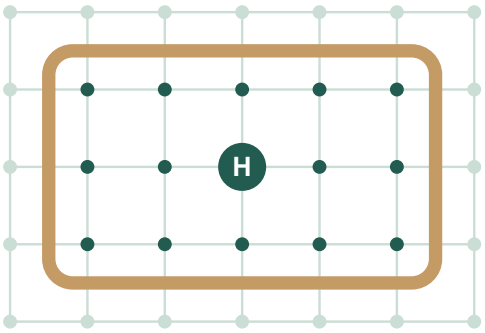
MATHEMATICAL ESSAY AND PROOFS

Enclosing a horse in polynomial time

A puzzle, an AI-assisted proof, and a generalization to plane graphs



Square grids: polynomial-time solvable

Arbitrary plane graphs: polynomial-time solvable

Narrative first. Detailed mathematical arguments follow.

Contents

The story 3

Part I. The square-grid proof 6

1. Graph model and preprocessing 6
2. Important walls and radial nooses 6
3. One maximal noose is compatible with the prescribed tree 8
4. Independent local ears give the exact score 8
5. The canonical clockwise ribbon and strict blocks 11
6. Endpoint normalization and exact-budget reductions 13
7. Ranked resources and elimination of mixed conflicts 14
8. Elimination of pure borrow conflicts 17
9. The complete algorithm and a polynomial bound 20

Part II. Arbitrary plane graphs 23

1. Preserve the objective while simplifying neutral regions 23
2. Represent an optimum by a forward radial walk 24
3. Compute the exact score with fixed local values 25
4. Optimize the global resource constraints in polynomial time 25
5. Size, bit complexity, and recovery 26

Appendix A. Planar resource termination 27

Appendix B. Verification and reading notes 34

The story

The question I started with

I asked an AI prover to establish that a stripped-down version of enclose.horse was NP-complete. The game suggests a natural optimization problem: place a limited number of walls, keep a horse from escaping, and leave it as much reachable grass as possible. Finding the right places to build walls felt like the sort of combinatorial search that might resist a polynomial-time algorithm.

The eventual result went the other way. For an explicitly listed square board containing only water, grass, and one horse, the problem is in P. The investigation then produced a polynomial-time algorithm for arbitrary finite plane graphs, with no bound on vertex degree, when escape occurs through the vertices of a designated exterior face.

Here, P means that one algorithm solves every instance in time bounded by a fixed polynomial in the input length. The algorithms developed in this investigation have very large polynomial upper bounds. The result settles the complexity classification; a fast puzzle solver would require further work.

The account below paraphrases the research record. It reports the mathematical work and its verification without reproducing the prover's transcript or identifying the model used.

The rules that determine the theorem

The board is an explicit square array. The horse moves between cells that share a side. Water is impassable; a new wall may replace any grass cell, including a boundary cell, at cost one. The horse's own cell cannot be walled. If its reachable component contains a cell on the outermost row or column, it escapes and the score is zero. Otherwise the score is the number of cells in that component, including the horse.

Given a budget B and a target K, the decision question is whether **at most B walls** can achieve score at least K. Both numbers may be written in binary. There are no extra horses, portals, bonuses, or other special objects in this model.

The words "at most" matter. If the horse is already enclosed, an unused wall need not be placed on grass that the horse could otherwise reach. In the graph extension, the escape vertices are fixed by the original embedding; deleting a vertex does not turn a newly exposed face into a new escape boundary.

From a hardness search to a geometric algorithm

The first pass gave a precise formulation and an efficient way to check proposed solutions. Let S be the horse's surviving component and let N(S) be its external vertex neighborhood: every traversable cell outside S that shares an edge with a cell in S. To make S exactly the horse's component, it is sufficient to wall N(S). Conversely, any wall placement producing S must wall that whole neighborhood.

For a positive target, the task is therefore to find a connected set S containing the horse, avoiding the boundary, with at least K vertices and at most B external neighbors. This reformulation fixes both what is being bought and what is being counted.

Several proposed hardness reductions ran into the planar geometry. A construction that works on a general graph can require escape paths that cross one another when drawn on a board. Large rooms attached to corridors also create extra grass that a purported reduction must account for. No valid hardness proof emerged from those approaches.

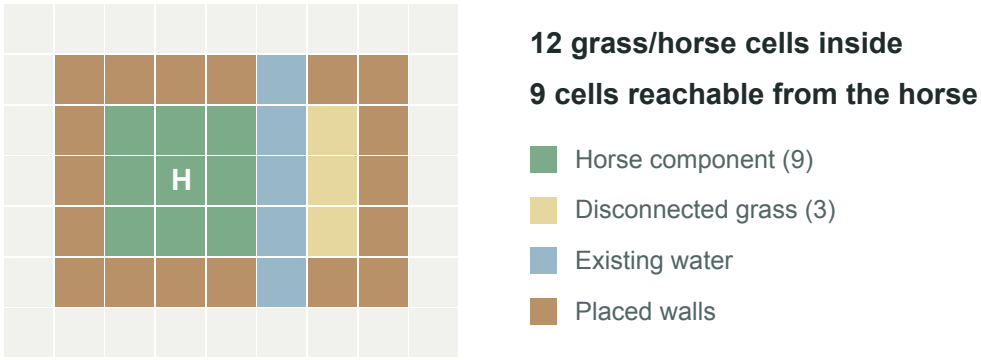
The algorithmic route had its own failures. A closed curve can surround grass that the horse cannot reach. Counting all grass inside a curve can therefore certify a score that no legal placement achieves. Other

candidate encodings allowed repeated visits to a junction or failed to represent the best enclosure. Small counterexamples were useful because they exposed the exact condition that an attractive general argument had omitted.

The successful square-grid proof combines three constructions. It represents a minimal wall separator by a curve through wall cells and face centers. It corrects the curve's score to exclude disconnected pieces. It then expresses the search as paths in an acyclic graph, with constraints preventing a face from being used incompatibly. The difficult step is proving that these particular path instances can always be simplified completely in polynomial time.

The transcript records distinct stages: a candidate mathematical argument, adversarial review, and a later formal development in Lean. Those stages are useful to keep separate. A successful finite test can expose bugs in an implementation, while a theorem about all board sizes needs a general proof.

