MATHDISCOVERIES
Mathematical Preprint

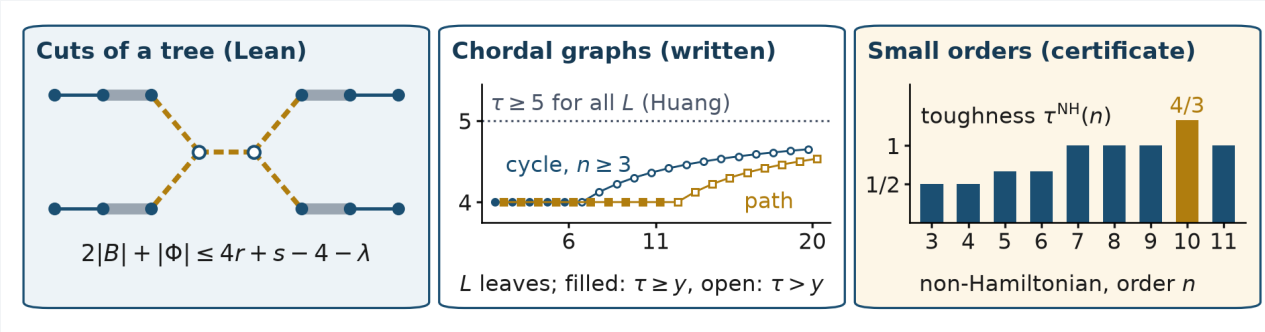
Working Draft

Toughness and Hamilton Cycles in Chordal Graphs of Bounded Leafage, in Two-Port Graphs, and in Graphs of Order at Most 11

Tyler B. Hughes*,†

*AI research agents built and directed by the author wrote this draft. †Independent Researcher, United States

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



ABSTRACT: Chvátal's open toughness conjecture says that for some t_0 every t_0-tough graph on at least three vertices is Hamiltonian. We prove partial results. A discharging inequality on trees, with a Hall theorem from a 2026 preprint of Huang, shows that a chordal graph on at least three vertices with a tree representation with L leaves is Hamiltonian if it is 4-tough and $L \le 6$, or if $L \ge 7$ and its toughness exceeds $5 - 7/L$, and has a Hamilton path if it is 4-tough and $L \le 11$. Among the non-Hamiltonian two-port graphs, which include those of Bauer, Broersma and Veldman, the least orders with toughness at least $3/2$, at least 2, and more than 2 are 25, 42 and 59. Their chordal graphs $G(M;\ell,m)$, $m \ge 2\ell + 1$, are fractionally Hamiltonian if and only if $m \le 4\ell$; hence for every $\varepsilon > 0$ some $(13/8 - \varepsilon)$-tough chordal graph is not fractionally Hamiltonian. The subtour linear program of Chvátal's split graphs of order n has optimum $n + 1/2$. The largest toughness of a non-Hamiltonian graph of order $n = 3, \dots, 11$ is $1/2, 1/2, 2/3, 2/3, 1, 1, 1, 4/3, 1$. Main results: 5 of 8 in Lean, 1 by certificate, 2 by written proof. AI research agents built and directed by Tyler B. Hughes produced and checked this work. No human mathematician has yet refereed the proof.

Verification Scope (G11): WARN: Human referee: pending
Main results: 5 of 8 in Lean Kernel (62%) · 1 Cert Only · 2 written
Abstract claims: 5 of 8 in Lean Kernel (62%) · 1 Cert Only · 2 written

Full Trust Boundary: Section 5.

MSC 2020: 05C45, 05C42, 05C85, 90C27, 68V20 **Keywords:** toughness, Hamilton cycles, chordal graphs, leafage, subtour elimination, Lean 4

1. INTRODUCTION AND MAIN RESULTS

A network that a tour can visit without repeating a point cannot be cut into many pieces by deleting few points: if a graph has a cycle through all its vertices, then deleting $s \ge 1$ vertices leaves at most s pieces. Toughness measures how far a graph is from being cut this way. Write $c(H)$ for the number of components of a graph H. A set S of vertices of G is *separating* if $c(G - S) \ge 2$, and G is t-*tough* if $|S| \ge t\, c(G - S)$ for every separating set S. The *toughness* $\tau(G)$ is the largest t for which G is t-tough; a complete graph has no separating set and $\tau = \infty$, and a disconnected graph has toughness 0, because the empty set separates it. Chvátal [5] conjectured that a constant t_0 exists such that every t_0-tough graph on at least three vertices is Hamiltonian. The conjecture is open. Bauer, Broersma and Veldman [1] constructed non-Hamiltonian graphs of toughness arbitrarily close to $9/4$. This paper does not settle the conjecture. It proves partial results for chordal graphs, for graphs built from gadgets with two ports, and for small orders.

A graph is *chordal* if every cycle of length at least four has a chord. Chen, Jacobson, Kézdy and Lehel [4] proved that 18-tough chordal graphs on at least three vertices are Hamiltonian, Kabela and Kaiser [9] lowered 18 to 10, and a 2026 preprint of Huang [8] proves that 5-tough chordal graphs on at least three vertices are Hamilton-connected. Non-Hamiltonian chordal graphs with toughness arbitrarily close to $7/4$ exist [1]. By Gavril's theorem [7] every chordal graph is the intersection graph of subtrees of a tree, a *tree representation*; the least number of leaves of such a host tree is the *leafage* of the graph [14]. A *branch vertex* of a tree is a vertex of degree at least 3.

Main results. The eight theorems below are proved in Section 3. Five are checked by the Lean 4 kernel, one rests on an exhaustive computation, and two have written proofs. Theorems 2 and 3 use Theorem 1.1 of the preprint [8], which as far as we know has not been refereed and whose proof is not reproduced here.

Theorem 1 (Discharging inequality; Lean). *Let T be a tree on at least two vertices with L leaves and b branch vertices. Let the* gray *edges be a matching of T whose edges have both endpoints of degree 2 in T. Let B be a nonempty set of edges of T and let Φ be a set of gray edges, each of which shares a vertex with an edge of B. Call a component of $T - B$* active *if it contains a vertex of degree at most 2 in T and* inactive *otherwise, let r and s be the numbers of active and inactive components, let λ be the number of pairs (v, e) with v a leaf of T, $e \in B$ and $v \in e$, and put $Q = 2|B| + |\Phi|$. Then $r + s = |B| + 1$, $r \geq 2$, $s \leq r - 2$, $s \leq b \leq L - 2$, and*

$$Q \leq 4r + s - 4 - \lambda, \qquad Q \leq 5r - 6,$$
$$Q \leq 4r + b - 4, \qquad Q \leq 4r + L - 6.$$

Theorem 2 (Chordal graphs of bounded leafage; written proof). *Let G be a chordal graph on at least three vertices that has a tree representation whose host tree has L leaves and b branch vertices.*

(*i*) *If G is 4-tough and at least one of $L \leq 6$, $b \leq 4$ holds, then G is Hamiltonian.*
(*ii*) *If $L \geq 7$ and $\tau(G) > 5 - 7/L$, then G is Hamiltonian.*
(*iii*) *If G is 4-tough and at least one of $L \leq 4$, $b \leq 2$ holds, then G is Hamilton-connected.*
(*iv*) *If $L \geq 5$ and $\tau(G) > 5 - 5/L$, then G is Hamilton-connected.*

Theorem 3 (Hamilton paths; written proof). *Let G be a chordal graph with at least one vertex that has a tree representation whose host tree has L leaves. If G is 4-tough and $L \leq 11$, then G has a Hamilton path. If $L \geq 12$ and $\tau(G) > 5 - 7/(L - 5)$, then G has a Hamilton path.*

For leafage at most three the cycle conclusion is known in a stronger form: every 3/2-tough spider graph on at least three vertices is Hamiltonian [10], and so is every 1-tough interval graph on at least three vertices [11]. So parts (i) and (ii) of Theorem 2 have content from leafage four on; for the Hamilton paths between prescribed ends in parts (iii) and (iv), interval graphs (leafage at most two) of toughness greater than 1 are Hamilton-connected, by the characterization of Broersma et al. [2] through the scattering number; at leafage three we know of no such comparison. The constants of Theorem 2 are not shown to be best possible: the non-Hamiltonian chordal graphs of small leafage that we know have toughness well below 4 (Proposition 14).

The second group concerns the construction of [1]. A *two-port graph* with parameters $\ell \geq 1$ and $m \geq 2\ell + 1$ is a graph G whose vertex set is a disjoint union $K \sqcup C \sqcup W_1 \sqcup \cdots \sqcup W_m$ with $|K| = \ell$, $|C| \geq 2$ and every W_i nonempty, such that no edge joins two different sets W_i, the neighbors of W_i in C lie in a set of two distinct *ports* $u_i, v_i \in C$, and the *gadget* H_i, obtained from $G[W_i \cup \{u_i, v_i\}]$ by deleting the edge $u_i v_i$ if it is present, has no Hamilton path from u_i to v_i. Edges inside $K \cup C$ and between K and the sets W_i are arbitrary, different gadgets may share ports, and their orders are unrestricted. Let M and L be the graphs of Figure 2, with ports 0 and 1. For $J \in \{M, L\}$ the graph $G(J; \ell, m)$ is obtained from m disjoint copies of J and a clique K on ℓ vertices by joining every vertex of K to all other vertices and making the $2m$ ports pairwise adjacent.

Theorem 4 (Orders of two-port graphs; Lean). *Every two-port graph G with $\ell \geq 1$ and $m \geq 2\ell + 1$ is non-Hamiltonian. If G is 3/2-tough, then $|V(G)| \geq 25$; if G is 2-tough, then $|V(G)| \geq 42$; if G is t-tough for some rational $t > 2$, then $|V(G)| \geq 59$. The graphs $G(L; 1, 3)$, $G(L; 2, 5)$ and $G(L; 3, 7)$ are non-Hamiltonian, have orders 25, 42 and 59, and are 3/2-tough, 2-tough and 31/15-tough.*

A *fractional Hamilton cycle* of G is a weighting $x \colon E(G) \to [0, 1]$ with total weight 2 at every vertex and weight at least 2 on the cut $\partial(U)$ for every nonempty proper subset U of $V(G)$; G is *fractionally Hamiltonian* if one exists [17, 18]. Dropping the bound $x \leq 1$ and the degree conditions leaves the *subtour program*: minimize $\sum_e x_e$ subject to $x \geq 0$ and $x(\partial(U)) \geq 2$ for all nonempty proper U. Scheinerman and Ullman conjectured that every 2-tough graph on at least three vertices is fractionally Hamiltonian, and Wang [18] proved it for 10-tough graphs. By the examples of Scheinerman and Ullman no constant below 3/2 suffices [18].

Theorem 5 (The chordal graphs $G(M; \ell, m)$; Lean). *Let $\ell \geq 1$ and $m \geq 0$. The graph $G(M; \ell, m)$ has $\ell + 7m$ vertices and is chordal, and every feasible point of its subtour program has total weight at least $29m/4$. If $m \geq 2\ell + 1$, then $G(M; \ell, m)$ is non-Hamiltonian, and it is fractionally Hamiltonian if and only if $m \leq 4\ell$.*

Theorem 6 (Chordal and not fractionally Hamiltonian; Lean). *For every rational $\varepsilon > 0$ there are $\ell \geq 2$ and $m = 4\ell + 1$ such that $G(M; \ell, m)$ is chordal, $(13/8 - \varepsilon)$-tough, and not fractionally Hamiltonian.*

Thus toughness below 13/8 does not force a fractional Hamilton cycle, not even in a chordal graph. The next theorem concerns the split graphs of Chvátal [5]: for $\ell \geq 1$ let Sp_ℓ consist of a clique on the disjoint union of A and R, $|A| = \ell$, $|R| = 2\ell + 1$, and an independent set $\{i_r : r \in R\}$ in which i_r is adjacent to r and to all of A.

Theorem 7 (Split graphs; Lean). *For every $\ell \geq 1$ the graph Sp_ℓ has $n = 5\ell + 2$ vertices, is non-Hamiltonian, and has toughness exactly $3\ell/(2\ell + 1)$. Some feasible point of its subtour program with $x \leq 1$ has total weight $n + 1/2$, and every feasible point of its subtour program has total weight at least $n + 1/2$. In particular Sp_ℓ is not fractionally Hamiltonian.*

Theorem 8 (Small orders; certificate). *For $n \geq 3$ let $\tau^{\mathrm{NH}}(n)$ be the largest toughness of a non-Hamiltonian graph of order n. Then*

n	3	4	5	6	7	8	9	10	11
$\tau^{\mathrm{NH}}(n)$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	1	1	1	$\frac{4}{3}$	1

The Petersen graph is the only non-Hamiltonian graph of order between 3 and 11 with toughness greater than 1.

In particular τ^{NH} is not monotone: $\tau^{\mathrm{NH}}(11) < \tau^{\mathrm{NH}}(10)$.

An illustration. The box builds the Petersen graph on the 2-subsets of $\{0, \ldots, 4\}$, two of them adjacent when disjoint, computes its toughness from all sets of kept vertices and looks for a Hamilton cycle. It prints `4/3 False`.

