalities	4
Neighborhood edge bounds	9
Neighborhood profile inequalities	23
Integer-vector squares	194
Total	816

There are also 87 homogeneous equality terms. These counts provide useful checks of the selected-row interface.

The 23 neighborhood profile rows arise by conditioning the five local facets on different rooted flag types. The stored degree-core defect rows are audited and have zero weight in the final sum.

Several auditors check a larger collection containing all rows selected by the final certificate. Their reports specify this coverage, so their row counts may exceed those in the table.

The final JSON and row-key arrays determine the rows used in the proof. Some auditors read `phase1_duals.npz` or `phase1_used_rows.json` to select a larger set of rows to check. These files are runtime dependencies; their role is to select checks, and their numerical weights are not premises of the final weighted sum.

III.25 Catalogue extraction and its precise trust boundary

A graph6 catalogue stores graphs; a `.profiles` file stores distinct tuples (n, e, t, q) . The local model uses the set of possible profiles. Graphs sharing a profile therefore contribute a single tuple. The header separately records the total graph-record count.

The program `catalogue_profiles.cpp` decodes every record into adjacency bitsets. It counts edges while decoding, computes $w = \sum_v \binom{d_v}{2}$, and computes triangles from

$$6t = \sum_{(u,v):uv \in E}^{\text{ordered}} |N(u) \cap N(v)|.$$

Each triangle contributes six times. The program checks divisibility by six, stores $(n, e, t, w - t)$ in an ordered set, and writes all distinct tuples.

The catalogue audit programs count every record but independently decode only a sample of roughly 100 profiles per catalogue. The closure audit verifies exact hull containment of all supplied profile points. Full extraction from the raw graph records is checked by the following separate program.

The program `regenerate_catalogue_profiles.py` compiles the supplied extractor in a temporary directory, copies raw catalogues there, runs the extractor on all records, and compares complete distinct profile sets and graph counts with the supplied files. Header source paths are not mathematical data and are ignored in the comparison. The collection contains 58 catalogue files and 11400325 graph records.

This exhaustively replays the supplied extractor. The sampled NetworkX decoder uses a separate implementation on its selected records. Completeness of the raw catalogue families remains the published external input.

The logical stages are therefore separate: external catalogue completeness; reproducible decoding and profile extraction; exact hull and recursive support verification; exact global coefficient construction; and exact final separation. The extraction replay uses the supplied extractor; the sampled decoder and the specified coefficient auditors provide distinct implementations of their respective calculations.

III.26 Environment and safe execution

The pinned Python requirements are NumPy 2.3.5, SciPy 1.16.3, and NetworkX 3.6. The tested Python version is 3.13.9. The intended reproduction environment is a fresh Python 3.13 virtual environment with these package versions. NumPy 2.x is relevant because the seven-graph auditor uses `bitwise_count`.

From the extracted verification tree, installation is typically

```
python3.13 -m venv .venv
. .venv/bin/activate
python -m pip install -r proof_84/requirements.txt
```

The package README specifies the outer layout and platform variants. Installing packages may require network access; the verification itself uses the supplied data and does not need a solver or online service.

SciPy supplies sparse-matrix operations and hull helpers. The verification commands execute exact checks without invoking an optimization solver.

Full profile regeneration additionally requires a C++11 compiler and zlib development support. On Debian-like systems these are normally provided by compiler and zlib development packages; on macOS a command-line developer toolchain supplies a suitable compiler and system zlib. The wrapper uses a temporary workspace and compares data without rewriting the archived originals.

Assertions must be active. Do not run with Python's `-O` option and do not set `PYTHONOPTIMIZE`. The supplied programs use assertions for checks; optimized execution disables them. The packaged launcher guards against this condition. A run with assertions disabled is not a verification run.

Use a fresh writable extraction. The driver and auditors create logs and reports. Set `PYTHONDONTWRITEBYTECODE=1` if desired to avoid bytecode caches, but this does not suppress intended reports. Preserve the original archive for comparison rather than expecting the working copy to remain byte-for-byte unchanged after execution.