Why counting the inside is insufficient



A geometric boundary can contain several disconnected grass components.

Imagine an enclosure containing two patches of grass separated by water and the chosen walls. The horse is in one patch. A geometric area calculation may count both patches; the game rewards only the patch containing the horse.

The square grid supplies an unusually strong restriction. Consider an optimal wall set from which no wall can be removed without allowing escape. At an ordinary wall cell, there must be at least two neighbors on the exterior side. Otherwise the wall can be moved to its sole exterior neighbor, retaining the old horse component and adding the restored wall cell at the same cost.

A square-grid cell has at most four neighbors. This exchange argument leaves at most one unused inward spoke at each selected wall in the radial representation. The proof organizes those spokes into local one-wall and two-wall corrections, called ears. Correcting the ears yields a boundary whose interior contains exactly the horse's component.

The reward can then be written as a sum of fixed values on boundary steps. A large bonus at the horse and a compensating negative charge at the exterior force a qualifying boundary to have the correct orientation. A face label acts as a resource: a valid path may use it only in the combinations allowed by the ear construction. This makes both the connectivity condition and the numerical objective available to the same finite search.

The extension needed a new argument

After the square-grid result, I asked whether the proof extended to arbitrary embedded planar graphs. It did not extend directly. A higher-degree wall can have several unused inward incidences, and one boundary vertex can connect to several interior vertices. Splitting such a vertex into smaller pieces can change what deleting one wall buys.

The investigation found a degree-five counterexample to the old independent-ear representation. The correct response was to replace that representation. The later proof first compresses regions that communicate with the rest of the graph through only one or two vertices. It preserves their total reward and proves that internal walls can be restored, replacing them by a pole wall when needed, without increasing cost or losing saved reward.

After this preprocessing, the algorithm tracks every face in a wall's inward fan. Cyclic ownership rules control how consecutive walls share those faces. Precomputed one-pole and two-pole component tables subtract the grass inside the boundary that remains disconnected from the horse. A ranked version of the resource search then handles arbitrarily large inward fans without enumerating arbitrary subsets of them.

The same obstruction drove both proofs: local boundary information must recover the actual connected component, and the search must enforce that information globally. The planar extension preserves this principle while replacing the degree-four shortcut.

What was checked

The final proof record contains executable decision procedures, Lean proofs of their equivalence to the original enclosure questions, uniform polynomial bounds on their counted primitive work, and bounds on the sizes of the integers used by the search. The formal statements quantify over the original inputs; they do not ask the caller to supply a successful noose or a termination certificate.

The recorded verification also includes finite comparisons against exhaustive solvers and source/generated-code audits intended to catch accidental recomputation of data that the work analysis assumes is shared. This document reports those recorded checks. It does not report a fresh rebuild of the formal project.

The machine-time interpretation combines the checked work and operand bounds with ordinary binary implementations of the stated primitives. The formal development does not verify the compiler or count every machine instruction. The exported formal interface is a decision procedure; the proofs also give polynomial-time optimization and wall-recovery arguments.

The remaining pages give the mathematical development. Part I retains the square-grid proof, including its geometric reduction argument. Part II gives the arbitrary-degree plane extension. Appendix A expands its ranked termination argument. Appendix B states the verification boundary in one place. Section and lemma numbers restart within each part.

Part I. The square-grid proof

Theorem. Let an instance be an explicit $n \times n$ array containing water, grass, and exactly one horse, together with nonnegative binary integers B,K. Walls have unit cost and may replace grass cells. Movement is by shared sides; reaching any traversable outermost-row or outermost-column cell is escape. An escaping placement has score zero, and otherwise its score is the number of cells reachable from the horse, including the horse cell. Deciding whether at most B walls can attain score at least K is in **P**. A maximum-score placement and its wall set can also be computed in polynomial time.

The theorem concerns this boundary convention, an upper-bound budget, unit positive cell prizes, and an explicit board. It includes no portals, bonus items, additional horses, or compressed geometric input. Write $m=n^2$ for the number of array cells.

1. Graph model and preprocessing

Let G be the induced grid graph on grass and horse cells, let h be the horse, and let T be its traversable boundary cells. For a wall set $W \subseteq V(G) \setminus \{h\}$, write $A(W)=C_h(G-W)$. A placement has positive score exactly when $A(W) \cap T=\varnothing$, in which case its score is $|A(W)|$.

For any connected set S containing h and avoiding T, walling its external **vertex** neighborhood $N_G(S)$ makes its horse component exactly S. Conversely, every successful wall set contains the external neighborhood of its horse component. Thus the positive-threshold problem is exactly

> find connected S containing h, disjoint from T,
> with $|S| \geq K$ and $|N_G(S)| \leq B$.

Handle the following cases directly. For $K=0$, answer yes. If the horse is on the boundary, no positive score is possible. If its initial component misses the boundary, the empty placement is optimal. A threshold larger than the number of traversable cells is impossible. Replace B by its minimum with the number of wallable cells; henceforth $B \leq m$.

In the remaining case put $Q=C_h(G-T)$. Every possible successful horse component is contained in Q. A nonboundary neighbor of such a component is also in Q; every other neighbor is a boundary cell adjacent to Q. Retain exactly Q and its adjacent boundary cells, and delete edges between boundary cells. This preserves every successful S and its entire external vertex neighborhood, including its cardinality.

Each retained boundary cell has exactly one neighbor in Q: a noncorner square-boundary cell has only one possible interior neighbor, and a corner has none. Add an undeletable vertex t in the exterior face and join it to all retained boundary cells through that face. Call the resulting connected simple plane graph Γ. The retained boundary cells are now degree-two **exit vertices**. Every other nonterminal has degree at most four. Only exit vertices are adjacent to t. The terminals h,t are nonadjacent; all other vertices are wallable. Separation of h,t in Γ is exactly successful enclosure on the original board.

Enumerate every wall set of size at most $\min(B,2)$ and evaluate it by flood fill, in $O(m^3)$ time. This handles the empty placement and the small multigraph nooses that will otherwise need exceptional root conventions. For completeness of the remaining search, it therefore suffices to represent important solutions having at least three walls. Soundness continues to hold for any smaller solutions found by that search.