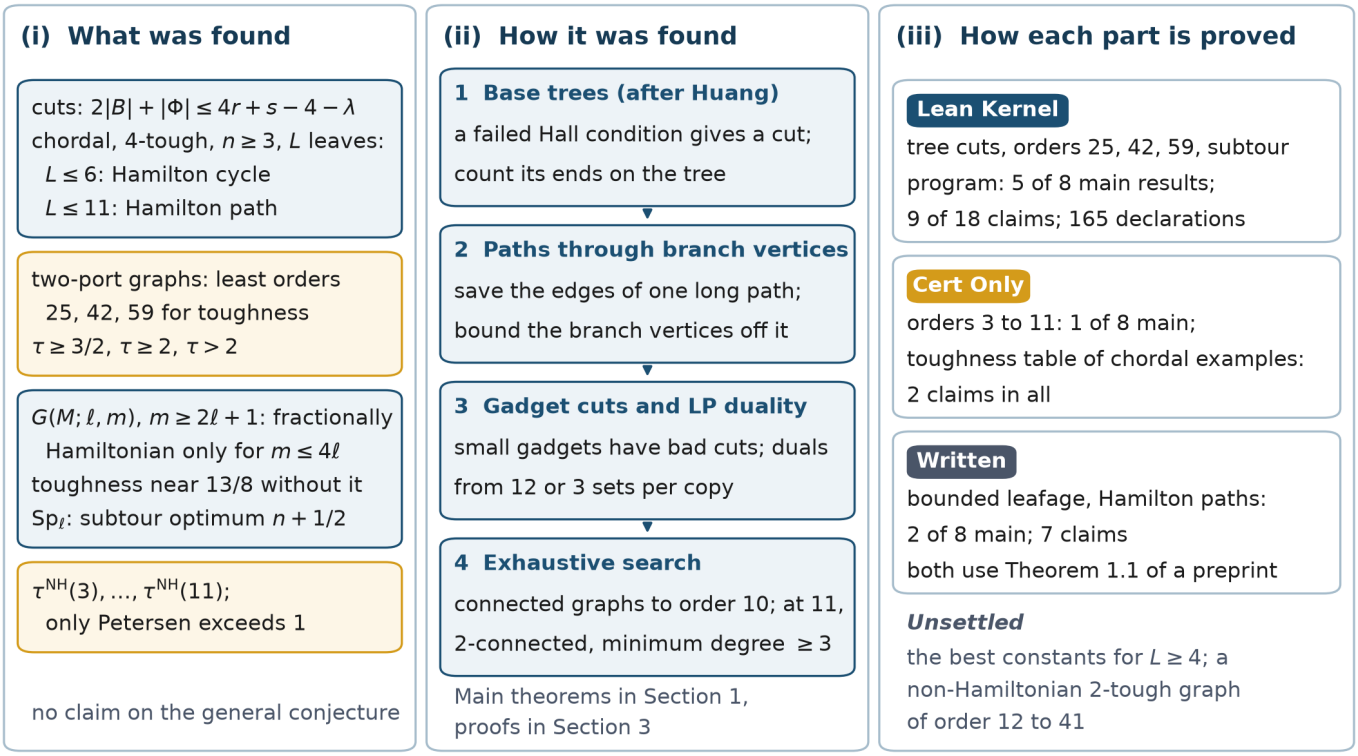


Figure 1. Discovery and proof architecture. Column (i) lists the results, column (ii) how they were found, column (iii) the proof tier of each result. The tiers and counts are those of the claim ledger of Section 5; the declaration count is read from the axiom receipt of the Lean library.

Instant Witness Check: the extremal graph of order 10

```
from fractions import Fraction as F
from itertools import combinations as C, permutations as P
V = [frozenset(s) for s in C(range(5), 2)]  # adjacent if disjoint
def c(X):  # number of components of the subgraph on X
    Y = [*X][:1]
    for u in Y: Y += [v for v in X if not (u & v or v in Y)]
    return 1 + (c(X - {*Y}) if len(Y) < len(X) else 0)
print(min(F(10 - k, c({*X})) for k in range(2, 10)
          for X in C(V, k) if c({*X}) > 1),
      any(all(not p[i-1] & p[i] for i in range(10)) for p in P(V)))
```

Roadmap. Section 2 fixes definitions and their Lean counterparts. Section 3 proves the results: the discharging inequality (§3.1), chordal graphs (§3.2, §3.3), examples of small leafage (§3.4), two-port graphs (§3.5), the subtour program (§3.6), and small orders (§3.7). Section 4 describes the computations, Section 5 the trust boundary, and Section 6 prior work and open questions.

2. DEFINITIONS, NOTATION, AND SETUP

All graphs are finite and simple. For $S \subseteq V(G)$, $G - S$ is the subgraph induced on the other vertices. A *Hamilton cycle* is a cycle through all vertices, so a graph of order 2 has none; a *Hamilton path* is a path through all vertices; a graph is *Hamilton-connected* if any two distinct vertices are the ends of a Hamilton path.

Lean counterparts. In Lean a graph is a `SimpleGraph` on a finite type. $c(G - S)$ is `compAfterDelete`, separating sets are `IsSeparatingSet`, and `IsTough G t`, for rational t, says that $t\,c(G - S) \le |S|$ for every separating set S. Non-Hamiltonicity is the negation of Mathlib's `IsHamiltonian`. That G has a Hamilton path with the ends u and v is written `HasHamiltonianPathBetween G u v`. Chordality appears as `HasPerfectEliminationOrdering`: the vertices can be ordered so that the later neighbors of every vertex are pairwise adjacent. A finite graph is chordal exactly when such an ordering exists [6].

Trees. A *leaf* is a vertex of degree 1; a tree with one vertex has no leaf. `leafCount` and `branchCount` count leaves and branch vertices. For a tree T on at least two vertices the degree sum gives

$$L = 2 + \sum_{\deg v \ge 3} (\deg v - 2) \ge 2 + b. \tag{1}$$

For a set B of edges the components of $T - B$ are trees; the numbers of active and inactive components of Theorem 1 are `activeCompCount` and `inactiveCompCount`, the gray edges form a `ValidGrayMatching`, and λ is `leafIncidence`.

Gadgets. For a graph H with two distinct ports u, v and $S \subseteq V(H)$ let $\omega'(S)$ (`omegaPrime`) be the number of components of $H - S$ that contain neither port. Two-port graphs are `IsGenBBVGraph`; the graphs M and L are `bbvGadgetM` and `bbvGadgetL`, both with 12 edges; $G(M; \ell, m)$, $G(L; \ell, m)$ and Sp_ℓ are `ChordalBBV`, `bbvLGraph` and `ChvatalSplit`. In M the ports are adjacent, in L they are not; in $G(L; \ell, m)$ the two ports of a copy are adjacent, as any two ports are.

Subtour program. `IsUnboundedSEP` says that $x \geq 0$ and $x(\partial(U)) \geq 2$ for every nonempty proper U, `IsBoundedSEP` adds $x \leq 1$, and `IsSEPPolytope` adds weight 2 at every vertex. Weights are rational. Since the constraints have rational coefficients, a graph has a rational fractional Hamilton cycle if and only if it has a real one. A fractional Hamilton cycle has total weight $|V(G)|$.

3. PROOFS OF THE MAIN RESULTS

The discharging inequality. The proof counts the ends of the edges of B by the degrees of the vertices at which they lie.

Proof of Theorem 1. Deleting $|B|$ edges of a tree leaves $|B|+1$ components, so $r+s = |B|+1 \geq 2$. Call a pair (v,e) with $v \in e \in B$ an *end* at v; there are $2|B|$ ends, and every component of $T-B$ contains a vertex with an end. Let K be an inactive component with k vertices. Its vertices have degree at least 3 in T and it has $k-1$ edges, so its vertices carry at least $3k-2(k-1) = k+2 \geq 3$ ends. Counting ends, $2(r+s-1) = 2|B| \geq 3s+r$, that is $s \leq r-2$, and so $r \geq 2$. Every inactive component contains a branch vertex, and these components are disjoint, so $s \leq b$, and $b \leq L-2$ is (1).

Each gray edge $g \in \Phi$ has an endpoint $z(g)$ that carries an end. Since $\deg_T z(g) = 2$, its component is active. The vertices $z(g)$ are distinct because gray edges are disjoint, and none is a leaf. Hence at least $|\Phi|$ ends lie at vertices of degree 2, λ ends lie at leaves, and at least $3s$ ends lie in inactive components, where all degrees are at least 3. So $|\Phi|+\lambda+3s \leq 2|B|$, and

$$Q = 2|B|+|\Phi| \leq 4|B|-3s-\lambda = 4(r+s-1)-3s-\lambda = 4r+s-4-\lambda.$$

The other three bounds follow from $\lambda \geq 0$ and $s \leq r-2$, $s \leq b$, $s \leq L-2$. □

Proposition 9 (Equality; written proof). *For every $L \geq 3$ there are a tree T with L leaves and $L-2$ branch vertices, a gray matching, a set B and a set Φ as in Theorem 1 with $\lambda = 0$ and $Q = 5r-6 = 4r+s-4 = 4r+b-4 = 4r+L-6$.*

Proof. Take a path on $L-2$ vertices and attach leaves so that every path vertex gets degree 3; the resulting tree has L leaves and $L-3$ edges between branch vertices. Subdivide each edge at a leaf y, joining y to a branch vertex w, by two vertices x_1, x_2 in the order y, x_1, x_2, w, and let x_1x_2 be gray. Let B consist of the $L-3$ edges between branch vertices and the L edges x_2w, and let Φ be the set of all L gray edges. Then $|B| = 2L-3$, the components of $T-B$ are the $L-2$ single branch vertices, which are inactive, and the L paths y, x_1, x_2, which are active. So $r = L$, $s = L-2$, $\lambda = 0$, and $Q = 2(2L-3)+L = 5L-6$. □

The proposition says that the inequality is sharp on trees. It does not produce chordal graphs for which Theorem 2 is sharp.

Chordal graphs of bounded leafage. Tree representations are used as in [9, 8]. In a tree representation of G with host tree T_0 every vertex x of G has a nonempty subtree F_x of T_0, and distinct x, y are adjacent if and only if F_x and F_y share a node. The *bag* of a node z is $\{x : z \in F_x\}$.

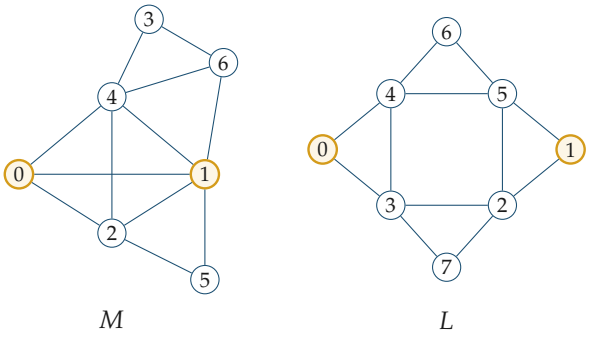


Figure 2. The gadgets M (seven vertices) and L (eight vertices), each with ports 0 and 1.

Lemma 10 (written proof). *If G has a tree representation whose host tree has L leaves and b branch vertices, then it has one whose host tree T_0 has at most L leaves and at most b branch vertices and in which no bag is contained in the bag of an adjacent node. If T_0 then has at least two nodes, every leaf z of T_0 satisfies $F_x = \{z\}$ for some vertex x of G.*

Proof. Let yz be an edge of the host tree such that the bag of z is contained in the bag of y, and contract yz to a node whose bag is the bag of y. The images of the subtrees are subtrees, and two of them share the new node if and only if both contained y, so the contracted tree is a tree representation. The new node has degree $\deg y + \deg z - 2$. If this is 1, then one of y, z was a leaf; if it is at least 3, then one of them had degree at least 3. So the numbers of leaves and of branch vertices do not increase. Repeat until no such edge is left. If z is then a leaf with neighbor y, some x lies in the bag of z and not in that of y, and $F_x = \{z\}$. □

The base tree. Let G be a chordal graph that is not complete, with a representation as in Lemma 10; then T_0 has an edge. Choose an inclusion-maximal independent set I of G such that for every $i \in I$ the subtree F_i is a path whose nodes have degree at most 2 in T_0 and F_i does not properly contain any F_x. Suppress every node of degree 2 of T_0 that is not an end of some F_i, $i \in I$. The resulting tree T is the *base tree*; it has the leaves and the branch vertices of T_0, with the same degrees. Put $Q_x = F_x \cap V(T)$.

(a) Q_x is nonempty for every x. Otherwise all nodes of F_x have degree 2 and are not ends of any F_i. If F_x meets some F_i, it lies inside F_i without the ends of F_i, so F_i properly contains F_x, which is excluded. If F_x meets no F_i, let F_y be minimal among the subtrees contained in F_x; then $I \cup \{y\}$ contradicts the maximality of I.

(b) Each vertex t of T of degree at most 2 lies in Q_i for some $i \in I$. If t has degree 2, it is an end of some F_i. If t is a leaf, $F_x = \{t\}$ for some x; by the maximality of I some $i \in I$ equals x or is adjacent to x, so $t \in F_i$ (and then $F_i = \{t\}$).

The paths F_i, $i \in I$, are disjoint, so Q_i is a vertex of T or the two ends of an edge of T. In the second case call that edge *gray* and write $i_e = i$; all other edges of T are *black*. The ends of a gray edge have degree 2, because a leaf t has some $F_x = \{t\}$, which F_i would properly contain. Gray edges are disjoint. Let $U = V(G) \setminus I$, and for $t \in V(T)$ let $P_t = \{x \in U : t \in Q_x\}$. For an edge $e = tu$ of T let A_e be the hypergraph on U whose edges are the loops $\{x\}$ with $x \in P_t \cap P_u$ and the

pairs $\{x,y\}$ with $xy \in E(G)$, $x \in P_t$ and $y \in P_u$. Write $\mathcal{A}(X)$ for the union of the A_e, $e \in X$.

Huang's theorem. A *tree-structured hypergraph system* on a tree T consists of a finite set U, a subtree of T for each element of U, and for each edge $e = tu$ of T a hypergraph A_e on U whose loops are the elements whose subtrees contain t and u and whose other edges are pairs $\{x,y\}$ with t in the subtree of x and u in that of y. For $q\colon E(T) \to \{1,2,\dots\}$ a *q-SDR* is a choice of $q(e)$ edges from every A_e such that all chosen edges are pairwise disjoint. Theorem 1.1 of [8] says that a q-SDR exists if and only if for every set X of edges of T the union of the A_e, $e \in X$, has a matching of size at least $q(X) = \sum_{e\in X} q(e)$.

Lemma 11 (written proof). *Let G be chordal and not complete, with base tree T as above.*

(i) If the matching condition fails for a set F_0 of edges of T and a function $q \geq 1$ on F_0, that is, if $\mathcal{A}(F_0)$ has no matching of size $q(F_0)$, then some $C \subseteq U$ with $|C| \leq q(F_0)-1$ meets every edge of $\mathcal{A}(F_0)$.

(ii) Let $C \subseteq U$, let B be the set of black edges e for which C meets every edge of A_e, and let D be a set of gray edges with the same property that share no vertex with an edge of B. Put $S = C \cup \{i_e : e \in D\}$ and let r be the number of active components of $T - B$. Then $|S| = |C| + |D|$ and $c(G-S) \geq r + |D|$.