III.27 Execution recipes and expected output

The commands below begin from a folder containing the complete file `verification.tar.xz`. This is the inner verification archive, not the outer arXiv upload archive. A reader who obtained the full arXiv source upload finds it under `anc`; a reader who obtained the separate verification distribution already has this file. Extract it and enter the folder it creates:

```
tar -xJf verification.tar.xz
cd verification
```

The terminal's current folder is now the extracted folder named `verification`. It contains `verify_package.py`, `README.txt`, `input_manifest.json`, and the subfolders `proof_84`, `lp_research`, `literature_research`, `root_research`, and `structural_research`. Every remaining command in this section is run from that folder. The commands neither assume nor use the author's research workspace.

If the compact archive was chosen. Its inner archive has the same name but omits the 58 raw public catalogue files. Restore them before verification by following its `START_HERE.txt`. For example,

```
python3 restore_catalogues.py --download
```

uses the network to retrieve the named catalogues and checks each against its recorded SHA256 digest. Alternatively, `-from-archive` reads them from the complete verification archive. After restoration, the compact route checks the same data by the same programs; it skips no mathematical test. This restoration step is unnecessary for the complete archive.

Install the Python environment once. With Python 3.13 installed, run the following macOS/Linux commands from the extracted `verification` folder:

```
python3.13 -m venv .venv
. .venv/bin/activate
python -m pip install -r proof_84/requirements.txt
```

The last command installs the exact package versions used by the supplied verifier and needs network access unless local installation files are available. Subsequent commands use `python` from this activated environment. A reader who opens a new terminal later must return to the same extracted folder and activate `.venv` again. Windows activation instructions are in `README.txt`.

Start with one interpretable calculation. To reproduce the worked 314-edge example after installing the dependencies, run

```
python check_local_example.py --root .
```

The dot means the extracted `verification` folder, which is the current folder. The checker evaluates the explicitly stated local inequalities on the required profile points and compares the resulting weighted identity with the saved certificate. The worked-method document explains those inequalities and coefficients on paper. This command is a small entry point into the argument; it does not replace the checks of the prerequisite containing polygons or the final seven-vertex contradiction.

Replay the complete certificate. Run

```
python verify_package.py
```

A successful run must exit normally and print a final summary with `"all_passed": true`. A partial success message followed by an error is not a successful run. The launcher first checks input hashes, then runs the main proof checks, the additional source-linkage check, and the worked-example calculation. Its reports identify which steps succeeded. No optimization solver and no network connection are needed for this replay.

The package-level launcher is `verify_package.py`. The package README documents its supported modes and their exact command lines. Its default mode checks package integrity, runs the 18 jobs in `verify_all.py`, performs the source-linkage audit, and runs the worked-example checker. The `-full` mode also regenerates all catalogue profiles, runs the three additional finite-estimator audits, and checks the separate degree-24 strengthening. The finite-estimator tests exhaust samples on their specified test graphs or populations; they test the estimator formulas on those specified finite populations. After installing the C++11 compiler and zlib support described in the preceding section, request the additional extraction and supplementary checks with

```
python verify_package.py --full
```

The commands below isolate parts of that full run for diagnosis or focused review. They still start from the same extracted `verification` folder with the environment active. The 18-job verification suite can be run directly:

```
python proof_84/verify_all.py
```

It launches 18 jobs with at most three running in parallel. A successful summary has `all_passed: true` and `check_count: 18`. It records `solver_used: false` and `network_used: false`. The following sections describe the jobs executed by this command.

For the local chain alone, run

```
python literature_research/replay_required_local_certificates.py
```

It must verify 2679 certificates: 1968 support bounds, 10 endpoints/exclusions, and 701 facet cases. The corresponding report is `required_local_certificate_audit.json`.

After the input rows have been justified by the preceding mathematical arguments and row checks, the final 979 column sums can be replayed separately with

```
python root_research/verify_exact_separation.py --replay
```

It must report strict separation and the numerator in (III.41). This command by itself does not prove validity of its row inputs; those are established by the local chain, row audits, and mathematical derivations.

The explicit package bridge and complete extraction checks can be run separately:

```
python audit_packaged_model.py
```

```
python regenerate_catalogue_profiles.py
```

If a check fails, retain the log and investigate rather than changing tolerances or bypassing assertions. Missing files, inconsistent row orders, changed package versions, stale data, or a genuine mathematical or programming error must be distinguished. Previous successful reports do not override a reproducible failure on the submitted data.