2. Important walls and radial nooses

A separator is *minimal* if no proper subset separates h,t. It is *important* if it is minimal and no separator of no greater size has a strictly larger horse component.

Whenever a positive enclosure is feasible, choose a globally maximum-score separator within the budget and restore redundant walls until it is minimal. Its score cannot decrease. Since all cell prizes are positive, a separator dominating it would improve the global optimum. Therefore an important optimum exists.

Let W be important, and let A,Z be the components of h,t in Γ-W. Every wall has a neighbor in both A and Z: restoring that wall alone must create an h-t path. An ordinary wall has at least two neighbors in Z. Otherwise its unique such neighbor z is wallable, since the wall is not adjacent to t. Replacing that wall by z preserves separation and all of A, but restores the wall next to A, strictly increasing its score at the same cost. Indeed, any newly possible simple h-t path would have to pass through the restored wall and leave it through its unique old Z neighbor, which has just been removed. This proves the claimed exchange.

Construct the radial **multigraph** of Γ: it has a vertex for every primal vertex, a vertex for every primal face, and one spoke for every corner incidence. Repeated incidences are distinct parallel spokes. Delete h,t and their spokes from the graph in which cycles are sought; call that graph R. Their original primal locations remain marked points in faces of the drawing. Put the face containing t at infinity. A *noose* is a simple radial cycle; its primal vertices are its walls. Parallel-spoke bigons are allowed. Every noose has

```
length(C) = 2 |W(C)|.
```

The graph has O(m) vertices, spokes, and faces. It is connected after this preprocessing: passages through a deleted terminal in the full radial graph can be replaced around that terminal through the other, wallable endpoints of its incident primal edges. Alternatively, every construction below can be performed in the root component of R.

Lemma 2.1 (minimal-separator noose). Every minimal separator W is the wall set of a simple noose separating h,t.

Proof. Activate every radial spoke incident with a wall in W. In the full radial subdivision, complementary regions containing surviving primal vertices correspond exactly to components of Γ-W: omitting an unwalled vertex's star joins its incident primal-edge regions, just as traversing that vertex joins primal paths. Edges with two walled ends may give additional zero-vertex regions. Consequently the activated graph separates the locations of h,t. A plane graph separating two points contains a simple separating cycle, obtained by decomposing a relevant facial boundary into simple cycles and discarding bridge excursions. Its wall set is a subset of W; minimality makes that subset all of W. Conversely, a primal path avoiding those walls cannot cross the cycle, because spoke interiors avoid the primal graph. The entire component A is on the horse side. ∎

Orient a noose clockwise, with its bounded interior on the right. At a wall of primal degree d, let its *inward interval* be the clockwise incidences strictly between its outgoing spoke and its incoming spoke. These are locally unused inward spokes; a spoke is counted even when its face endpoint is elsewhere on the noose.

Lemma 2.2 (strong wall gate). Every noose of an important separator uses at most d-3 inward spokes at an ordinary wall of degree d, and none at an exit wall.

Proof. All A edge sectors are inside and all Z sectors outside. At an ordinary wall at least two sectors are outside. There are therefore at most d-2 inside sectors, and one fewer unused spoke between them. At an exit wall the two sectors are one horse sector and one exterior sector. ∎

In particular, degree below three is forbidden for an ordinary selected wall, degree three permits no inward spoke, and degree four permits one. The search will impose this gate on every candidate; it need not impose minimality itself.

3. One maximal noose is compatible with the prescribed tree

A raw geometric disk can contain grass disconnected from the horse. The following lemma is only a search-completeness statement, and does not identify its disk area with the score.

Lemma 3.1 (uniform shortest-tree compatibility). Fix a globally optimal minimal wall set W. There is a noose C with wall set W such that every shortest path from a root on C to a vertex of C lies in the closed disk of C. In particular, the same C is compatible with every BFS tree at every root on it.

Proof. Among all such nooses choose one enclosing an inclusion-maximal set of radial faces. Suppose a shortest root path has an exterior excursion P between consecutive contacts x,y with C. Its interior is disjoint from C. The exterior chord and the appropriate old arc form a simple outer noose C' enclosing the old disk and an additional radial face. The discarded arc is a path in the same graph R, so shortestness gives $|C'|\le|C|$, and hence $|W(C')|\le|W|$.

New walls lie outside the old disk, so they do not delete any old horse vertex. The entire old component remains connected to h. If an old wall were omitted, its old horse-side neighbor would make that restored positive-prize vertex reachable too, improving the optimum. Thus all old walls remain. The cost bound then forces W(C')=W, contradicting the maximal radial-face set. The argument applies to every root and every shortest path, proving the uniform statement. Both terminals are omitted from R throughout this comparison. ∎

No enumeration of shortest-path tie choices will be needed. An inward replacement used to correct the score later need not preserve this compatibility; the search continues to represent the maximal noose.

4. Independent local ears give the exact score

First fix a minimal separator W and any separating noose C with wall set exactly W. A literal radial chord gives an exact equality. Suppose P is a simple path between two distinct vertices of C, its interior misses C, its spokes are unused by C, and every wall of P belongs to W. Joining P to either old arc gives two simple cycles. Their edge supports have symmetric difference E(C), so exactly one of the two cycles separates h,t. Its walls form a separating subset of W, which minimality makes all of W. Both that cycle and C therefore have length 2|W|. Consequently the old arc discarded in this replacement has length exactly |P|.

Apply this observation to the shortest possible one- and two-spoke chords; bipartiteness certifies their shortestness. It proves both cofacial facts with their actual noose incidences:

- If a face f is on C and a selected wall u is incident with f, a used u-f spoke already makes them neighbors. Otherwise that spoke is a one-edge chord, and the equality just proved supplies a one-edge u-f arc in C. Every selected wall incident with f is therefore one of its two actual cycle neighbors. There are at most two such walls.
- If f is not on C and touches distinct selected walls u,v, the path u-f-v is a two-edge chord with no intermediate noose contact. The equality supplies a two-edge walk u-g-v supported by the **original** noose. Thus u,v are consecutive on that noose. It also supplies a separating replacement through f, with exactly the same wall set W. Applying the visited-face conclusion to this replacement shows that f touches at most two selected walls.

These statements retain parallel spokes as distinct incidences.