Proof. (i) This is the argument of [8, Section 3.4]. Let Λ be the set of loops in $\mathcal{A}(F_0)$. If Q_x contains both ends of an edge of F_0, then $x \in \Lambda$. So for every other x the set Q_x lies in one component of $T - F_0$. Contract each component of $T - F_0$ to a point and color the points of the resulting tree with two colors; the pairs of $\mathcal{A}(F_0)$ that avoid Λ join elements of different colors, so they form a bipartite graph H. By Kőnig's theorem H has a vertex cover of the size of its largest matching, and the loops together with a matching of H are a matching of $\mathcal{A}(F_0)$. So Λ and a smallest vertex cover of H form a set of size at most $q(F_0) - 1$ that meets every edge.

(ii) The first equation holds because $C \subseteq U$ and $i_e \in I$. The edges of D lie in active components of $T - B$. Call the components of $T - (B \cup D)$ that lie in active components of $T - B$ *regions*; there are $r + |D|$ of them.

Every region R contains a vertex of Q_x for some $x \notin S$. Let K be the component of $T - B$ that contains R. If K contains no edge of D, then $R = K$ contains a vertex t of degree at most 2, and $t \in Q_i$ for some $i \in I$ by (b). If $i = i_e$ with $e \in D$, then e contains t and lies in K, a contradiction; so $i \notin S$. If K contains an edge of D, then R contains an end t of some $e \in D$. The other edge f at t is black, and it is not in B because e shares no vertex with an edge of B. So some edge of A_f avoids C, and it contains an $x \in U \setminus C$ with $t \in Q_x$.

If $x \notin S$, then Q_x does not contain both ends of an edge $e \in B \cup D$. For $x \in U$ the loop $\{x\}$ would belong to A_e and avoid C. For $x \in I$ the set Q_x is a vertex or the ends of a gray edge, and $e \in D$ would give $x = i_e \in S$. So Q_x lies in one component of $T - (B \cup D)$.

If $x, y \notin S$ are adjacent, then Q_x and Q_y lie in the same component of $T - (B \cup D)$. Suppose not. Then some $e = tu \in B \cup D$ has Q_x on the side of t and Q_y on the side of u in $T - e$. Let z be a common node of F_x and F_y. It is not a vertex of T, so it was suppressed and lies inside the path of T_0 that replaces an edge f of T. By (a) and connectedness each of F_x, F_y contains an end of f. If $f \neq e$, both ends of f are on the same side of $T - e$, a contradiction. So $f = e$, $t \in F_x$, $u \in F_y$, $u \notin F_x$ and $t \notin F_y$. If $x \in I$, the path F_x contains t and z and its ends are vertices of T, so it contains u, a contradiction; the same holds for y. So $x, y \in U$, $x \in P_t$, $y \in P_u$, and $\{x,y\}$ is an edge of A_e that avoids C, a contradiction.

So the vertices of $G - S$ whose sets Q_x lie in a given region form a nonempty union of components of $G - S$, and there are at least $r + |D|$ components. □

Lemma 12 (written proof). *Let G be chordal and not complete, with base tree T, and let $q(e) = 2$ for black and $q(e) = 1$ for gray edges e. If $(A_e)_{e\in E(T)}$ has a q-SDR, then G has a Hamilton cycle or at most two vertices.*

Proof. Let W be a closed walk in T that uses every edge exactly twice, once in each direction. Assign the two chosen edges of A_e to the two passages of a black edge e. For a gray edge e assign the chosen edge of A_e to one passage and $\{i_e\}$ to the other. Follow W and record, for every passage from t to u, the element of the assigned loop, or the elements x, y of the assigned pair in the order $x \in P_t$, $y \in P_u$. Vertices recorded for one passage are adjacent. If two consecutive passages meet at $w \in V(T)$, then the last vertex recorded for the one and the first vertex recorded for the other have w in their subtrees, so they are adjacent. The recorded vertices are distinct, because the chosen edges are disjoint and lie in U, and the vertices $i_e \in I$ are distinct. Every vertex x not yet recorded has some $w \in Q_x$ by (a); insert it at one visit of W to w, between the two passages. All vertices inserted there and their two neighbors in the sequence contain w in their subtrees and are pairwise adjacent. The result is a cyclic sequence of all vertices of G in which consecutive vertices are adjacent. □

Proof of Theorem 2. If G is complete, all claims hold. Otherwise replace the representation by that of Lemma 10; L and b do not increase, and the hypotheses in (i) and (iii) persist. In (ii) and (iv) let $L' \leq L$ be the new number of leaves. If $L' \geq 7$, respectively $L' \geq 5$, then $5 - 7/L' \leq 5 - 7/L$, respectively $5 - 5/L' \leq 5 - 5/L$, and the hypothesis persists. Otherwise $\tau(G) > 4$, since $5 - 7/L \geq 4$ for $L \geq 7$ and $5 - 5/L \geq 4$ for $L \geq 5$, and the case is covered by (i), respectively (iii). So assume that the host tree is T_0 and let T be the base tree, which has L leaves and b branch vertices. Let q be as in Lemma 12.

Hamilton cycles. Suppose that G is not Hamiltonian. Then by Lemma 12 there is no q-SDR, so by Huang's theorem the matching condition fails for some F_0, which is nonempty, and Lemma 11(i) gives C with $|C| \leq q(F_0) - 1$. Let B be the set of black and N the set of gray edges e for which C meets every edge of A_e, so $F_0 \subseteq B \cup N$; let $\Phi \subseteq N$ be the set of those that share a vertex with an edge of B, let $D = N \setminus \Phi$, and put $Q = 2|B| + |\Phi|$. Then

$$|S| = |C| + |D| \leq q(B \cup N) - 1 + |D| = Q + 2|D| - 1. \qquad (2)$$

Write $c = c(G - S) \geq r + |D|$. If $B = \emptyset$, then $Q = 0$, $r = 1$, $D = N$ is nonempty, S is separating, and $|S| \leq 2|D| - 1 \leq$

$2c-3<4c$. If $B\neq\emptyset$, then $c\geq r\geq 2$, and Theorem 1 gives

$$|S|\leq 4r+s-5+2|D|\leq 4c+s-5,\qquad |S|\leq 5r-7+2|D|\leq 5c-7. \tag{3}$$

If $s\leq 4$, then $|S|<4c$; since $s\leq b$ and $s\leq L-2$, this proves (i). If $L\geq 7$, then $|S|\leq\min(5c-7,4c+L-7)$, so $|S|/c\leq 5-7/L$: use the first bound for $c\leq L$ and the second for $c\geq L$. This and $2c-3<4c\leq(5-7/L)c$ prove (ii).

Hamilton paths with given ends. Let $u\neq v$ and $Z=\{u,v\}\setminus I$. By [8, Lemma 3.3] a q-SDR of the system $(A_e-Z)_e$, in which the elements of Z are deleted from U and the edges that meet Z from every A_e, gives a Hamilton path from u to v. If there is none, Huang's theorem and the proof of Lemma 11(i), applied to this system, give a set $C_0\subseteq U\setminus Z$ with $|C_0|\leq q(F_0)-1$ that meets every edge of $\mathcal{A}(F_0)$ that avoids Z, so $C=C_0\cup Z$ meets every edge of $\mathcal{A}(F_0)$ and $|C|\leq q(F_0)+1$. As before, $|S|\leq Q+2|D|+1$. If $B=\emptyset$, then $|S|\leq 2|D|+1\leq 2c-1<4c$. Otherwise $|S|\leq\min(5c-5,\,4c+s-3)$. If $s\leq 2$, then $|S|<4c$, and this gives (iii). If $L\geq 5$, then $|S|\leq\min(5c-5,4c+L-5)$, so $|S|/c\leq 5-5/L$, which gives (iv). □

Hamilton paths.

Proof of Theorem 3. A complete graph has a Hamilton path. Otherwise let T_0 and T be as in the proof of Theorem 2. A smaller number of leaves is harmless here too, because $5-7/(L-5)$ is increasing in $L\geq 12$ and at least 4 there. The base tree T has an edge. Let P be a path in T between two distinct leaves s_1 and s_2 that contains as many branch vertices as possible. By (b) there are $i_1,i_2\in I$ with $Q_{i_1}=\{s_1\}$ and $Q_{i_2}=\{s_2\}$.

Multiplicities. Let $q^P(e)$ be 1 for black and 0 for gray edges e of P, and 2 for black and 1 for gray edges off P. Contract the gray edges of P. The images of the sets Q_x, $x\in U$, span subtrees of the contracted tree T', and (A_e), for $e\in E(T')$, is a tree-structured system on T': an element whose image contains both ends of an edge e of T' has, since Q_x is connected in T, both ends of e in Q_x.

A q^P-SDR gives a Hamilton path. Let W be the walk from s_1 to s_2 that follows P and makes a round trip through every branch off P; it uses the edges of P once and all others twice. For a black edge off P assign the two chosen edges to the two passages, for a black edge of P the chosen edge to its passage, for a gray edge e off P the chosen edge to one passage and $\{i_e\}$ to the other, and for a gray edge e of P assign $\{i_e\}$. Record vertices as in the proof of Lemma 12, with i_1 first and i_2 last; i_1 and i_2 are not of the form i_e. Insert the missing vertices as before, at s_1 after i_1 and at s_2 before i_2. The result is a Hamilton path.

A violated condition gives a small cut. Suppose that there is no Hamilton path. Huang's theorem on T' gives a set $F_0\subseteq E(T')$, nonempty, for which $\mathcal{A}(F_0)$ has no matching of size $q^P(F_0)$. Lemma 11(i) gives C with $|C|\leq q^P(F_0)-1$. Let B be the set of black edges e for which C meets every edge of A_e, with B_{on} its edges on P and B_{off} the others; let N be the set of gray edges off P with the same property, $\Phi\subseteq N$ the set of those that share a vertex with an edge of B, and $D=N\setminus\Phi$. Put $Q^P=2|B_{\mathrm{off}}|+|B_{\mathrm{on}}|+|\Phi|$ and $S=C\cup\{i_e:e\in D\}$. Then $F_0\subseteq B\cup N$ and $|S|\leq Q^P+2|D|-1$, and Lemma 11(ii) gives $c=c(G-S)\geq r+|D|$. If $B=\emptyset$, then D is nonempty and $|S|\leq 2c-3<4c$. Let $B\neq\emptyset$, so $c\geq r\geq 2$. Since $Q^P\leq 2|B|+|\Phi|$, Theorem 1 gives $|S|\leq 5c-7$.

Counting off the path. Let s_{on} and s_{off} be the numbers of inactive components of $T-B$ that meet P and that do not. Each $g\in\Phi$ has an end $z(g)$ on an edge of B. If $z(g)$ were on P, then, having degree 2, it would be an inner vertex of P and both its edges would be on P, which is false for g. So the edge of B at $z(g)$ is off P. An inactive component with k vertices carries at least $k+2$ ends by the proof of Theorem 1. If it misses P, then all of them belong to B_{off}, and there are at least 3. If it meets P, it meets P in a path that contains no end of P, because leaves are in active components; so exactly two of its ends belong to edges of P and at least $k\geq 1$ belong to B_{off}. Counting the $2|B_{\mathrm{off}}|$ ends of B_{off} as before, $|\Phi|+3s_{\mathrm{off}}+s_{\mathrm{on}}\leq 2|B_{\mathrm{off}}|$. With $|B|=r+s_{\mathrm{on}}+s_{\mathrm{off}}-1$ this gives

$$Q^P\leq 4|B_{\mathrm{off}}|+|B_{\mathrm{on}}|-3s_{\mathrm{off}}-s_{\mathrm{on}}=4r-4+s_{\mathrm{off}}+3s_{\mathrm{on}}-3|B_{\mathrm{on}}|.$$

Deleting B_{on} cuts P into $|B_{\mathrm{on}}|+1$ pieces, the two outer ones contain leaves, and different components meet P in different pieces, so $s_{\mathrm{on}}\leq|B_{\mathrm{on}}|-1$ if B_{on} is nonempty and $s_{\mathrm{on}}=0$ otherwise. In both cases $Q^P\leq 4r-4+s_{\mathrm{off}}$, and so $|S|\leq 4r+s_{\mathrm{off}}-5+2|D|\leq 4c+s_{\mathrm{off}}-5$.

Branch vertices off the path. Inactive components that miss P contain branch vertices off P, so $s_{\mathrm{off}}\leq b-d$, where d is the number of branch vertices on P. We claim

$$b-d\leq\max(0,\lfloor L/2\rfloor-2,\,L-7). \tag{4}$$

If $d\geq 5$, use $b\leq L-2$. Let T_b be the tree whose vertices are the branch vertices, two being adjacent when the path of T between them has no other branch vertex. Every end of a longest path of T_b has at least two branches of T that lead to leaves without further branch vertices (when $b\geq 1$), so a longest path of T_b extends to a path between two leaves and has d vertices. If $d\leq 2$, then $b=d$. If $d=3$, then T_b is a star with $b-1$ leaves, each of which is adjacent in this sense to at least two leaves of T, so $L\geq 2(b-1)$ and $b-3\leq\lfloor L/2\rfloor-2$. If $d=4$, then T_b is a double star with $b-2$ leaves, $L\geq 2(b-2)$, and $b-4\leq\lfloor L/2\rfloor-2$.

For $L\leq 11$ the right side of (4) is at most 4, so $|S|\leq 4c-1$, and G is not 4-tough. For $L\geq 12$ it is $L-7$, so $|S|\leq\min(5c-7,4c+L-12)$ and $|S|/c\leq 5-7/(L-5)$: use the first bound for $c\leq L-5$ and the second for $c\geq L-5$. Also $2c-3<4c\leq(5-7/(L-5))c$. □

Examples of small leafage. For $\ell\geq 1$ and $a,b\geq 0$ with $a+b=2\ell+1$ let $H(\ell,a,b)$ be the graph that consists of a clique K on ℓ vertices, a copies of M and b copies of an edge $\{p,i\}$ with the single port p, with K joined to all other vertices and all $2a+b$ ports made pairwise adjacent. Its order is $\ell+7a+2b$, $H(\ell,0,2\ell+1)=\mathrm{Sp}_\ell$, and $H(\ell,2\ell+1,0)=G(M;\ell,2\ell+1)$.