III.28 The 18-job suite: local and catalogue checks

This section and the next two describe all jobs launched by `verify_all.py`, grouped by the part of the proof they check.

III.28.1 Required local dual replay

`replay_required_local_certificates.py` derives the required claims from support directions, edge intervals, and selected facets. It reconstructs each model and objective, parses rational duals, forms integer residuals, and checks strict rounding thresholds. It compares the reconstructed objective with the saved objective. The report contains a hash for every required certificate.

III.28.2 Recursive polygons

`audit_recursive_polygons.py` reconstructs all 149 polygons. It checks exact intersections of the 123 support systems and exact hulls for 26 catalogue edge classes. Support validity is established by the dual replay, not by this geometric comparison alone.

III.28.3 Dependency closure

`audit_dependency_closure.py` compares the model's selected catalogue hull vertices with exact hulls of supplied profile points. Its current scope is 58 profile files, 331 edge-class hulls, and 7624 distinct profile entries across files. It also verifies full degree ranges, edge-bin coverage, row counts, and the version 3 restriction of the version 4 model.

III.28.4 Small catalogue records

`audit_catalogues.py` checks record counts against saved source-page information, computes hashes, and independently decodes sampled records. Its sampled profile checks are not exhaustive recomputation. The additional extraction wrapper described above supplies the latter replay.

III.28.5 The $(4, 5)$ catalogues

`audit_r45_catalogues.py` checks the order-24 catalogue of 352366 graphs and the selected complete extreme-edge classes of orders 19 through 23. It checks all record counts and samples decoded profiles. The expected counts for extreme classes are recorded in supplied headers; mathematical completeness remains external.

Together, these jobs check the catalogue records, their profile hulls, and the local dual inequalities that depend on them.

III.29 The 18-job suite: global coefficients

III.29.1 Seven-vertex completeness

`audit_seven_graph_completeness.py` enumerates all 2^{21} labelled graphs and compares allowed counts with exact orbit sizes of the saved representatives. It also checks pairwise nonisomorphism. This is a full finite enumeration, unlike the extraction sampling just described.

III.29.2 Base rows

`audit_used_base_rows.py` reconstructs degree-square equalities and base capacity inequalities by labelled injections and finite subset weights. It checks every final-used base row and additional stored rows.

III.29.3 Mixed cells and general profiles

`audit_used_exact_rows.py` independently reconstructs the selected mixed capacities and general cell-profile rows. It checks the Ramsey family corresponding to each color, tests each selected facet on all stored points, and uses group-preserving automorphism factors.

III.29.4 Integrality and binary degrees

`audit_used_integer_rows.py` reconstructs cell-integrality and binary-degree rows by direct subset and ordered-pair enumeration. In degree products, coincident neighbor choices are allowed and weighted according to the number of distinct exterior vertices.

III.29.5 Critical cores

`audit_used_core17_rows.py` reconstructs selected cubic rows by enumerating products in the edge and triangle cores, including repeated positions. It checks coefficients on every seven-vertex representative. The underlying catalogue degree fact supplies the sign of the cubic polynomial.

III.29.6 Cell edge supports

`audit_used_cell_edge_rows.py` validates the selected supporting lines on the $(4, 4)$ point set and reconstructs their lifted coefficients.

III.29.7 Neighborhood edges

`audit_used_neighborhood_edge_rows.py` reconstructs the finite lifts of the four local edge bounds. The local dual chain supplies their mathematical validity; this job checks finite sampling and multiplicity.

III.29.8 Neighborhood profile features

`audit_root_local_features.py` recomputes the complete local feature array via the general cell estimator at twice the stored scale, checks divisibility, and compares entries. The packaged bridge then checks its integer contraction with the selected five facets into the exact global matrix.

III.29.9 Restricted edge cores

`audit_restricted_core_rows.py` checks all restricted profile rows and additional degree-core defect rows. It tests inequalities on restricted profile points and independently reconstructs averages. Some checked rows have zero final multiplier.

III.30 The 18-job suite: Gram matrices and final sum

III.30.1 Integer Gram reconstruction

`rebuild_integer_moments.py` with `-check-only` rebuilds all eight integer Gram blocks by explicit overlap counting and compares them with `moment_integer_verified.npz`. It writes an audit report but does not replace the checked matrix file in this mode.