For an oriented candidate, call its right-hand complementary region its designated interior. This is a disk on the sphere, but it need not yet be the side avoiding the physical exterior mark t. A face label is **visited** if its center lies on the noose, and is **borrowed** once for each unused inward spoke ending there. Impose the following *independent-ear rule*:

- At a visited face, an additional inward spoke may come only from either actual neighboring wall. It is parallel to that wall's used spoke and is a singleton correction. Both neighbors may contribute.
- An unvisited face may receive one inward spoke, or two from distinct consecutive walls. Two such spokes give a pair correction. No other repeated use is permitted.

The strong gate gives each wall at most one unused inward spoke. A changed singleton therefore gives an actual two-edge cycle, consisting of its old and new parallel spokes. A pair gives the actual four-edge cycle u-g-v-f-u: u-g-v is the old short arc and u-f-v is the inward alternative. The poles u,v and the faces f,g are distinct. An unchanged singleton contributes no local cycle. The gate and the cofacial facts prove this rule for the important noose of Lemma 3.1; the search imposes it on every candidate, including nonminimal ones.

Lemma 4.1 (sound ear correction). The simultaneous local replacements give a simple central noose C_* with exactly the same walls. If h is in its designated interior and t is outside, the nonwall cells on that side are exactly the horse component of Γ-W.

Proof. Represent each singleton or wall-face-wall pair by its literal corrected word. Distinct pieces use disjoint wall sets. A singleton keeps its face endpoint; a pair replaces its middle face visit by its borrowed face. The independent-ear rule makes all these new visits distinct and prevents a borrowed face from appearing elsewhere in the corrected word. Two singleton changes may share an old face endpoint while using its two different incident spokes. Thus the corrected cyclic word still visits each wall and each face at most once, with distinct incoming and outgoing spokes. It is the claimed simple cycle; no intermediate replacement order is needed.

Now activate every spoke at every wall in W. On the designated interior side of C_*, each extra activated spoke is a pendant slit whose face endpoint is unvisited and distinct from every other such endpoint. At an edited wall, the discarded old spoke is on the outer side of the corrected turn. At an unedited wall, the only possible extra inward spoke is its unmatched borrow. Repeated inward labels have been incorporated into the cycle, and the remaining labels occur once. Removing these slits does not disconnect the open disk. The activated-star/component correspondence of Lemma 2.1 therefore identifies every surviving primal vertex on that side with the horse component. Since t is outside, this component consists of board cells. ∎

For the minimal wall set, horse retention follows from the supports of the actual local cycles, without assuming disjoint lens interiors. Let E_1,…,E_s be the nonempty two- or four-edge cycles just constructed. At each piece its old and new spoke sets differ exactly by E_i; common spokes cancel. Different pieces have disjoint wall footprints, so their edge identities combine to give

$$E(C) \triangle E(C_*) = E(E_1) \triangle \dots \triangle E(E_s).$$

For a cycle D, let χ(D,x) be one when D separates x from t and zero otherwise. A dual walk from t to x changes sides exactly when it crosses a selected spoke. The support identity consequently gives, modulo two,

$$\chi(C,x) + \chi(C_*,x) = \chi(E_1,x) + \dots + \chi(E_s,x).$$

Each E_i uses at most two walls from W. If it separated h,t, those poles would themselves separate the primal terminals, contrary to minimality when |W|≥3. Hence every χ(E_i,h)=0, and the old and new cycles separate h,t identically. Moreover every newly added spoke lies on the old inward side. An old exterior path to t meets none of them and remains an exterior path for C_*. Thus the physical exterior stays outside and the horse stays inside. Lemma 4.1 now gives exactly the original horse component. This proves horse retention by support parity.

4.2 Computing the value from local turns

Write $g=|V(G)|\le m$ and set $M=g+1$. Give each retained board cell one unit of prize, including the horse, and give h an additional bonus M. Choose one actual incident-sector anchor a_v for each retained cell v, with all anchors off the cycle spokes. For a wall represented in the chosen branch use its parent-spoke anchor; other cells use an incident primal location found by finite search. A nonwall cell's sectors all lie on the same side of the noose, so any such anchor counts it correctly. A selected wall has unit mass, which is removed exactly when its anchor is inside.

Let $p_v=1+M\cdot 1_{v=h}$ and let $P=\Sigma_v p_v$ over retained cells. There is one row per retained cell, with no duplicate cell rows, and a separate exterior row of mass $-P$ at a mark belonging to t. For the designated interior D_* of the corrected candidate define

$$F(C) = \Sigma_v p_v \cdot 1_{a_v \in D_*} - P \cdot 1_{t \in D_*} - \Sigma_{w \in W} 1_{a_w \in D_*}.$$

If $t\in D_*$, the positive rows sum to at most P, so $F(C)\le 0$. If the horse is outside, its bonus is absent and $F(C)\le g<M$. If the horse is inside and t is outside, Lemma 4.1 and the wall-anchor subtraction give $F(C)=M+|A(W)|$, with $A(W)$ the original board's horse component. Consequently

$$F(C) \ge M+K \iff h\in D_*,\ t\notin D_*,\ \text{and } |A(W)|\ge K.$$

A designated interior containing the physical exterior cannot pass, so the orientation need not be guessed by the path search. For the important optimum, the retention argument establishes the required horse and exterior sides, so its value is exactly $M+|A(W)|$.

Construct a signed antisymmetric spoke cochain ω by routing each positive row from the exterior mark to its anchor along a finite walk in the full radial dual. A crossing contributes its signed row mass; steps within a dual face contribute zero. Telescoping along the walks gives the finite Stokes identity

$$\Sigma_{\text{directed spokes of } C_*} \omega = \Sigma_v p_v \cdot 1_{a_v \in D_*} - P \cdot 1_{t \in D_*}.$$

This uses actual full-board anchors even when the searched cycle lies in just one radial component. The cochain is prepared once per root/corner choice, before its candidate branches are evaluated, and is independent of their cycles.