Proposition 13 (written proof). *Every $H(\ell,a,b)$ is chordal and non-Hamiltonian and has leafage exactly $2a+b$.*

Proof. Representation. Take a center node with bag K and all ports. For each copy of M attach to the center a node with bag $K\cup\{0,1,2,4\}$, attach to this node two nodes with bags $K\cup\{1,2,5\}$ and $K\cup\{1,4,6\}$, and attach to the latter a node with bag $K\cup\{3,4,6\}$, all numbers read in the copy. For each copy of $\{p,i\}$ attach a node with bag $K\cup\{p,i\}$ to the center. Every vertex occupies a subtree, every bag is a clique, and

every edge lies in a bag, so this is a tree representation, and $H(\ell,a,b)$ is chordal by Gavril's theorem [7]. Since $a+b\geq 3$ the center is not a leaf, and the tree has $2a+b$ leaves.

Lower bound. Let Y consist of the vertices 3 and 5 of every copy of M and the vertex i of every copy of $\{p,i\}$, so $|Y| = 2a+b\geq 3$. The closed neighborhoods are $N[3] = K\cup\{3,4,6\}$, $N[5] = K\cup\{1,2,5\}$ and $N[i] = K\cup\{p,i\}$; no two elements of Y are adjacent, and for every $y\in Y$ the other elements of Y lie in one component of $H(\ell,a,b)-N[y]$: in the copy of y the paths $5,2,0$ and $3,4,0$ lead to the port 0, in other copies every element of Y reaches a port, and the remaining ports are pairwise adjacent. Take any tree representation with host tree T. For $y\in Y$ let X_y be the component of $T-F_y$ that contains all F_x with $x\in Y\setminus\{y\}$; it exists by the last sentence. Let Y_y be the subtree $T-X_y$, which contains F_y, and let e_y be the one edge between Y_y and X_y; it has an end in F_y. The subtree Y_y contains a leaf of T: if it has one node, this node is a leaf of T, and otherwise it has at least two leaves, one of which is not the end of e_y. It remains to see that Y_x and Y_y are disjoint for $x\neq y$. If $e_x = e_y$, they are the two sides of this edge, since $F_x\subseteq Y_x\cap X_y$. Otherwise e_y has an end in $F_y\subseteq X_x$ and $e_y\neq e_x$, so e_y is an edge of X_x and not of Y_x. Hence Y_x is connected in $T-e_y$ and contains $F_x\subseteq X_y$, so $Y_x\subseteq X_y$ is disjoint from Y_y. So T has at least $2a+b$ leaves.

No Hamilton cycle. Deleting K from a Hamilton cycle leaves at most ℓ paths, with at most 2ℓ end vertices. Some copy contains none of them, as there are $2\ell+1$ copies. It is not a copy of $\{p,i\}$, because i has the single neighbor p outside K. In a copy of M without end vertices the paths induce subpaths that begin and end at ports, because only ports have neighbors outside the copy and K. A subpath through a vertex that is not a port therefore contains both ports, so it is the only subpath and a Hamilton path of M from 0 to 1. But M has no such path (`recompute.py`; in Lean it is part of the proof of Theorem 5). □

Proposition 14 (certificate). *For $3\leq L\leq 14$ the largest toughness of a graph $H(\ell,a,b)$ with $2a+b=L$ is the one in Table 1, attained by the graph listed there. In particular $H(2,4,1)$ is a chordal non-Hamiltonian graph of order 32, leafage 9, and toughness exactly 3/2.*

The computation uses that a separating set contains K, so the toughness is obtained by adding, over the copies, the least number of deleted vertices for each pair (number of components without a port, whether a port is left). Whether a non-Hamiltonian chordal graph of toughness exactly 3/2 exists was asked on MathOverflow [19]; $H(2,4,1)$ is one, with the status of the proposition.

Two-port graphs.

Proposition 15 (Small gadgets; Lean). *Let H be a graph on at least 3 vertices and let $u\neq v$ be vertices of H such that H has no Hamilton path from u to v. If $|V(H)|\leq 5$, then some S has $\omega'(S)\geq 1$ and $|S|\leq\omega'(S)$. If $|V(H)|\leq 7$, then some S has $\omega'(S)\geq 1$ and $|S|\leq 2\omega'(S)-1$.*

Proposition 16 (The gadget L; Lean). *L has no Hamilton path from 0 to 1, and $|S|\geq 2\omega'(S)$ for every $S\subseteq V(L)$ with $\omega'(S)\geq 1$.*

The proof of Proposition 15 in Lean is a case analysis over the gadgets of at most seven vertices; we do not reproduce it.

Table 1. The largest toughness of the graphs $H(\ell,a,b)$ of leafage $L = 2a+b$, with the least order attaining it. Since $L = 2\ell+1+a$, only finitely many (ℓ,a,b) have a given L.

L	(ℓ,a,b)	order	τ	L	(ℓ,a,b)	order	τ
3	(1,0,3)	7	1	9	(2,4,1)	32	3/2
4	(1,1,2)	12	1	10	(2,5,0)	37	17/11
5	(2,0,5)	12	6/5	11	(3,4,3)	37	3/2
6	(1,3,0)	22	4/3	12	(3,5,2)	42	20/13
7	(3,0,7)	17	9/7	13	(3,6,1)	47	11/7
8	(2,3,2)	27	4/3	14	(3,7,0)	52	8/5

Proposition 16 is a finite check, also made by `recompute.py`.

Proof of Theorem 4. Non-Hamiltonicity. Suppose that G has a Hamilton cycle. Deleting K from it leaves at most ℓ paths with at most $2\ell<m$ end vertices, so some W_i contains none of them. Then both neighbors on the cycle of every vertex of W_i lie in $W_i\cup\{u_i,v_i\}$, and the cycle meets W_i in segments, each of which is preceded and followed by a port of H_i. If the same port preceded and followed a segment, or if there were two segments, the cycle would be contained in $W_i\cup\{u_i,v_i\}$ and would miss K. So there is one segment, and with the two ports it is a Hamilton path of H_i from u_i to v_i.

Cuts. Let G be t-tough and $w_i = |W_i|$, $n = \ell+|C|+\sum_i w_i$.

(C1) Deleting $K\cup\{u_1,v_1\}$ separates W_1 from W_2, so $2t\leq \ell+2$.

(C2) Deleting $K\cup C$ leaves at least m components, so $tm\leq \ell+|C|$.

(C3) If two gadgets H_i, H_j share a port, deleting K and their at most three ports leaves at least three components, so $3t\leq\ell+3$. If no two gadgets share a port, then $|C|\geq 2m$.

(C4) Let J be a set of fewer than m gadgets, $S_i\subseteq V(H_i)$ for $i\in J$, and $\Omega = \sum_{i\in J}\omega'(S_i)\geq 1$. Deleting K and all S_i leaves at least $\Omega+1$ components, the last one because a gadget outside J keeps its set W_j. So $t(\Omega+1)\leq \ell+\sum_{i\in J}|S_i|$.

$t=3/2$. If $\ell=1$, then (C3) gives $|C|\geq 2m\geq 6$. If $w_i\leq 5$, then $|V(H_i)|\leq 7$, and (C4) with the set S of Proposition 15 gives $\frac{3}{2}(\omega'+1)\leq 1+|S|\leq 2\omega'$, so $\omega'\geq 3$ and $|S|\geq 5$, but $\omega'+|S|\leq|V(H_i)|\leq 7$. So all $w_i\geq 6$ and $n\geq 1+6+18=25$. Let $\ell\geq 2$. At most $2\ell-3$ gadgets have $w_i\leq 3$: otherwise take J of size $2\ell-2<m$ among them; Proposition 15 gives $|S_i|\leq\omega'(S_i)$, and (C4) gives $3(\Omega+1)\leq 2\ell+2\Omega$, so $\Omega\leq 2\ell-3$, but $\Omega\geq|J|$. With (C2),

$$n\geq \tfrac{3}{2}m+4m-3(2\ell-3)\geq \tfrac{11}{2}(2\ell+1)-6\ell+9 = 5\ell+\tfrac{29}{2}\geq\tfrac{49}{2},$$

so $n\geq 25$.

$t=2$. By (C1) $\ell\geq 2$. At most $\ell-2$ gadgets have $w_i\leq 5$: otherwise take J of size $\ell-1$; then $\sum|S_i|\leq 2\Omega-|J|$ and $2(\Omega+1)\leq\ell+2\Omega-|J|$, so $|J|\leq\ell-2$. Hence $\sum_i w_i\geq 6m-5(\ell-2)$. If $\ell=2$, (C3) gives $|C|\geq 2m$ and $n\geq 2+8m\geq 42$. If $\ell\geq 3$, (C2) gives $n\geq 8m-5(\ell-2)\geq 11\ell+18\geq 51$.

$t>2$. Now $|S|\geq 2c(G-S)+1$ for all separating S, so (C1) gives $\ell\geq 3$, and as before at most $\ell-3$ gadgets have $w_i\leq 5$, so $\sum_i w_i\geq 6m-5(\ell-3)$. If $\ell=3$, then $3t>6=\ell+3$, so (C3) gives $|C|\geq 2m$ and $n\geq 3+8m\geq 59$. If $\ell\geq 4$, then $|C|+\ell\geq 2m+1$ and $n\geq 8m+1-5(\ell-3)\geq 11\ell+24\geq 68$.

Attainment. $G(L;\ell,m)$ has $\ell+8m$ vertices and is a two-port graph by Proposition 16, so it is non-Hamiltonian for

$m \geq 2\ell+1$. Let S be separating. Then $K \subseteq S$. Each copy has at most two components without a port, as its other vertices form the triangles $\{2,3,7\}$ and $\{4,5,6\}$, and the surviving ports lie in one component. With S_i the part of S in the ith copy, $c = c(G-S) \leq 1 + \sum_i \omega'(S_i) \leq 2m+1$ and $|S| \geq \ell + 2\sum_i \omega'(S_i) \geq \ell + 2(c-1)$. For $\ell = 1$ this is $2c-1 \geq \frac{3}{2}c$, for $\ell = 2$ it is $2c$, and for $\ell = 3, m = 7$ it is $2c+1 \geq \frac{31}{15}c$ because $c \leq 15$. □

The subtour program. Every feasible x of the subtour program satisfies $\sum_e x_e \geq 2\sum_U y_U$ for all weights $y_U \geq 0$ on nonempty proper subsets U such that every edge lies in cuts $\partial(U)$ of total weight at most 1, since $\sum_e x_e \geq \sum_U y_U\, x(\partial(U))$.

Proof of Theorem 5. Chordality. Order the vertices as follows: for each copy of M its vertices $3,5,6,4,2$ in this order, then all ports, then K. The later neighbors of $3,5,6,4,2$ in the copy are $\{4,6\}$, $\{1,2\}$, $\{1,4\}$, $\{0,1,2\}$, $\{0,1\}$; together with K these are cliques, and all ports with K form a clique.

Lower bound. On each copy of M put the weights

$$y_{\{0\}} = y_{\{2\}} = y_{\{4\}} = \tfrac{3}{8}, \quad y_{\{1\}} = \tfrac{1}{4}, \quad y_{\{3\}} = y_{\{6\}} = \tfrac{1}{2},$$
$$y_{\{5\}} = \tfrac{5}{8}, \quad y_{\{2,5\}} = y_{\{3,6\}} = y_{\{3,4,6\}} = \tfrac{1}{8},$$
$$y_{\{1,\dots,6\}} = y_{\{0,\dots,6\}} = \tfrac{1}{8}.$$

Their sum is $\frac{29}{8}$. For every edge of M, every edge from a vertex of the copy to K, and every edge between two ports of different copies the total weight of the cuts that contain it is at most 1; these are finitely many sums up to symmetry, checked in Lean. With m copies the bound is $2m \cdot \frac{29}{8} = \frac{29m}{4}$.

Threshold. A fractional Hamilton cycle has weight $\ell + 7m$, and $29m/4 \leq \ell + 7m$ holds exactly when $m \leq 4\ell$. Conversely let $2\ell+1 \leq m \leq 4\ell$ and $\theta = 4\ell/m - 1 \in [0,1)$. Give weight $1/(2\ell)$ to the edges between K and the vertices 5, weight $\theta/(2\ell)$ to those between K and the vertices 3, weight $\frac{1-\theta}{2(m-1)}$ to the edges between ports 0 of different copies, weight $\frac{1+\theta}{2(m-1)}$ to those between a port 0 and a port 1 of different copies, weight 0 to the other edges outside the copies, and inside each copy the weights $(1-\theta)w_0 + \theta w_1$, where $2w_0$ is 1 on $02,04,15,24$, 2 on $16,25,34,36$ and 0 on the other edges, and $2w_1$ is 1 on $02,04,15,16,24,34,46$, 2 on $25,36$ and 0 on the other edges. The Lean proof checks that this is a fractional Hamilton cycle. Non-Hamiltonicity for $m \geq 2\ell+1$ is Theorem 4. □

Proof of Theorem 6. In M every S has $\omega'(S) \leq 2$, because the other vertices form the cliques $\{2,5\}$ and $\{3,4,6\}$, and $|S| \geq \frac{3}{2}\omega'(S)$: one such component is not cut off from the ports by one vertex, and two are not by two vertices. As for $G(L;\ell,m)$, a separating set S of $G(M;\ell,m)$ with $c = c(G-S)$ satisfies $c \leq 2m+1$ and $|S| \geq \ell + \frac{3}{2}(c-1)$. For $\ell \geq 2$ the quotient $(\ell + \frac{3}{2}(c-1))/c$ decreases in c, so the graph is $\frac{\ell+3m}{2m+1}$-tough. With $m = 4\ell+1$ this is

$$\frac{13\ell+3}{8\ell+3} = \frac{13}{8} - \frac{15}{8(8\ell+3)},$$

which exceeds $\frac{13}{8} - \varepsilon$ for large ℓ, and by Theorem 5 the graph is not fractionally Hamiltonian. □

Proof of Theorem 7. Toughness and Hamiltonicity. The graph Sp_ℓ equals $H(\ell,0,2\ell+1)$ and is non-Hamiltonian by Proposition 13. A separating set contains A; if it contains j vertices of R and leaves $c \geq 2$ components, then $c \leq j+1$ and, as there are $2\ell+1$ vertices i_r, $c \leq 2\ell+1$, so $|S|/c \geq (\ell + c - 1)/c \geq 3\ell/(2\ell+1)$, as the middle term does not increase with c. Deleting A and all but one vertex of R gives equality.