III.30.2 Direct separation replay

`verify_exact_separation.py` with `-replay` checks multiplier signs and unique indices, verifies named file hashes, recomputes square rows with a checked overflow bound, and accumulates the exact floor lower bounds. It explicitly separates probability normalization from homogeneous equalities.

III.30.3 Arbitrary-precision replay

`replay_global_certificate.py` recomputes the square contractions with Python integers and reproduces the final sum by arbitrary-precision arithmetic. This avoids reliance exclusively on fixed-width array contractions for the critical numerical part.

III.30.4 A further global auditor

`audit_global_exact_certificate.py` provides another exact replay and coefficient consistency check. Where fixed-width contractions are used, a sufficient no-overflow bound is checked first. The final accumulated separation is exact.

These four jobs, the nine global coefficient jobs, and the five local/catalogue jobs make the 18 jobs in `verify_all.py`. The package launcher also offers the source-linkage audit, exhaustive extraction replay, finite-estimator checks, and the optional strengthening checks described above.

Scope of independence. The auditors use different enumerations for important formulas, but share some data, conventions, and helper functions. Here an independent check means a distinct implementation of the specified calculation. Reimplementing the displayed formulas in another language and comparing exact outputs would provide another useful level of review.

III.31 Writing a verifier from this specification

A fresh implementation can be divided into five components. This keeps the trusted mathematical code understandable even though it processes many files.

First, decode graph6 into symmetric zero-diagonal adjacency matrices and recompute profiles. Preserve the difference between graph-record counts and distinct-profile counts. Gzip compression affects storage only.

Second, generate the local variables and equations from the profile sets. Use a declared order or named coordinates with an explicit translation to certificate positions. Check all degree and edge ranges before building the matrix. Missing input must cause an error.

Third, implement the rational certificate kernel:

for each required claim:

```
M, b, active = construct_model(claim.model)
c = construct_objective(claim)
y = parse_rationals(certificate.dual)
assert c == parse_rationals(certificate.objective)
for j in active:
    assert c[j] - sum(M[i,j]*y[i] for i) >= 0
lower = sum(y[i]*b[i] for i)
assert lower == certificate.exact_lower_bound
assert lower > claim.rounding_threshold
```

Use strict thresholds appropriate to the integer conclusion, not rounded decimal values. Equality rows may receive signed multipliers, whereas nonnegative slack coefficients must have the stated sign.

Fourth, enumerate rooted flag placements on all 979 graph types and reconstruct each used global coefficient by finite sampling. Validate the underlying inequality independently of its lift. This links each reconstructed row to the counting statement that gives its sign.

Fifth, enumerate Gram extension pairs with overlaps, contract integer vectors, and form the final rational sum or conservative floor lower bound. Check positivity for all columns and apply normalization only in the final contradiction.

These components use the supplied rational multipliers. Graph canonicalization, hull calculation, and sparse storage can be implemented differently, provided the resulting coefficients agree exactly after matching graph types, variables, and rows.

III.32 Small arithmetic checks and common errors

Complement conventions. For K_3 , $(n, e, t, q) = (3, 3, 1, 2)$. Equations (III.1)–(III.3) give the empty graph profile $(3, 0, 0, 0)$. For P_3 , the profile $(3, 2, 0, 1)$ becomes $(3, 1, 0, 0)$, as expected.

Four-vertex paths. For P_4 , $e = 3, t = 0, w = q = 2$. Equation (III.22) gives triple counts $(0, 2, 2, 0)$. In (III.28), the middle edge contributes 1 and the two end edges contribute zero. For a triangle, the sum over middle edges gives 3, and the correction $3t = 3$ removes every coincident outer-endpoint choice.

A degree-square coefficient. If a five-vertex flag has $r_0 = 2$ and the actual degree is 38, the exterior neighbor count is $m_{\text{ext}} = 36$. The expected value of (III.32) is

$$2 \cdot 36 \cdot 35 + 2(4 - 76)36 + 2(-36)(-37) = 0.$$

This checks both the linear coefficient and the constant in a concrete case.