Store $\omega(e)$ in the direction from wall to face. At a corrected wall turn with incoming spoke e_{in} and outgoing spoke e_{out}, its local value is

$$\omega(e_{out}) - \omega(e_{in}) - 1_{\{\text{the wall's parent anchor is on the inward side}\}}.$$

The last term is a finite lookup in the wall's cyclic incidence order. Sum these terms over the corrected wall turns of each singleton or pair piece; face-only pieces have value zero. Every cycle spoke has exactly one wall endpoint, so these piece values sum to $F(C)$. Singleton and pair corrections are therefore computed directly from their two or four corrected spoke entries and their wall-anchor lookups. Equivalently, their local old-to-new circulations can be subtracted from an outgoing-event sum; that convention requires one separate root-spoke boundary term. Section 6 uses the wall-local convention, in which the root spoke is already counted at its wall and requires no extra term.

The positive mass is $P \le 2g+1 = O(m)$. The implemented dual walks have $O(m)$ steps, giving the conservative bound $|\omega(e)| = O(m^2)$ and the same bound for a piece value. The resulting integers use $O(\log m)$ bits. Piece values and partial path values may be negative and must be retained by the exact-cost path algorithm.

5. The canonical clockwise ribbon and strict blocks

For each radial face root r and each corner at r, compute ordinary BFS distances d. Every spoke joins consecutive levels, by bipartiteness. Retain only its increasing-level direction and run recursive DFS in this directed graph. At a nonroot vertex scan incidences counterclockwise starting immediately before its parent incidence; at the root start immediately before the first clockwise incidence of the selected corner. The resulting tree T_r is a BFS tree. The graph of increasing-level spokes reaches every vertex, so this DFS spans the graph.

Cut along the tree and list its ports in clockwise contour order, starting at the chosen root corner. A tree spoke has the two forward real arcs corresponding to its outward and return traversals. A non-tree spoke gives one real arc from its earlier port to its later port. Forward zero runs skip between occurrences of the same original vertex, possibly skipping whole child intervals. The contour has $O(m)$ ports. Let $I_v = [L_v, U_v]$ be the hull of the subtree rooted at v.

Lemma 5.1 (aligned orientation). Every forward non-tree arc goes from level i to level $i+1$.

Proof. When DFS scans an unused increasing-level spoke $u \to v$, v is already finished: it cannot be an ancestor because tree levels strictly increase. If it is outside u's subtree, its branch was explored earlier by DFS and is therefore later in the reversed, clockwise contour. If it is a descendant, the level difference makes it an immediate child reached by a different parallel tree spoke. That tree spoke was scanned first; reversing the local scan puts the non-tree port before the child's entire hull. In both cases the port at u precedes the port at v. ∎

A *turn event* is an incoming real arc and an outgoing real arc at the same original vertex with nondecreasing intervening port positions. Consecutive events share their intervening real arc. The root occurs only at the beginning and end. Reject equal original neighboring wall vertices at a face turn, equal original neighboring face vertices at a wall turn, and equal first and last walls at closure. Impose the strong wall gate from Lemma 2.2. These finite event graphs are acyclic because successive real arcs strictly advance their ports.

A face event emits a typed visit V_f. A wall event with an unused inward spoke emits a borrow B_f. First absorb a borrow parallel to an adjacent used spoke into that adjacent face visit, charging its singleton lens at the wall. There is a unique such adjacent visit, since equal neighboring face labels were rejected. Two singleton ears at the same face are both absorbed; its one visit still emits f once. Root singletons are permitted at the two endpoint walls and are charged there. Every other borrow of the root label is forbidden.

For the remaining nonabsorbed borrows, use single events and pair blocks

$$B_f,\ V_g,\ B_f \qquad (f \neq g),$$

where a pair block emits both f and g, and has the external real-arc interfaces of its first and last atoms. Pair the leftmost unmatched window greedily. A face-event copy records the previous wall's borrow label and whether that wall has already been matched. An equal next borrow is represented by the pair block when unmatched, and is rejected when already matched. Thus the raw pattern

$$B_f,\ V_g,\ B_f,\ V_q,\ B_f$$

has no strict encoding. Every resource-valid path is preserved: the removed pattern would require two distinct blocks emitting f. Copies retain their original atoms and labels, and a block contains at most one face visit. In particular, the exact local interface is:

An adjacent equal-borrow window across a face is one pair block, or it is absent because its first borrow was already matched.

Require emitted labels of distinct blocks to be disjoint. This is exactly the independent-ear condition: after singleton absorption, every allowed repeated borrow is one disjoint pair window. Keeping the middle visit label g is essential. It also makes face visits unique.

Lemma 5.2 (simplicity). A resource-valid strict path projects to a simple radial cycle.

Proof. If a wall repeats on a raw forward path, the strong gate and port order force exactly the two runs

$$P \to w \to F \dots F \to w \to P,$$

using its parent and one child tree spoke in opposite directions. Four distinct used incidences put the later two inward at the earlier turn; exactly three incidences violate the combined degree gates; the remaining two-incidence case is the displayed parent-child return. It forces two different visits to the child face F: one turn w-F-w was locally rejected. That contradicts distinct block emissions, since no block has two face visits. Parallel spokes are retained in this argument, and the distinct endpoint walls exclude the root exception. Thus both wall and face vertices are unique. ∎

For completeness, choose the root from the fixed maximal noose before choosing its BFS tree. Relate two distinct selected walls when their unused inward spokes end at the same face. The strong gate makes that spoke unique at each wall, and the cofacial bound gives it at most one other selected wall as a mate. This is an intrinsic matching, defined by the original corner incidences and the physical inside side of the noose. It does not change when the root or the BFS tree changes.

Take any three successive walls u,v,w of a noose with $k \geq 3$, and let its intervening face visits be u-f-v-q-w. Consider the two possible cuts at f and q. The pairs {u,v} and {v,w} cannot both belong to the matching, since $u \neq w$. Choose the face corresponding to an unmatched pair. The cofacial bound says that this face touches exactly its two displayed selected walls. Thus no two selected walls incident with the chosen root can share unused inward spokes to any face. In particular, a pair block cannot straddle this root cut. These are the two adjacent candidate cuts.

Choose an exterior corner at that root face. Lemma 3.1 makes the same noose compatible with the canonical BFS tree now selected for the corner. Cutting the union of its tree paths makes its clockwise boundary forward; reinserting the other tree branches only inserts intervals that a zero run can skip. The literal boundary therefore has a strict tiling into singleton and pair pieces. Its resources are distinct, no pair crosses the root, and any root-directed inward spoke is absorbed into the adjacent endpoint singleton. The intrinsic matching argument established this root property before any rerooting.