Bounds. Put $y_{\{i_r\}} = \frac{3}{4}$, $y_{\{r\}} = y_{\{r,i_r\}} = \frac{1}{4}$. The edge ri_r lies in cuts of weight $\frac{3}{4} + \frac{1}{4}$, the edges from i_r to A in $\frac{3}{4} + \frac{1}{4}$, from r to A in $\frac{1}{2}$, and between two vertices of R in $\frac{1}{2} + \frac{1}{2}$. So the optimum is at least $2(2\ell+1) \cdot \frac{5}{4} = 5\ell + \frac{5}{2} = n + \frac{1}{2}$. The weighting with $x = 1$ on the edges ri_r, $1/\ell$ on the edges from i_r to A, $\frac{1}{2}$ on the edges of a Hamilton cycle of R, and 0 elsewhere has total weight $(2\ell+1)(1 + 1 + \frac{1}{2}) = n + \frac{1}{2}$, and the Lean proof checks that all cuts have weight at least 2. A fractional Hamilton cycle would be a feasible point of weight n. □

Proposition 17 (The graphs $G(L;\ell,m)$; Lean). *For $\ell \geq 1$ and $m \geq 2\ell+1$ the graph $G(L;\ell,m)$ has $\ell + 8m$ vertices, is non-Hamiltonian, and is fractionally Hamiltonian.*

The fractional Hamilton cycle gives weight $1/m$ to the edges between K and ports, $\frac{m-\ell}{2m(m-1)}$ to the edges between ports of different copies, weight 1 to $27,37,46,56$ and $\frac{1}{2}$ to $03,04,12,15,25,34$ in each copy, and 0 to the other edges.

Small orders.

Proposition 18 (The Petersen graph; Lean). *The Petersen graph is non-Hamiltonian, has toughness exactly $4/3$, and is fractionally Hamiltonian.*

The proposition is classical; weight $2/3$ on every edge is a fractional Hamilton cycle.

Proof of Theorem 8. A disconnected graph has toughness 0. For $3 \leq n \leq 10$ every connected graph of order n, up to isomorphism, was generated with `geng` [15], tested for a Hamilton cycle by exhaustive search, and, if non-Hamiltonian, given its exact toughness over all vertex subsets. Table 2 lists the results. The graph of order 10 with toughness $4/3$ is cubic, adjacent vertices have no common neighbor, and non-adjacent vertices have exactly one, which characterizes the Petersen graph. Let $n = 11$. A non-Hamiltonian graph of toughness greater than 1 is not complete, and a separating set of at most two vertices would give toughness at most 1, so it is 3-connected and has minimum degree at least 3. All 577,050,233 graphs of order 11 that are 2-connected with minimum degree at least 3 were generated; 878,099 of them are non-Hamiltonian, and none of these has toughness greater than 1. Toughness 1 is attained for every $n \geq 7$: take a clique on $n-3$ vertices $a, r_1, r_2, r_3, \dots$ and three further vertices i_j adjacent to a and r_j. A Hamilton cycle would contain three edges at a. A separating set contains a, and if it contains j of r_1, r_2, r_3, it leaves at most $j+1$ components. Deleting a and r_1 leaves two. □

Table 2. Connected graphs of order n: their number, the number of non-Hamiltonian ones, the largest toughness $\tau^{\mathrm{NH}}(n)$ of those, and the number of graphs that attain it.

n	connected	non-Hamiltonian	$\tau^{\mathrm{NH}}(n)$	attained by
3	2	1	1/2	1
4	6	3	1/2	2
5	21	13	2/3	2
6	112	64	2/3	6
7	853	470	1	4
8	11,117	4,921	1	44
9	261,080	83,997	1	1,103
10	11,716,571	2,411,453	4/3	1

4. COMPUTATIONAL VERIFICATION

The census. `cert/census.py` compiles `geng` of nauty 2.8.9 with the hook `cert/nonham.c`, which tests every generated graph inside the generator, runs it for $n = 3, \dots, 10$ on connected graphs and for $n = 11$ in 6 parts on 2-connected graphs of minimum degree at least 3, prints every non-Hamiltonian graph of toughness greater than 1 in graph6 format and ends with `CENSUS CHECK: PASS` only if the four numbers of every order agree with the expected values and the only graph listed is one graph of order 10 with toughness 4/3 that is the Petersen graph; the output is `cert/census.out`. On an arm64 machine with 16 logical CPUs the recorded run took, after compilation, 64 s of wall time and 247 s of CPU time in the generator processes (`cert/census-time.txt`). The program relies on `geng` and on the C compiler. Independent of them, `recompute.py` repeats the values for $3 \le n \le 6$ over all labeled graphs, and decodes the graph of order 10. The numbers of connected and of Hamiltonian graphs of order at most 10 agree with the sequences A001349 and A003216 of the OEIS [16].

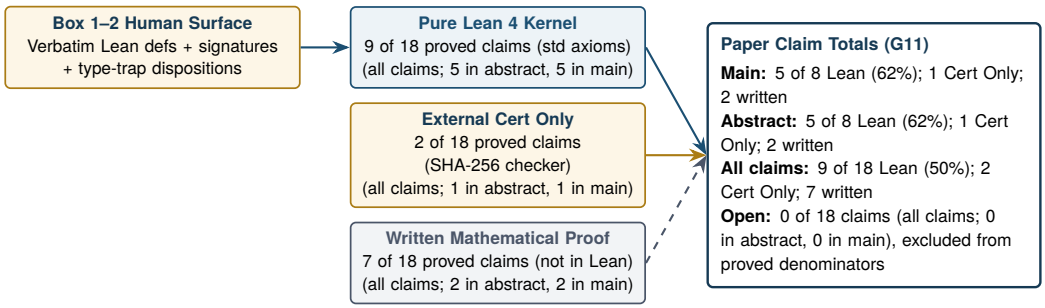
The replay. `recompute.py` uses exact arithmetic and the standard library. It runs the box of Section 1; checks the two gadgets over all vertex subsets; computes the toughness of $G(L;1,3)$, $G(L;2,5)$, $G(L;3,7)$, of $G(M;\ell,m)$ for $2 \le \ell \le 5$ and four values of m each, of Sp_ℓ for $1 \le \ell \le 6$ and of the graphs of Table 1, and compares the method with all subsets for four graphs of orders 7, 12, 12 and 17; checks the weights printed in Section 3.6 for $\ell \le 3$, for the split graphs $\ell \le 5$, with exact minimum cuts; checks the clique trees, the sets Y and perfect elimination orderings for the graphs of the table; checks the trees of Proposition 9 for $3 \le L \le 12$, and the inequalities of Theorem 1 and of Section 3.3 on seeded random trees; compares Tables 1 and 2 with the computed values; and checks the threshold arithmetic of Theorems 2 and 3 for $L < 60$ and $c < 400$. These replays of written proofs are checks, not parts of the proofs.

5. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel & Toolchain:** leanprover/lean4:v4.35.0; committed axiom receipt anchored to git commit b98496b0cde4 (independent gate-box kernel replay pending); standard axioms only (propext, Classical.choice, Quot.sound).
2. **Core Definitions & Fidelity (Boxes 1–2):** Verify that the verbatim Lean definitions in Box 1 match Section 2 and that each Lean signature in Box 2 matches its stated theorem.
3. **External Certificates & Written Proofs:** 2 SHA-256-pinned external artifact(s) plus any claims marked Written or Unsettled in Table 4 below.

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



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

WARN: Committed receipt (gate replay pending) | PASS: sorry/admit: 0 (per committed receipt)
PASS: native_decide: 0 (per committed receipt) | PASS: Custom axioms & meta: 0 (per committed receipt)
PASS: Coverage: 165/165 public theorems (source-level approx)
WARN: Fidelity: 9/9 exact (draft; auditor/formalizer relationship unknown) | WARN: Witnesses: pending gate replay
WARN: Human referee: pending

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

ChvatalToughness.numComponents (ChvatalToughness/Basic.lean:89-90) — c(H) (Section 1)

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

```
noncomputable def numComponents (G : SimpleGraph V) : ℕ :=
  Nat.card G.ConnectedComponent
```

ChvatalToughness.compAfterDelete (ChvatalToughness/Basic.lean:93-94) — c(G - S) (Section 1)

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

```
noncomputable def compAfterDelete (G : SimpleGraph V) (S : Finset V) : ℕ :=
  numComponents (G.induce (S : Set V)ᶜ)
```

ChvatalToughness.IsSeparatingSet (ChvatalToughness/Basic.lean:224-225) — Separating sets (Section 1)

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

```
def IsSeparatingSet (G : SimpleGraph V) (S : Finset V) : Prop :=
  2 ≤ compAfterDelete G S
```

ChvatalToughness.IsTough (ChvatalToughness/Basic.lean:229-230) — Toughness (Section 1)

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

```
def IsTough (G : SimpleGraph V) (t : ℚ) : Prop :=
  ∀ S : Finset V, IsSeparatingSet G S → t * (compAfterDelete G S : ℚ) ≤ (S.card : ℚ)
```

ChvatalToughness.HasHamiltonianPathBetween (ChvatalToughness/Basic.lean:326-327) — Hamilton paths (Section 2)

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

```
def HasHamiltonianPathBetween (G : SimpleGraph V) (u v : V) : Prop :=
  ∃ p : G.Walk u v, p.IsHamiltonian
```

ChvatalToughness.crosses (ChvatalToughness/Basic.lean:580-582) — Cuts (Section 1)

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

```
def crosses (U : Finset V) : Sym2 V → Bool :=
  Sym2.lift ⟨fun u v => decide (u ∈ U) != decide (v ∈ U),
    fun u v => by dsimp only; by_cases hu : u ∈ U <;> by_cases hv : v ∈ U <;> simp
      ↪ [hu, hv]⟩
```

ChvatalToughness.cutWeight (ChvatalToughness/Basic.lean:610-611) — Cuts (Section 1)

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

```
def cutWeight (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) (U : Finset
  ↪ V) : ℚ :=
  ∑ e ∈ G.edgeFinset.filter (fun e => crosses U e = true), f e
```

ChvatalToughness.fracDeg (ChvatalToughness/Basic.lean:614-615) — Fractional Hamilton cycles (Section 1)

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

```
def fracDeg (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) (v : V) : ℚ :=
  ∑ e ∈ G.incidenceFinset v, f e
```

ChvatalToughness.totalWeight (ChvatalToughness/Basic.lean:618-619) — Subtour program (Section 1)

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

```
def totalWeight (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) : ℚ :=
  ∑ e ∈ G.edgeFinset, f e
```

ChvatalToughness.IsUnboundedSEP (ChvatalToughness/Basic.lean:677-679) — Subtour program (Section 2)

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

```
def IsUnboundedSEP (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) : Prop :=
  (∀ e ∈ G.edgeFinset, 0 ≤ f e) ∧
  (∀ U : Finset V, U.Nonempty → U ≠ Finset.univ → 2 ≤ cutWeight G f U)
```

ChvatalToughness.IsBoundedSEP (ChvatalToughness/Basic.lean:682-683) — Subtour program (Section 2)

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

```
def IsBoundedSEP (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) : Prop :=
  IsUnboundedSEP G f ∧ ∀ e ∈ G.edgeFinset, f e ≤ 1
```

ChvatalToughness.IsSEPPolytope (ChvatalToughness/Basic.lean:687-688) — Fractional Hamilton cycles (Section 1)

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

```
def IsSEPPolytope (G : SimpleGraph V) [DecidableRel G.Adj] (f : Sym2 V → ℚ) : Prop :=
  IsBoundedSEP G f ∧ ∀ v : V, fracDeg G f v = 2
```

ChvatalToughness.leafCount (ChvatalToughness/LeafageHamiltonicity.lean:48-49) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
def leafCount (T : SimpleGraph W) [DecidableRel T.Adj] : ℕ :=
  (Finset.univ.filter (fun v : W => T.degree v = 1)).card
```

ChvatalToughness.branchCount (ChvatalToughness/LeafageHamiltonicity.lean:52-53) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
def branchCount (T : SimpleGraph W) [DecidableRel T.Adj] : ℕ :=
  (Finset.univ.filter (fun v : W => 3 ≤ T.degree v)).card
```

ChvatalToughness.edgeSubsetDeg (ChvatalToughness/LeafageHamiltonicity.lean:200-201) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
def edgeSubsetDeg (B : Finset (Sym2 W)) (v : W) : ℕ :=
  (B.filter (fun e => v ∈ e)).card
```

ChvatalToughness.leafIncidence (ChvatalToughness/PaperDraft1.lean:31-32) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
def leafIncidence (T : SimpleGraph W) [DecidableRel T.Adj] (B : Finset (Sym2 W)) : ℕ
  ↪ :=
  ∑ v ∈ Finset.univ.filter (fun v : W => T.degree v = 1), edgeSubsetDeg B v
```

ChvatalToughness.activeCompCount (ChvatalToughness/LeafageHamiltonicity.lean:250-253) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
noncomputable def activeCompCount (T : SimpleGraph W) [DecidableRel T.Adj]
    (B : Finset (Sym2 W)) : ℕ :=
  Nat.card {c : (T.deleteEdges (B : Set (Sym2 W))).ConnectedComponent |
    ∃ v : W, v ∈ c ∧ T.degree v ≤ 2}
```

ChvatalToughness.inactiveCompCount (ChvatalToughness/LeafageHamiltonicity.lean:256-259) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
noncomputable def inactiveCompCount (T : SimpleGraph W) [DecidableRel T.Adj]
    (B : Finset (Sym2 W)) : ℕ :=
  Nat.card {c : (T.deleteEdges (B : Set (Sym2 W))).ConnectedComponent |
    ∀ v : W, v ∈ c → 3 ≤ T.degree v}
```

ChvatalToughness.IsEdgeMatching (ChvatalToughness/LeafageHamiltonicity.lean:262-263) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
def IsEdgeMatching (M : Finset (Sym2 W)) : Prop :=
  ∀ e₁ ∈ M, ∀ e₂ ∈ M, ∀ v : W, v ∈ e₁ → v ∈ e₂ → e₁ = e₂
```

ChvatalToughness.ValidGrayMatching (ChvatalToughness/LeafageHamiltonicity.lean:267-271) — Trees (Section 2)

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
structure ValidGrayMatching (T : SimpleGraph W) [DecidableRel T.Adj]
    (E_gray : Finset (Sym2 W)) : Prop where
  subset_edges : E_gray ⊆ T.edgeFinset
  is_matching : IsEdgeMatching E_gray
  endpoints_deg_two : ∀ e ∈ E_gray, ∀ v : W, v ∈ e → T.degree v = 2
```