Two and three exterior vertices. A specified edge outside a five-vertex flag appears with probability $1/\binom{79}{2}$. The factor $79 \cdot 78$ estimates twice its count. A specified triple outside a four-vertex flag appears with probability $1/\binom{80}{3}$; the factor $80 \cdot 79 \cdot 78$ estimates six times its count.

Negative floor division. For a term $-1/3$ and scale 10, use $\lfloor -10/3 \rfloor = -4$, giving a valid lower bound -0.4 . Truncating toward zero gives -0.3 , which is larger than $-1/3$ and therefore invalid as a lower bound.

Signs and colors. Every global inequality row uses nonpositive average. A nonnegative square is therefore negated. A support $e \geq ac + b$ becomes $ac + b - e \leq 0$. For blue cells, all internal edge-based statistics refer to the complement consistently, while a root's degree parameter in a neighborhood facet still refers to X .

Order and degree. A nonneighborhood B_v has order $83 - d_v$. Substituting that order into a coefficient intended for d_v changes the inequality. A root that is excluded from a cell remains excluded even if its diagonal adjacency entry happens to satisfy the cell's zero-one tests.

III.33 What a successful reproduction establishes

A complete reproduction establishes a chain of different statements:

1. The external Ramsey bounds and catalogue completeness assumptions are precisely identified.
2. Supplied graph and profile data are linked by counts, hashes, and the specified extraction checks.
3. Recursive polygons contain all justified profiles, with no omitted degree or edge class.
4. Every required local dual has the correct exact residual signs and strict threshold.
5. Every used global row is the correct finite average of a valid underlying inequality or identity.
6. The seven-vertex type set is complete and consistently ordered.
7. Every square row comes from the exact finite Gram identity with all overlaps.
8. The final signed and nonnegative multipliers produce a strictly positive coefficient on every allowed type.

The resulting contradiction proves $R(5, 6) \leq 84$ by exact rational arithmetic.

The argument combines the stated external catalogue and Ramsey results, the mathematical derivations of the counting identities, and the exact finite checks performed by the supplied programs. Reproduction relies on the correctness of the software and arithmetic environment used for those checks.

The separate strengthening that every $X \in \mathcal{R}(5, 5, 45)$ has $e(X) \geq 462$ discussed in the manuscript is not needed for the certificate. The proof above uses the bound 460. Its optional audit verifies that separate statement. The coefficients of the final model are those specified above.

The package supports four levels of review: understanding the reasoning, checking a worked local example, replaying supplied programs, and independently reconstructing the calculations from this specification. The mathematical derivations establish the validity of the rows; the executable checks verify their numerical coefficients and weighted sums.

References

- [III.1] S. P. Radziszowski, *Small Ramsey Numbers*, Electronic Journal of Combinatorics, Dynamic Survey DS1, revision 18, 24 April 2026. <https://doi.org/10.37236/21>. See Table Ib for $R(4, 6) \leq 40$.
- [III.2] V. Angeltveit and B. D. McKay, $R(5, 5) \leq 46$, Journal of Graph Theory **112** (2026), 198–208. <https://doi.org/10.1002/jgt.70029>; <https://arxiv.org/html/2409.15709v2>. See Theorem 1.1 and Appendix A, Table 1.
- [III.3] B. D. McKay and S. P. Radziszowski, *Subgraph counting identities and Ramsey numbers*, Journal of Combinatorial Theory, Series B **69** (1997), 193–209. <https://users.cecs.anu.edu.au/~bdm/papers/r55.pdf>.
- [III.4] B. D. McKay, *Ramsey Graphs*, catalogue collection and descriptions. <https://users.cecs.anu.edu.au/~bdm/data/ramsey.html>.
- [III.5] J. Goedgebeur and S. P. Radziszowski, *New Computational Upper Bounds for Ramsey Numbers $R(3, k)$* , Electronic Journal of Combinatorics **20**(1) (2013), P30. <https://doi.org/10.37236/2824>. Theorem 2. Author manuscript: <https://www.cs.rit.edu/~spr/PUBL/e3k12.pdf>.
- [III.6] Electronic verification supplement to the accompanying manuscript (2026). Local mathematical input is documented in `LOCAL_CERTIFICATES.txt`; the 18-job verification driver is `verify_all.py`; the packaged launcher is `verify_package.py`; the final weighted sum is `exact_separation_certificate.json`. The package README specifies installation and the archived layout.