The explicit solver enumerates every allowed root incidence and every cyclic starting position at its root face. Each index has $O(m)$ possibilities, giving the conservative total $O(m^2)$ root/corner pairs. This enumeration includes the chosen geometric root and exterior cut.

6. Endpoint normalization and exact-budget reductions

Give a wall event cost one and a face event cost zero; a block sums its atomic costs. Use the wall-local piece value from §4.2, which sums the corrected outgoing-minus-incoming spoke values and subtracts the selected interior wall anchors. A face-only piece has value zero. The root spoke is already included at its wall endpoint. By simplicity, a valid word's total cost is its number of distinct walls, and its sum of piece values is $F(C)$.

Enumerate the actual first and last **blocks**, their two real root arcs, and the last block's automaton state. The first block starts in the initial state. Reject choices whose first and last wall vertices coincide. Exclude every other block using a resource emitted by either endpoint block, and restrict the graph to the source-to-sink interval. If distinct endpoint blocks share a resource, they cannot both be live. A single-block path is permitted; it has at most two walls and merely duplicates a case covered by the direct search. This normalization preserves every valid word in its corresponding branch. The selected root and the strict resource rules exclude a pair correction across the root cut.

Resource normalization does not remove numerical charges. Put the first block's cost and value in source constants (b_0,v_0). Every transition pays the full cost and value of its target block, including the last block. Compute an exact-cost profile for the remaining path, then shift it: its entry at total cost b is unreachable when $b<b_0$, and otherwise is v_0 plus the unshifted entry at $b-b_0$. Only total costs $0\le b\le B$ are examined. Thus the source is charged exactly once, including for a path with no transition. Computing the unshifted profile through B and discarding the excess after the shift, as the implementation does, is equivalent to allocating only $B-b_0$ entries for that branch.

Let F_0 be the fixed resource-conflict relation: two distinct live blocks conflict when their emitted sets intersect. Endpoint freshness ensures that no such pair involves a terminal. A block path is valid exactly when it avoids F_0.

For every current arc maintain an exact-cost table

```
L_uv[b] = maximum value of a represented u-v passage of cost b,
          or unreachable,                    0 ≤ b ≤ B.
```

Maintain original-path provenance for each retained entry. The rules are:

1. Bypass an interior vertex x with no reachable forbidden mate, using $L_{uv}[b] \leftarrow \max(L_{uv}[b], \max_{i+j=b}(L_{ux}[i]+L_{xv}[j]))$.
2. Delete an arc whose endpoints form a forbidden pair.
3. Discard a pair whose endpoints are no longer reachable, and trim vertices outside the σ-τ core.

Reachability uses the support graph of table entries. A current path may have total cost above B; this conservative reachability still supports every structural argument below. Nonnegative costs justify truncating individual table entries above B. No truncation by the sign of a value is valid or used.

6.1 Global provenance, not an arcwise compatibility assumption

Lemma 6.1. For every current directed path, and every choice of its retained table-entry witnesses, its expansion in the original normalized DAG contains no forbidden pair with an eliminated endpoint.

Proof. Induct on the reductions, with the assertion quantified over all current paths. Initially it is vacuous. Deleting arcs and trimming create no new expanded paths. Reachability between surviving vertices never increases: a new shortcut only replaces an old passage. Hence an incomparable discarded pair cannot later become possible.

When an unpaired vertex x is bypassed, every new current path lifts to an old current path through x. The inductive assertion, applied to that entire old path, excludes every conflict involving a previously eliminated block, including conflicts whose two occurrences lie in different shortcut passages. A new violation involving x and a still-retained mate would make them comparable before the bypass, contrary to x being unpaired. Acyclicity prevents repeated original vertices during concatenation. This proves the invariant. ∎

Consequently, if two distinct original blocks sharing a label occur on an expanded current path, both remain as current vertices. Labels prepaid at the endpoints are a separate initial case: every other occurrence was deleted before reductions, and the endpoint sets were disjoint.

The convolution loses no feasible exact-cost value: it checks every split, and alternatives are maximized only at equal cost. Replacing a passage by its maximizing witness cannot create a forgotten compatibility condition, by Lemma 6.1 for the whole surrounding current path. Visible forbidden endpoints remain to be handled by the rules. Thus the rules preserve all optimal feasible profiles. When only σ,τ remain, every retained entry expands to a valid original path, and its value is the exact optimum at its cost. It remains to prove that this always happens for the constructed radial instances.

7. Ranked resources and elimination of mixed conflicts

Rank nonroot face resources by decreasing depth, and at equal depth by earlier contour hull. Thus the priority is (d(f),-L_f), maximized lexicographically. Equal-depth hulls are disjoint.

Lemma 7.1 (atomic rank and anchor sandwich). A face resource consumed at an atomic anchor in I_f is either f or has higher priority. If comparable blocks A,B have selected anchors in I_f, every strict middle block has its entire atomic span in I_f.

Proof. A visited proper-descendant face has depth at least d(f)+2. For a borrow at a descendant wall x, adjacency gives d(g)≥d(f). Equality with g≠f forces x to be a child of f and the spoke g-x to be non-tree. Its increasing-level direction is forward, so I_g is before I_f; hence g has higher priority.

For a block, take its span from the target port of its first incoming arc to the source port of its last outgoing arc. Every atom is in this span. A transition, and therefore every shortcut, strictly separates consecutive spans. If α,β are the selected endpoint anchors and X is a strict middle block, then

$$\alpha \le \mathrm{hi}(A) < \mathrm{lo}(X) \le \mathrm{hi}(X) < \mathrm{lo}(B) \le \beta.$$

This proves the sandwich, including straddling endpoint blocks. ∎

Assume for contradiction that a fully reduced nontrivial core remains. Every interior vertex is paired; choose the highest active resource f, where *active* means that two retained occurrences are comparable. By global provenance, an inactive higher label cannot occur in two distinct blocks of an expanded path. In particular no higher face is visited twice, since each block has at most one face visit. Call this path property *higher-clean for f*.