ChvatalToughness.omegaPrime (ChvatalToughness/BBVRigidity.lean:980-982) — Gadgets (Section 2)

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

```
noncomputable def omegaPrime (H : SimpleGraph V) (u v : V) (S₀ : Finset V) : ℕ :=
  Nat.card {c : (H.induce (S₀ : Set V)ᶜ).ConnectedComponent |
    ∀ x : ↑(S₀ : Set V)ᶜ, x ∈ c → x.1 ≠ u ∧ x.1 ≠ v}
```

ChvatalToughness.BBVInteriorCutCond (ChvatalToughness/BBVRigidity.lean:986-987) — Proposition on the gadget L

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

```
def BBVInteriorCutCond (H : SimpleGraph V) (u v : V) : Prop :=
  ∀ S₀ : Finset V, 1 ≤ omegaPrime H u v S₀ → 2 * omegaPrime H u v S₀ ≤ S₀.card
```

ChvatalToughness.bbvGadgetM (ChvatalToughness/BBVRigidity.lean:2975-2976) — Gadgets (Section 2)

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

```
def bbvGadgetM : SimpleGraph (Fin 7) :=
  graphOfEdgeList 7 bbvGadgetMEdges
```

ChvatalToughness.bbvGadgetMEdges (ChvatalToughness/BBVRigidity.lean:2970-2972) — Gadgets (Section 2)

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

```
def bbvGadgetMEdges : List (ℕ × ℕ) :=
  [(0, 1), (0, 2), (0, 4), (1, 2), (1, 4), (1, 5), (1, 6),
   (2, 4), (2, 5), (3, 4), (3, 6), (4, 6)]
```

ChvatalToughness.bbvGadgetL (ChvatalToughness/BBVRigidity.lean:2987-2988) — Gadgets (Section 2)

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

```
def bbvGadgetL : SimpleGraph (Fin 8) :=
  graphOfEdgeList 8 bbvGadgetLEdges
```

ChvatalToughness.GenBBVVert (ChvatalToughness/BBVRigidity.lean:3584-3585) — Two-port graphs (Section 1)

```
abbrev GenBBVVert (ℓ c m : ℕ) (wSize : Fin m → ℕ) : Type :=
  Fin ℓ ⊕ Fin c ⊕ (Σ i : Fin m, Fin (wSize i))
```

ChvatalToughness.LocalGadgetVert (ChvatalToughness/BBVRigidity.lean:3592-3592) — Two-port graphs (Section 1)

```
abbrev LocalGadgetVert (w : ℕ) : Type := Bool ⊕ Fin w
```

ChvatalToughness.localToGen (ChvatalToughness/BBVRigidity.lean:3595-3600) — Two-port graphs (Section 1)

```
def localToGen {ℓ c m : ℕ} {wSize : Fin m → ℕ}
    (portU portV : Fin m → Fin c) (i : Fin m) :
    LocalGadgetVert (wSize i) → GenBBVVert ℓ c m wSize
  | .inl false => .inr (.inl (portU i))
  | .inl true => .inr (.inl (portV i))
  | .inr w => .inr (.inr ⟨i, w⟩)
```

ChvatalToughness.localGadgetGraph (ChvatalToughness/BBVRigidity.lean:3614-3630) — Two-port graphs (Section 1)

```
def localGadgetGraph {ℓ c m : ℕ} {wSize : Fin m → ℕ}
    (G : SimpleGraph (GenBBVVert ℓ c m wSize))
    (portU portV : Fin m → Fin c) (i : Fin m) :
    SimpleGraph (LocalGadgetVert (wSize i)) where
  Adj x y := match x, y with
    | .inl _, .inl _ => False
    | x, y => G.Adj (localToGen portU portV i x) (localToGen portU portV i y)
  symm := ⟨fun x y h => by
    cases x <;> cases y
    · exact h
    · exact h.symm
    · exact h.symm
    · exact h.symm⟩
  loopless := ⟨fun x h => by
    cases x with
    | inl _ => exact h
    | inr w => exact G.loopless.irefl _ h⟩
```

ChvatalToughness.IsGenBBVGraph (ChvatalToughness/BBVRigidity.lean:3634-3647) — Two-port graphs (Section 1)

```
structure IsGenBBVGraph {ℓ c m : ℕ} {wSize : Fin m → ℕ}
    (G : SimpleGraph (GenBBVVert ℓ c m wSize))
    (portU portV : Fin m → Fin c) : Prop where
  ell_pos : 1 ≤ ℓ
  c_ge_two : 2 ≤ c
  m_ge : 2 * ℓ + 1 ≤ m
  w_pos : ∀ i : Fin m, 1 ≤ wSize i
  ports_ne : ∀ i : Fin m, portU i ≠ portV i
  no_inter_W : ∀ (i j : Fin m) (a : Fin (wSize i)) (b : Fin (wSize j)),
    i ≠ j → ¬G.Adj (.inr (.inr ⟨i, a⟩)) (.inr (.inr ⟨j, b⟩))
  only_ports : ∀ (i : Fin m) (a : Fin (wSize i)) (x : Fin c),
    G.Adj (.inr (.inr ⟨i, a⟩)) (.inr (.inl x)) → x = portU i ∨ x = portV i
  no_ham_path : ∀ i : Fin m,
    ¬HasHamiltonianPathBetween (localGadgetGraph G portU portV i) (.inl false) (.inl
      ↪ true)
```

ChvatalToughness.BBVLVert (ChvatalToughness/BBVRigidity.lean:4896-4896) — G(L; l, m) (Section 1)

```
abbrev BBVLVert (ℓ m : ℕ) : Type := Fin ℓ ⊕ (Fin m × Fin 8)
```

ChvatalToughness.bbvLAdj (ChvatalToughness/BBVRigidity.lean:4902-4910) — G(L; l, m) (Section 1)

```
def bbvLAdj (ℓ m : ℕ) : BBVLVert ℓ m → BBVLVert ℓ m → Prop
  | .inl k₁, .inl k₂ => k₁ ≠ k₂
  | .inl _, .inr _ => True
  | .inr _, .inl _ => True
  | .inr (i, u), .inr (j, v) =>
      if i = j then
        bbvGadgetL.Adj u v ∨ (u = 0 ∧ v = 1) ∨ (u = 1 ∧ v = 0)
      else
        (u = 0 ∨ u = 1) ∧ (v = 0 ∨ v = 1)
```

ChvatalToughness.bbvLGraph (ChvatalToughness/BBVRigidity.lean:4913-4945) — G(L; l, m) (Section 1)

```
def bbvLGraph (ℓ m : ℕ) : SimpleGraph (BBVLVert ℓ m) where
  Adj := bbvLAdj ℓ m
  symm := ⟨fun x y h => by
    cases x with
    | inl k₁ =>
      cases y with
      | inl k₂ => exact Ne.symm h
      | inr _ => trivial
    | inr p₁ =>
      cases y with
      | inl _ => trivial
      | inr p₂ =>
        rcases p₁ with ⟨i, u⟩; rcases p₂ with ⟨j, v⟩
        dsimp only [bbvLAdj] at h ⊢
        by_cases hij : i = j
        · subst hij; simp only [↓reduceIte] at h ⊢
          rcases h with h1 | ⟨rfl, rfl⟩ | ⟨rfl, rfl⟩
          · exact Or.inl h1.symm
          · exact Or.inr (Or.inr ⟨rfl, rfl⟩)
          · exact Or.inr (Or.inl ⟨rfl, rfl⟩)
        · have hji : j ≠ i := Ne.symm hij
          simp only [hij, hji, ↓reduceIte] at h ⊢
          exact ⟨h.2, h.1⟩⟩
  loopless := ⟨fun x h => by
    cases x with
    | inl k => exact h rfl
    | inr p =>
      rcases p with ⟨i, u⟩
      simp only [bbvLAdj, ↓reduceIte] at h
      rcases h with h1 | ⟨h1, h2⟩ | ⟨h1, h2⟩
      · exact bbvGadgetL.loopless.irrefl u h1
      · rw [h1] at h2; cases h2
      · rw [h1] at h2; cases h2⟩
```

ChvatalToughness.ChordalBBVVert (ChvatalToughness/SEPPhaseTransition.lean:148-148) — G(M; l, m) (Section 1)

```
abbrev ChordalBBVVert (ℓ m : ℕ) : Type := Fin ℓ ⊕ (Fin m × Fin 7)
```

ChvatalToughness.chordalBBVAdj (ChvatalToughness/SEPPhaseTransition.lean:154-162) — G(M; l, m) (Section 1)

```
def chordalBBVAdj (ℓ m : ℕ) : ChordalBBVVert ℓ m → ChordalBBVVert ℓ m → Prop
  | .inl k₁, .inl k₂ => k₁ ≠ k₂
  | .inl _, .inr _ => True
  | .inr _, .inl _ => True
  | .inr (i, u), .inr (j, v) =>
      if i = j then
        bbvGadgetM.Adj u v
      else
        (u = 0 ∨ u = 1) ∧ (v = 0 ∨ v = 1)
```

ChvatalToughness.ChordalBBV (ChvatalToughness/SEPPhaseTransition.lean:165-190) — G(M; l, m) (Section 1)

```
def ChordalBBV (ℓ m : ℕ) : SimpleGraph (ChordalBBVVert ℓ m) where
  Adj := chordalBBVAdj ℓ m
  symm := ⟨fun x y h => by
    cases x with
    | inl k₁ =>
      cases y with
      | inl k₂ => exact Ne.symm h
      | inr _ => trivial
    | inr p₁ =>
      cases y with
      | inl _ => trivial
      | inr p₂ =>
        rcases p₁ with ⟨i, u⟩; rcases p₂ with ⟨j, v⟩
        dsimp only [chordalBBVAdj] at h ⊢
        by_cases hij : i = j
        · subst hij; simp only [↓reduceIte] at h ⊢; exact h.symm
        · have hji : j ≠ i := Ne.symm hij
          simp only [hij, hji, ↓reduceIte] at h ⊢
          exact ⟨h.2, h.1⟩⟩
  loopless := ⟨fun x h => by
    cases x with
    | inl k => exact h rfl
    | inr p =>
      rcases p with ⟨i, u⟩
      simp only [chordalBBVAdj, ↓reduceIte] at h
      exact bbvGadgetM.loopless.irrefl u h⟩
```

ChvatalToughness.HasPerfectEliminationOrdering (ChvatalToughness/PaperDraft1.lean:254-256) — Chordality (Section 2)

```
def HasPerfectEliminationOrdering {V : Type*} (G : SimpleGraph V) : Prop :=
  ∃ rank : V → ℕ, Function.Injective rank ∧
    ∀ u v w : V, G.Adj u v → G.Adj u w → v ≠ w → rank u < rank v → rank u < rank w →
      ↪ G.Adj v w
```

ChvatalToughness.ChvatalSplitVert (ChvatalToughness/SplitAndExtremals.lean:45-46) — Split graphs (Section 1)

```
abbrev ChvatalSplitVert (ℓ : ℕ) : Type :=
  Fin ℓ ⊕ Fin (2 * ℓ + 1) ⊕ Fin (2 * ℓ + 1)
```

ChvatalToughness.chvatalSplitAdj (ChvatalToughness/SplitAndExtremals.lean:54-61) — Split graphs (Section 1)

```
def chvatalSplitAdj (ℓ : ℕ) : ChvatalSplitVert ℓ → ChvatalSplitVert ℓ → Prop
  | .inl a₁, .inl a₂ => a₁ ≠ a₂
  | .inl _, .inr _ => True
  | .inr _, .inl _ => True
  | .inr (.inl r₁), .inr (.inl r₂) => r₁ ≠ r₂
  | .inr (.inl r₁), .inr (.inr r₂) => r₁ = r₂
  | .inr (.inr r₁), .inr (.inl r₂) => r₁ = r₂
  | .inr (.inr _), .inr (.inr _) => False
```

ChvatalToughness.ChvatalSplit (ChvatalToughness/SplitAndExtremals.lean:64-81) — Split graphs (Section 1)

```
def ChvatalSplit (ℓ : ℕ) : SimpleGraph (ChvatalSplitVert ℓ) where
  Adj := chvatalSplitAdj ℓ
  symm := ⟨fun x y h => by
    rcases x with a₁ | (b₁ | i₁) <;> rcases y with a₂ | (b₂ | i₂)
    · exact Ne.symm h
    · trivial
    · trivial
    · trivial
    · exact Ne.symm h
    · exact h.symm
    · trivial
    · exact h.symm
    · exact h⟩
  loopless := ⟨fun x h => by
    rcases x with a | (b | i)
    · exact h rfl
    · exact h rfl
    · exact h⟩
```

ChvatalToughness.petersenGraph (ChvatalToughness/BBVRigidity.lean:160-161) — Petersen graph (Section 3)

```
def petersenGraph : SimpleGraph (Fin 10) :=
  graphOfEdgeList 10 petersenEdges
```

ChvatalToughness.petersenEdges (ChvatalToughness/BBVRigidity.lean:154-157) — Petersen graph (Section 3)

```
def petersenEdges : List (ℕ × ℕ) :=
  [(0, 1), (1, 2), (2, 3), (3, 4), (4, 0),
   (5, 7), (7, 9), (9, 6), (6, 8), (8, 5),
   (0, 5), (1, 6), (2, 7), (3, 8), (4, 9)]
```

ChvatalToughness.bbvGadgetLEdges (ChvatalToughness/BBVRigidity.lean:2982-2984) — Gadgets (Section 2)

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

```
def bbvGadgetLEdges : List (ℕ × ℕ) :=
  [(0, 3), (0, 4), (1, 2), (1, 5), (2, 3), (2, 5), (2, 7),
   (3, 4), (3, 7), (4, 5), (4, 6), (5, 6)]
```

ChvatalToughness.edgeListAdjBool (ChvatalToughness/Basic.lean:47-48) — Gadgets (Section 2)

```
def edgeListAdjBool (edges : List (ℕ × ℕ)) (u v : ℕ) : Bool :=
  (u != v) && edges.any (fun p => (p.1 == u && p.2 == v) || (p.1 == v && p.2 == u))
```

ChvatalToughness.graphOfEdgeList (ChvatalToughness/Basic.lean:71-74) — Gadgets (Section 2)

```
def graphOfEdgeList (n : ℕ) (edges : List (ℕ × ℕ)) : SimpleGraph (Fin n) where
  Adj u v := edgeListAdjBool edges u.val v.val = true
  symm := ⟨fun u v h => by rw [edgeListAdjBool_symm]; exact h⟩
  loopless := ⟨fun u h => by simp [edgeListAdjBool_irrefl] at h⟩
```

Selected Concrete Sanity Examples (checked in Lean; examples only, not global identity proofs): ChvatalToughness.sanity_gadgetL_edges (bbvGadgetL.edgeFinset.card = 12); ChvatalToughness.sanity_gadgetM_edges (bbvGadgetM.edgeFinset.card = 12); ChvatalToughness.sanity_leafCount (leafCount extremalLeafNeckTree = 4); ChvatalToughness.sanity_branchCount (branchCount extremalLeafNeckTree = 2).