Corollary 7.2 (highest anchored pair). Two comparable retained f occurrences cannot both have actual consumption anchors in I_f. Every strict middle block would emit only f or higher labels, so its pairedness forces an f emission. The first edge would then join two f occurrences, contrary to full reduction; a direct endpoint edge is also forbidden. Therefore no expanded current path visits f twice.

7.3 The cap lemmas

We record the precise geometric facts used below. All concern complete raw paths that are higher-clean for f; they do not assume those paths are resource-valid. A *non-tree neighbor* is a wall that is neither the parent nor a child of f, not merely a different parallel spoke.

1. **Lower-to-inside cap.** A non-tree-neighbor borrow before an actual f anchor in I_f is at a wall x of depth d(f)-1. Its turn is parent entry into an actual child face H, with d(H)=d(f) and I_H before I_f. All strict intermediate atoms lie in I_H ∪ I_f.
2. **Inside-to-higher cap.** An actual f anchor in I_f before a non-tree-neighbor borrow forces the wall y to have depth d(f)+1. Its turn exits from an actual child Z to its parent, with d(Z)=d(f)+2. All strict intermediate atoms lie in I_f ∪ I_Z.
3. **Early-parent cap.** A parent-wall borrow before an actual anchor in I_f has clockwise incidences P,h,H,f, using P→u→H and the actual tree spokes to H,f. Their consecutive child hulls satisfy I_H<I_f. Intermediate atoms lie in I_H ∪ I_f.

Here and below a hull union describes actual atoms, not an assignment of all labels of an endpoint macro to its middle visited face.

For clarity, the common barrier argument is as follows. After cutting the tree, non-tree spokes are noncrossing chords: their endpoint ports cannot strictly alternate. Entering or leaving a child hull uses its parent spoke or a non-tree chord. Gaps between an indicated child hull and the adjacent wall port contain only that wall's markers. A forward wall turn borrowing a nonparent spoke must use the wall's parent spoke; otherwise both the borrowed spoke and the parent incidence would lie inward. Parallel alternatives that leave the actual child spoke inward would violate the one-inward gate.

For the first cap let the unused chord have ports a<b, at x,f. A parent exit would trap a child immediately after a, forcing an immediate reversal or a repeated higher face. Thus the turn enters a child H immediately before a. A non-tree outgoing face would have its equal-depth hull after I_f, too late for the selected inside anchor. The exit from I_H cannot return to its parent, since that would repeat H. Its non-tree chord ends after b by noncrossing; since it still precedes the inside anchor, it ends in I_f.

For the second cap a parent entry similarly traps a deeper child before the unused wall port. The remaining parent exit cannot have a non-tree incoming face before I_f, since that would precede the selected inside anchor. It therefore arrives from the actual child Z after the wall port. Entry into I_Z is non-tree and must start before the unused port at f; chronology puts that source in I_f. The early parent case uses its consecutive child hulls; an outgoing non-tree face would be after the whole parent-wall hull, and a wall turn in the gap either violates the gate or exits past I_f.

These arguments prove the statements; the full literal-incidence checks are in the non-tree cap technical note [1], anchor extension [2], and early-parent proof [3]. Every extra cap has priority above f. Lemma 7.1 and pairedness therefore exclude these pairs from a fully reduced core. This covers all mixed visit/borrow conflicts except a visit followed by a late borrow at its parent wall.

7.4 The late-parent sibling configuration

For that remaining mixed case, let u=parent(f) and let P be its parent face. Port order and the strong gate force the literal order

```
P, f, Q, h
```

with actual tree children f,Q. The late borrow is Q→u→P, borrowing the actual tree spoke to f. A non-tree incoming spoke would originate at a lower face before I_u, too early to follow the selected visit to f. The child hulls I_f,I_Q are consecutive, in this order.

Only f,Q can be active labels at atoms in I_Q: other labels are deeper or in an earlier equal-depth hull, and there is no hull between I_f,I_Q. Also u cannot borrow Q nonabsorbed. Its Q incidence is opposite its parent incidence, and the only one-inward passage through the other two spokes goes backward. Parallel uses next to a Q visit are singleton-absorbed.

Sibling exit fact. On a higher-clean path, a borrow of f at a wall x in I_Q is immediately followed by V_Q.

Depth and adjacency first make x a child of Q; its turn must use its parent spoke. Suppose it entered x from Q instead of exiting to Q. Its rotation is Q,A,Z,f. The four-spoke cycle u-f-x-Q-u bounds a disk whose wedge at u is between consecutive f,Q spokes and has no further incidence. The root lies outside that disk by tree depth. The outgoing xZ enters it. It cannot subsequently leave through u; all ports at f and all disk-side ports at Q are behind the entry. Thus it must return to x. The repeated-wall classification forces two visits of a child face at depth d(f)+2, contradicting higher-cleanness. Therefore it uses x→Q. This argument holds for every continuation, including a mate on another branch. See the full disk argument [3].

Lemma 7.3 (inside-sibling exclusion). No retained f occurrence anchored in I_Q reaches the late-parent block B.

Proof. If there is one, choose the last retained f occurrence A before B on such a path. All strict middles lie in I_Q, so every paired middle emits f or Q. None emits f by the choice of A. A direct A-B edge is forbidden; two middle vertices give a forbidden Q-Q edge. Hence the current path is A→X→B, with one Q-emitting middle block, and neither endpoint emits Q.

The sibling exit fact gives an actual Q visit immediately after the borrow in A; the late-parent turn has an actual Q visit immediately before it. Neither can have been bypassed separately from the retained Q occurrence in X, by global provenance. Since a block has at most one face visit, these are the same actual visit in X. The borrowing atoms are last in A and first in B, as the endpoints do not emit Q. Thus the raw window is B_f,V_Q,B_f across distinct blocks. The strict interface groups it or rejects it; it cannot remain split. This contradiction proves the lemma. ∎

7.5 The full sibling-mate argument

The following argument will be used both now and for pure borrows. Suppose an f pair A,B with late-parent endpoint B has been reduced to A→X→B, where X contains an actual V_Q, every strict middle f emission has been excluded, and neither endpoint emits Q. Assume additionally that an expanded path

```
A ... V_Q ... E
```

with an f anchor of E in I_Q is impossible. The cap lemmas will verify this assumption for each use of the argument.