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

ChvatalToughness.paper_discharge (ChvatalToughness/PaperDraft1.lean:35-248) src:d5626c368e1e2b

Context binders in scope: `variable {W : Type*} [Fintype W] [DecidableEq W]`

```
theorem paper_discharge {T : SimpleGraph W} [DecidableRel T.Adj]
    (hT : T.IsTree) (hcard : 2 ≤ Fintype.card W)
    {E_gray B F_gray : Finset (Sym2 W)}
    (hgray : ValidGrayMatching T E_gray)
    (hB_sub : B ⊆ T.edgeFinset) (hB_ne : B.Nonempty)
    (hF_sub : F_gray ⊆ E_gray)
    (hF_touch : ∀ g ∈ F_gray, ∃ b ∈ B, ∃ v : W, v ∈ g ∧ v ∈ b) :
    let r := activeCompCount T B
    let s := inactiveCompCount T B
    let Q := 2 * B.card + F_gray.card
    r + s = B.card + 1 ∧
    2 ≤ r ∧ s + 2 ≤ r ∧ s ≤ branchCount T ∧ branchCount T + 2 ≤ leafCount T ∧
    Q + 4 + leafIncidence T B ≤ 4 * r + s ∧ Q + 6 ≤ 5 * r ∧
    Q + 4 ≤ 4 * r + branchCount T ∧ Q + 6 ≤ 4 * r + leafCount T
```

Assumptions: T is a tree on a finite vertex type with at least two elements, with decidable vertex equality and decidable adjacency; E_gray is a matching of edges of T whose endpoints have degree 2; B is a nonempty set of edges of T; F_gray is a subset of E_gray, each member of which shares a vertex with a member of B. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_bbv_orders (ChvatalToughness/PaperDraft1.lean:814-829) src:0446afd26743cf

```
theorem paper_bbv_orders :
    (∀ {ℓ c m : ℕ} {wSize : Fin m → ℕ} {G : SimpleGraph (GenBBVVert ℓ c m wSize)}
      {portU portV : Fin m → Fin c}, IsGenBBVGraph G portU portV →
      ¬G.IsHamiltonian ∧
      (IsTough G (3 / 2) → 25 ≤ Fintype.card (GenBBVVert ℓ c m wSize)) ∧
      (IsTough G 2 → 42 ≤ Fintype.card (GenBBVVert ℓ c m wSize)) ∧
      (∀ t : ℚ, 2 < t → IsTough G t → 59 ≤ Fintype.card (GenBBVVert ℓ c m wSize))) ∧
    (Fintype.card (BBVLVert 1 3) = 25 ∧ ¬(bbvLGraph 1 3).IsHamiltonian ∧
      IsTough (bbvLGraph 1 3) (3 / 2)) ∧
    (Fintype.card (BBVLVert 2 5) = 42 ∧ ¬(bbvLGraph 2 5).IsHamiltonian ∧
      IsTough (bbvLGraph 2 5) 2) ∧
    (Fintype.card (BBVLVert 3 7) = 59 ∧ ¬(bbvLGraph 3 7).IsHamiltonian ∧
      IsTough (bbvLGraph 3 7) (31 / 15) ∧ (2 : ℚ) < 31 / 15)
```

Assumptions: A closed conjunction of four statements. The first is universal: G is any graph on K + C + (the union of the W_i) that satisfies IsGenBBVGraph: l >= 1, |C| >= 2, m >= 2l + 1, every W_i nonempty, two distinct ports per gadget, no edges between different W_i, no edges from W_i to C other than to its ports, and no Hamilton path between the ports in the gadget. The three order bounds respectively assume 3/2-toughness, 2-toughness, and t-toughness for a rational t > 2. The other three are statements about the fixed graphs G(L; 1, 3), G(L; 2, 5) and G(L; 3, 7): their orders 25, 42 and 59, non-Hamiltonicity, and that they are 3/2-tough, 2-tough and 31/15-tough, with 31/15 > 2. Applicability: Unconditional (closed statement; no outer premise witness needed).

ChvatalToughness.paper_chordalBBV (ChvatalToughness/PaperDraft1.lean:795-804) src:537639790557b7

```
theorem paper_chordalBBV {ℓ m : ℕ} (hℓ : 1 ≤ ℓ) :
    Fintype.card (ChordalBBVVert ℓ m) = ℓ + 7 * m ∧
    HasPerfectEliminationOrdering (ChordalBBV ℓ m) ∧
    (∀ f : Sym2 (ChordalBBVVert ℓ m) → ℚ, IsUnboundedSEP (ChordalBBV ℓ m) f →
      29 * (m : ℚ) / 4 ≤ totalWeight (ChordalBBV ℓ m) f) ∧
    (2 * ℓ + 1 ≤ m → ¬(ChordalBBV ℓ m).IsHamiltonian ∧
      ((∃ f : Sym2 (ChordalBBVVert ℓ m) → ℚ, IsSEPPolytope (ChordalBBV ℓ m) f) ↔ m ≤
        ↪ 4 * ℓ))
```

Assumptions: l and m are natural numbers with 1 <= l; the weight bound assumes that f is a rational feasible point of the subtour program; the non-Hamiltonicity and the equivalence assume 2l + 1 <= m. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_chordal_not_fractional (ChvatalToughness/PaperDraft1.lean:773-789) src:ee24c177d5c534

```
theorem paper_chordal_not_fractional (ε : ℚ) (hε : 0 < ε) :
    ∃ ℓ m : ℕ, 2 ≤ ℓ ∧ m = 4 * ℓ + 1 ∧
      HasPerfectEliminationOrdering (ChordalBBV ℓ m) ∧
      IsTough (ChordalBBV ℓ m) (13 / 8 - ε) ∧
      ¬∃ f : Sym2 (ChordalBBVVert ℓ m) → ℚ, IsSEPPolytope (ChordalBBV ℓ m) f
```

Assumptions: eps is a rational number with 0 < eps. The conclusion chooses natural l >= 2 and m = 4l + 1, asserts an elimination ordering, and excludes a rational fractional Hamilton cycle. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_split (ChvatalToughness/PaperDraft1.lean:848-861) src:8b6ce2c7694e1c

```
theorem paper_split {ℓ : ℕ} (hℓ : 1 ≤ ℓ) :
    Fintype.card (ChvatalSplitVert ℓ) = 5 * ℓ + 2 ∧
    ¬(ChvatalSplit ℓ).IsHamiltonian ∧
    IsTough (ChvatalSplit ℓ) ((3 * (ℓ : ℚ)) / (2 * (ℓ : ℚ) + 1)) ∧
    (∀ t : ℚ, IsTough (ChvatalSplit ℓ) t → t ≤ (3 * (ℓ : ℚ)) / (2 * (ℓ : ℚ) + 1)) ∧
    (∃ f, IsBoundedSEP (ChvatalSplit ℓ) f ∧
      totalWeight (ChvatalSplit ℓ) f = (5 * (ℓ : ℚ) + 2) + 1 / 2) ∧
    (∀ f, IsUnboundedSEP (ChvatalSplit ℓ) f →
      (5 * (ℓ : ℚ) + 2) + 1 / 2 ≤ totalWeight (ChvatalSplit ℓ) f) ∧
    (¬∃ f, IsSEPPolytope (ChvatalSplit ℓ) f)
```

Assumptions: l is a natural number with 1 <= l. The toughness upper bound ranges over rational t for which the graph is t-tough. The weight lower bound ranges over feasible rational weightings, and the attaining weighting is an existential witness with edge weights at most one. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_gadget (ChvatalToughness/PaperDraft1.lean:832-839) src:9995581a8aeac0

```
theorem paper_gadget {V : Type*} [Fintype V] [DecidableEq V] {H : SimpleGraph V} {u v
  ↪ : V}
    (huv : u ≠ v) (hlo : 3 ≤ Fintype.card V) (hno : ¬HasHamiltonianPathBetween H u v)
      ↪ :
    (Fintype.card V ≤ 5 →
      ∃ S₀ : Finset V, 1 ≤ omegaPrime H u v S₀ ∧ S₀.card ≤ omegaPrime H u v S₀) ∧
    (Fintype.card V ≤ 7 →
      ∃ S₀ : Finset V, 1 ≤ omegaPrime H u v S₀ ∧ S₀.card + 1 ≤ 2 * omegaPrime H u v
        ↪ S₀)
```

Assumptions: H is a simple graph on a finite vertex type with at least 3 elements, u and v are distinct vertices, and H has no Hamilton path from u to v. Vertex equality is decidable. The two existence clauses additionally assume |V(H)| <= 5 and |V(H)| <= 7, respectively, and each chooses its own vertex subset S. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_gadgetL (ChvatalToughness/PaperDraft1.lean:843-845) src:cdf3b5e3003f5f

```
theorem paper_gadgetL : ¬HasHamiltonianPathBetween bbvGadgetL 0 1 ∧
    BBVInteriorCutCond bbvGadgetL 0 1
```

Assumptions: The graph bbvGadgetL and the ports 0 and 1 are fixed. The cut clause ranges over every vertex subset S with omega'(S) >= 1. Applicability: Unconditional (closed statement; no outer premise witness needed).

ChvatalToughness.paper_bbvL_sep (ChvatalToughness/PaperDraft1.lean:807-811) src:4655595cde7b9d

```
theorem paper_bbvL_sep {ℓ m : ℕ} (hℓ : 1 ≤ ℓ) (hm : 2 * ℓ + 1 ≤ m) :
    Fintype.card (BBVLVert ℓ m) = ℓ + 8 * m ∧ ¬(bbvLGraph ℓ m).IsHamiltonian ∧
    ∃ f : Sym2 (BBVLVert ℓ m) → ℚ, IsSEPPolytope (bbvLGraph ℓ m) f
```

Assumptions: l and m are natural numbers with 1 <= l and 2l + 1 <= m. The fractional Hamilton cycle is existentially quantified in the conclusion. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

ChvatalToughness.paper_petersen (ChvatalToughness/PaperDraft1.lean:864-868) src:837a649c14b997

```
theorem paper_petersen : ¬petersenGraph.IsHamiltonian ∧ IsTough petersenGraph (4 / 3)
  ↪ ∧
    (∀ t : ℚ, IsTough petersenGraph t → t ≤ 4 / 3) ∧
    ∃ f : Sym2 (Fin 10) → ℚ, IsSEPPolytope petersenGraph f
```

Assumptions: A closed statement about one fixed graph; there are no external parameters and no witness to supply. The arrow in Sym2 (Fin 10) -> Q belongs to the type of the existentially quantified weighting. The upper bound ranges over rational t for which the Petersen graph is t-tough, and the fractional Hamilton cycle is existential. Applicability: Instantiable — closed instantiation witness PENDING gate-box replay.

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

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`paper_bbv_orders`	`nat_div`	`/ 2`	intended_convention	The toughness bounds 3/2 in IsTough G (3 / 2) have type Rat, the type of the second argument of IsTough; division is exact.
`paper_bbv_orders`	`nat_div`	`/ 15`	intended_convention	31 / 15 is the rational toughness argument of IsTough and is compared in Rat; division is exact.
`paper_chordalBBV`	`nat_div`	`/ 4`	intended_convention	29 * (m : Rat) / 4 is a rational number because m is cast to Rat; division is exact.
`paper_chordal_not_fractional`	`nat_div`	`/ 8`	intended_convention	13 / 8 - eps is the rational toughness argument of IsTough; division is exact.
`paper_chordal_not_fractional`	`nat_sub`	`- ε`	intended_convention	eps has type Rat, so the subtraction is ordinary; a negative value makes the toughness clause weaker, not vacuous for small eps.

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`paper_petersen`	`nat_div`	`/ 3`	intended_convention	4 / 3 is the rational toughness argument of IsTough; division is exact.
`paper_split`	`nat_div`	`/ (2 * (ℓ : ℚ) + 1)`	intended_convention	The expression 3 * (l : Rat) / (2 * (l : Rat) + 1) is rational because l is cast to Rat; division is exact and the denominator is positive.
`paper_split`	`nat_div`	`/ 2`	intended_convention	(5 * (l : Rat) + 2) + 1 / 2 is a rational number because l is cast to Rat; division is exact.
`instDecidableRelPetersen`	`local_instance_or_notation`	`instDecidableRelPetersen`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelBBVGadgetM`	`local_instance_or_notation`	`instDecidableRelBBVGadgetM`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelBBVGadgetL`	`local_instance_or_notation`	`instDecidableRelBBVGadgetL`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelBBVLGraph`	`local_instance_or_notation`	`instDecidableRelBBVLGraph`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelGraphOfEdgeList`	`local_instance_or_notation`	`instDecidableRelGraphOfEdgeList`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelExtremalLeafNeckTree`	`local_instance_or_notation`	`instDecidableRelExtremalLeafNeckTree`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelChordalBBV`	`local_instance_or_notation`	`instDecidableRelChordalBBV`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelChvatalSplit`	`local_instance_or_notation`	`instDecidableRelChvatalSplit`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.
`instDecidableRelHerschel`	`local_instance_or_notation`	`instDecidableRelHerschel`	intended_convention	A decidability instance for an adjacency relation. It is used to enumerate edges, degrees and cuts of concrete finite graphs and does not change the meaning of a statement.

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

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

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 1: Let T be a tree on at least two vertices with L leaves and b branch vertices, let the gray edges be a matching of T whose edges have both endpoints of degree 2, let B be a nonempty set of edges of T and Phi a set of gray edges each of which shares a vertex with an edge of B. Let r and s be the numbers of active and inactive components of T - B, lambda the number of pairs (v, e) with v a leaf, e in B and v in e, and Q = 2|B| + |Phi|. Then r + s = |B| + 1, r >= 2, s <= r - 2, s <= b <= L - 2, Q <= 4r + s - 4 - lambda, Q <= 5r - 6, Q <= 4r + b - 4 and Q <= 4r + L - 6	Lean Kernel	`ChvatalToughness.paper_discharge`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Thm. 2: Let G be a chordal graph on at least three vertices with a tree representation whose host tree has L leaves and b branch vertices. (i) If G is 4-tough and at least one of L <= 6, b <= 4 holds, then G is Hamiltonian. (ii) If L >= 7 and tau(G) > 5 - 7/L, then G is Hamiltonian. (iii) If G is 4-tough and at least one of L <= 4, b <= 2 holds, then G is Hamilton-connected. (iv) If L >= 5 and tau(G) > 5 - 5/L, then G is Hamilton-connected	Written	Written proof	Lean: none; fidelity: n/a	Written proof from thm_discharge (Lean), lem_host, lem_hall and lem_sdr, with Theorem 1.1 (Hall theorem) and Lemma 3.3 of the preprint ar Xiv:2608.10193v2, which are not reproved here
Thm. 3: Let G be a chordal graph with at least one vertex and a tree representation whose host tree has L leaves. If G is 4-tough and L <= 11, or if L >= 12 and tau(G) > 5 - 7/(L - 5), then G has a Hamilton path	Written	Written proof	Lean: none; fidelity: n/a	Written proof from thm_discharge (Lean), lem_host and lem_hall, with Theorem 1.1 of the preprint arXiv:2608.10193v2, which is not reproved here
Thm. 4: Every two-port graph with parameters l >= 1 and m >= 2l + 1 is non-Hamiltonian; it has at least 25 vertices if it is 3/2-tough, at least 42 if it is 2-tough, and at least 59 if it is t-tough for some rational t > 2. The graphs G(L; 1, 3), G(L; 2, 5) and G(L; 3, 7) are non-Hamiltonian, have orders 25, 42 and 59, and are 3/2-tough, 2-tough and 31/15-tough	Lean Kernel	`ChvatalToughness.paper_bbv_orders`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 5: Let l >= 1 and m >= 0. The graph G(M; l, m) has l + 7m vertices and is chordal, and every feasible point of its subtour program has total weight at least 29m/4. If m >= 2l + 1, then G(M; l, m) is non-Hamiltonian, and it is fractionally Hamiltonian if and only if m <= 4l	Lean Kernel	`ChvatalToughness.paper_chordalBBV`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Thm. 6: For every rational eps > 0 there are l >= 2 and m = 4l + 1 such that G(M; l, m) is chordal, (13/8 - eps)-tough, and not fractionally Hamiltonian	Lean Kernel	`ChvatalToughness.paper_chordal_not_fractional`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Thm. 7: For every l >= 1 the split graph Sp_l has n = 5l + 2 vertices, is non-Hamiltonian, and has toughness exactly 3l/(2l + 1). Some feasible point of its subtour program with x <= 1 has total weight n + 1/2, and every feasible point of its subtour program has total weight at least n + 1/2. Sp_l is not fractionally Hamiltonian	Lean Kernel	`ChvatalToughness.paper_split`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Thm. 8: For n = 3, ..., 11 the largest toughness of a non-Hamiltonian graph of order n is 1/2, 1/2, 2/3, 2/3, 1, 1, 1, 4/3, 1. The Petersen graph is the only non-Hamiltonian graph of order between 3 and 11 with toughness greater than 1	Cert Only	census.py (94d11a4b77c4) [regenerable; replay pending]; recompute.py (5369a0778499) [regenerable; replay pending]	External cert only; fidelity: n/a	Exhaustive generation with geng of nauty 2.8.9 and exact tests of every graph (cert/census.py); the reduction to 2-connected graphs of minimum degree at least 3 for n = 11 and the witness for n >= 7 are written proofs
Prop. 9: For every L >= 3 there are a tree with L leaves and L - 2 branch vertices, a gray matching, and sets B and Phi with lambda = 0 and Q = 5r - 6 = 4r + s - 4 = 4r + b - 4 = 4r + L - 6	Written	Written proof	Lean: none; fidelity: n/a	Written proof for every L >= 3; recompute.py checks the trees for 3 <= L <= 12 only, as a check of the written proof
Lem. 10: If G has a tree representation whose host tree has L leaves and b branch vertices, it has one whose host tree has at most L leaves and at most b branch vertices and in which no bag is contained in the bag of an adjacent node; if that host tree has at least two nodes, every leaf z satisfies F_x = {z} for some vertex x	Written	Written proof	Lean: none; fidelity: n/a	Written proof
Lem. 11: (i) If A(F_0) has no matching of size q(F_0), some C with |C| <= q(F_0) - 1 meets every edge of A(F_0). (ii) For C, B, D and S as in the lemma, |S| = |C| + |D| and c(G - S) >= r + |D|	Written	Written proof	Lean: none; fidelity: n/a	Written proof, with Konig's theorem; part (i) is the argument of arXiv:2608.10193v2, Section 3.4

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Lem. 12: If the system (A_e) of a chordal graph that is not complete has a q-SDR for q = 2 on black and q = 1 on gray edges, then the graph has a Hamilton cycle or at most two vertices	Written	Written proof	Lean: none; fidelity: n/a	Written proof
Prop. 13: Every H(l, a, b) is chordal and non-Hamiltonian and has leafage exactly 2a + b	Written	Written proof	Lean: none; fidelity: n/a	Written proof for every l, a, b; it uses that M has no Hamilton path between its ports, which is the Lean lemma bbvGadgetM_no_ham_path (BBVRigidity.lean) and is also checked by recompute.py
Prop. 14: For 3 <= L <= 14 the largest toughness of a graph H(l, a, b) with 2a + b = L is the one in the table of examples, attained by the graph listed there. In particular H(2, 4, 1) is a chordal non-Hamiltonian graph of order 32, leafage 9 and toughness exactly 3/2	Cert Only	recompute.py (5369a0778499) [regenerable; replay pending]	External cert only; fidelity: n/a	Exact computation by recompute.py; chordality, non-Hamiltonicity and leafage by prop_hybrid
Prop. 15: Let H be a graph on at least 3 vertices with no Hamilton path from u to v, where u and v are distinct. If |V(H)| <= 5, some S has omega'(S) >= 1 and |S| <= omega'(S). If |V(H)| <= 7, some S has omega'(S) >= 1 and |S| <= 2 omega'(S) - 1	Lean Kernel	`ChvatalToughness.paper_gadget`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Prop. 16: The gadget L has no Hamilton path from 0 to 1, and |S| >= 2 omega'(S) for every S with omega'(S) >= 1	Lean Kernel	`ChvatalToughness.paper_gadgetL`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Prop. 17: For l >= 1 and m >= 2l + 1 the graph G(L; l, m) has l + 8m vertices, is non-Hamiltonian, and is fractionally Hamiltonian	Lean Kernel	`ChvatalToughness.paper_bbvL_sep`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.
Prop. 18: The Petersen graph is non-Hamiltonian, has toughness exactly 4/3, and is fractionally Hamiltonian	Lean Kernel	`ChvatalToughness.paper_petersen`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel; axioms propext, Classical.choice, Quot.sound; definitions of Section 2; hypotheses as stated.

[a] Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: bac987fd639d | Lean mode: committed_receipt | Closure SHA-256: 679e95ebf2a1 | Git anchor: b98496b0cde4 | Axioms SHA-256: 855e1acb2584 | Fidelity SHA-256: b62f420a28ba.

6. PRIOR WORK AND OPEN QUESTIONS

In the references, [to check] marks a source not read in full; what we say about it comes from reviews, abstracts and citing papers.

Chordal graphs. Theorem 2 refines the argument with which Huang [8] proves that 5-tough chordal graphs are Hamilton-connected: his Theorem 1.1 and the construction of the separating set are used as they are, and his Lemma 3.1, which he takes from Kabela and Kaiser [9], is replaced by Theorem 1. His argument gives $|S| \leq 5c - 6$ where (3) gives $|S| \leq \min(5c - 7, 4c + L - 7)$. Related classes with better constants are split graphs (3/2) [13], spider graphs (3/2) [10], planar chordal graphs (toughness greater than 1) [3], and interval graphs (1) [11]; chordal graphs of leafage at most three are spider graphs, so the cycle conclusions (i) and (ii) of Theorem 2 are weaker than [10] there. Conclusions (iii) and (iv) are weaker than [2] at leafage two; at leafage three we found no source. Huang conjectures that 2-tough chordal graphs on at least three vertices are Hamilton-connected [8, Conjecture 3.5].

Two-port graphs. The construction and the gadgets are those of [1]. The definition of two-port graphs is wider: unequal gadgets, shared ports, and any edges inside $K \cup C$ and between K and the gadgets. Theorem 4 says nothing about 2-tough non-Hamiltonian graphs outside this class.

Fractional Hamilton cycles. Wang [18] proves that a connected graph on at least three vertices that is not fractionally Hamiltonian has a non-Hamiltonian chordal spanning supergraph, and attributes to Scheinerman and Ullman [17] graphs that are not fractionally Hamiltonian with toughness approaching 3/2 from below; Theorem 6 raises 3/2 to 13/8.

Small orders. That the Petersen graph is 4/3-tough and non-Hamiltonian is classical. The README and file listing of Kemp's code [12] show an enumeration of tough non-Hamiltonian graphs and no table of results; we read neither code nor thesis.

The search. The folder `prior-work-search` holds the log of a search run on 25 November 2026 through Google Scholar, CORE, OEIS, arXiv, zbMATH Open, Semantic Scholar, Math.SE, MathOverflow, Zenodo and GitHub. The statements of Theorems 1–8 were not found in the searches logged; `litsearch.py check` passes on the log (`prior-work-search/check.out`). Not searched: OpenAlex (stopped by its host), MathSciNet, books in print, and full texts of paywalled sources [to check].

Open questions. (1) Does leafage at most L allow a constant below 4 for some $L \geq 4$? (2) Is there a 2-tough non-Hamiltonian graph of order 12 to 41? (3) What is the infimum of the t for which every t-tough chordal graph on at least three vertices is fractionally Hamiltonian? Theorem 6 and [8] place it between 13/8 and 5. (4) What is $\tau^{\mathrm{NH}}(12)$?

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

Files. The paper folder holds the programs (`recompute.py`, `cert`) and their outputs, the figure scripts, the ledgers `claims.md` and `references-checked.md`, the statement audit, the referee reports and the search log. The Lean library is `findings/chvatal-toughness/lean`; `PaperDraft1.lean` holds the statements cited here.

Mathematics written for this draft. The proofs of Sections 3.2–3.4, the leaf term λ, the census for orders 3 to 11, Theorem 6, and Propositions 9, 13 and 14 were written or rewritten by the author agent of this draft from a finding record; `README.md` lists them with their status, and the claims of the finding not kept here.

Use of AI. AI research agents built and directed by Tyler B. Hughes found the results, wrote the proofs, the Lean library, the programs and this text, and refereed it; author and referee agents came from different model families. No human mathematician has yet refereed the proof.

REFERENCES

(1) D. Bauer, H. J. Broersma, H. J. Veldman, Not every 2-tough graph is Hamiltonian, *Discrete Applied Mathematics* **99** (2000), 317–321. [to check]

(2) H. Broersma, J. Fiala, P. A. Golovach, T. Kaiser, D. Paulusma, A. Proskurowski, Linear-time algorithms for scattering number and Hamilton-connectivity of interval graphs, arXiv:1301.5953 (2013). [to check]

(3) T. Böhme, J. Harant, M. Tkáč, More than one tough chordal planar graphs are Hamiltonian, *Journal of Graph Theory* **32** (1999), 405–410. [to check]

(4) G. Chen, M. S. Jacobson, A. E. Kézdy, J. Lehel, Tough enough chordal graphs are Hamiltonian, *Networks* **31** (1998), 29–38. [to check]

(5) V. Chvátal, Tough graphs and Hamiltonian circuits, *Discrete Mathematics* **5** (1973), 215–228. [to check]

(6) D. R. Fulkerson, O. A. Gross, Incidence matrices and interval graphs, *Pacific Journal of Mathematics* **15** (1965), 835–855. [to check]

(7) F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *Journal of Combinatorial Theory, Series B* **16** (1974), 47–56. [to check]

(8) X. Huang, A necessary and sufficient Hall condition for hypergraphs, arXiv:2608.10193v2 (2026).

(9) A. Kabela, T. Kaiser, 10-tough chordal graphs are Hamiltonian, *Journal of Combinatorial Theory, Series B* **122** (2017), 417–427; arXiv:1508.02568v3.

(10) T. Kaiser, D. Král', L. Stacho, Tough spiders, *Journal of Graph Theory* **56** (2007), 23–40. [to check]

(11) J. M. Keil, Finding Hamiltonian circuits in interval graphs, *Information Processing Letters* **20** (1985), 201–206. [to check]

(12) T. Kemp, Tough nonhamiltonian graphs (code), Zenodo (2020), record 3974274. [to check]

(13) D. Kratsch, J. Lehel, H. Müller, Toughness, Hamiltonicity and split graphs, *Discrete Mathematics* **150** (1996), 231–245. [to check]

(14) I.-J. Lin, T. A. McKee, D. B. West, The leafage of a chordal graph, *Discussiones Mathematicae Graph Theory* **18** (1998), 23–48. [to check]

(15) B. D. McKay, A. Piperno, Practical graph isomorphism, II, *Journal of Symbolic Computation* **60** (2014), 94–112. [to check]

(16) OEIS Foundation, *The On-Line Encyclopedia of Integer Sequences*, entries A001349 and A003216.

(17) E. R. Scheinerman, D. H. Ullman, *Fractional Graph Theory*, Wiley, New York, 1997. [to check]

(18) Z. Wang, Toughness bounds for fractional Hamiltonicity and resistance positivity, arXiv:2609.03412v1 (2026).

(19) L. Zhang, Is there a chordal graph whose toughness is exactly $\frac{3}{2}$ and which is non-Hamiltonian?, MathOverflow question 504792 (2025).