Since X is an interior retained vertex, it has a comparable mate Y emitting Q. The mate path need not retain both A and B. Use exactly the following available concatenations:

- If Y→X, append X→B. Any intervening f occurrence in I_Q would reach B, contrary to Lemma 7.3.
- If X→Y, prepend A→X. An intervening f occurrence in I_Q contradicts the displayed additional assumption.

If Y has a Q anchor in I_Q, the anchor sandwich puts every strict middle block of the X-Y path in I_Q. Its only possible active labels are f,Q; the relevant concatenation excludes f. Every paired middle therefore emits Q, exposing a forbidden edge.

Otherwise Y is a non-tree-neighbor borrow of Q: visits and child borrows have anchors in I_Q, and u cannot borrow Q. Apply the mixed non-tree cap to the actual visit in X. It is available for Q, because all faces above f have no repeated visit by provenance, and f itself has no repeated visit by Corollary 7.2. No priority lies strictly between the equal-depth consecutive faces f,Q.

For an earlier Y, the extra cap has depth d(Q) and is before I_Q. It cannot be f, because its parent is the borrowing wall of Y, distinct from u=parent(f). It is therefore before I_f and outranks f. For a later Y, the extra cap has depth d(Q)+2, also above f. Every label in these extra caps is inactive. Within I_Q, the appropriate

concatenation again excludes f. All paired strict middles on the mate path therefore emit Q; its first edge is forbidden, including when the mate path is a single edge.

Thus no mate Y exists, contradicting the pairedness of X. This proves the sibling-mate argument without an inference about incomparable occurrences on other branches.

Apply it first to a block A visiting f and the late-parent block B. All strict middle atoms lie in $I_f \cup I_Q$; a further turn at u in their gap either violates the wall gate or uses a parent-child return whose first run is before I_f. A middle f anchor in I_f is excluded by Corollary 7.2, and one in I_Q by Lemma 7.3. Only Q can remain active in the middle, forcing $A \to X \to B$ as above. The actual visit to Q before B must be in X, by provenance. Finally, an f borrow in I_Q is at a child wall of Q, a higher non-tree neighbor of f. The inside-to-higher cap from the visit in A excludes the intervening V_Q. All hypotheses of the mate argument hold, completing elimination of this mixed case.

We have now excluded every mixed conflict for the highest active f. A current path through an f conflict cannot visit f anywhere in its atomic expansion. If that visit's block had been eliminated, Lemma 6.1 would forbid its coexistence with a retained f endpoint. If it remains, it supplies a mixed conflict or an anchored visit pair already excluded. Thus every remaining f conflict has a wholly pure borrow witness. This is a consequence of global provenance, not a pathwise assumption about the selected endpoints.

8. Elimination of pure borrow conflicts

8.1 One cycle and a fixed incidence order

The repeated-wall classification gives a useful restriction on a higher-clean pure-borrow witness. If a repeated wall x borrows f and $d(x)=d(f)+1$, its repeated child face is deeper than f, which is impossible. If $d(x)=d(f)-1$ and $x \neq \text{parent}(f)$, the unused non-tree spoke points forward from x to f. Thus $I_x < I_f$; the repeated child face has depth d(f) and an earlier hull, again impossible. The only remaining borrow at a repeated wall is the actual tree spoke at parent(f), on its second return from a later equal-depth child. There is at most one such borrow. A parallel alternative either puts the tree spoke inward too or forces the earlier, higher-priority child-return case. These are the literal-incidence cases in the repeated-borrow lemma [4].

Contract every used wall of a raw path to an edge between its two used face neighbors. The resulting connected plane multigraph is a cactus. Indeed, forward face runs take consecutive used darts; a skipped dart cannot be used later without reversing contour order. The full walk is therefore the boundary walk of one common left face, incident with every edge. A connected plane graph with all edges incident to one face has only cycle blocks and bridges. A wall used once belongs to one cycle; a wall used twice is a bridge.

A borrowed spoke at a bridge enters the common left face. A borrow at a cycle wall enters that cycle's opposite face. The latter faces are distinct for distinct cycle blocks. Since f is not visited, its point is off the used cactus. If it receives at least two borrows, it cannot lie in the common left face, where all borrowers would be repeated walls and there is at most one. Thus all its borrowing walls lie once each on a single cycle block.

Lemma 8.1 (fixed order). On such a pure witness, borrowed incidences occur in clockwise order at f, cut just after its tree-parent spoke. A borrow of that parent spoke is first when its wall uses parent entry, and last when its wall uses parent exit.

Proof. Let F be the base face at which the full facial walk enters and leaves the relevant cycle block, using the root if it is the root block. At every other cycle vertex an unused parent incidence lies on the cycle's right, by forward port order. The tree path from the root to f can consequently first meet this cycle only at F; after entering its opposite-face disk it cannot leave and reenter at another cycle vertex. Its segment from F to f stays in that disk.

It cannot meet a selected borrowed spoke except at f or its parent wall. Any such meeting elsewhere would be at an ancestor wall adjacent to f, and BFS distance makes that ancestor its immediate parent. If that wall is used, the actual tree spoke is the borrowed incidence; a parallel alternative would put both spokes inward. Cutting the disk along this tree segment cuts the embedded star of borrowed spokes in the parent-spoke sector. Its boundary order is the stated clockwise linear order. Parent entry and exit put this sector before or after the parent-wall occurrence, respectively. ∎

Classify a borrow anchor as follows:

Type	Location
L	Non-tree-neighbor wall at depth d(f)-1, before l_f
C	Child wall of f, anchored inside l_f
R	Non-tree-neighbor wall at depth d(f)+1, after l_f
E	Early borrow at parent(f)
S	Late borrow at parent(f)

Two L borrows would have their wall ports and their f ports in the same order, with both wall ports before l_f. Their non-tree chords would strictly alternate. Two R borrows give the same contradiction after l_f. Thus there is at most one of each. E is first and S last. They cannot coexist, since their shared borrowing wall is used once. E and L cannot coexist either: the latter's equal-depth wall hull is before the parent-wall hull, contrary to E being first. The sequence is therefore an optional E or L, followed by child borrows, an optional R, and an optional S, with those incompatibilities.

8.2 A non-tree later borrow has only higher extra caps

Consider distinct blocks with a later R borrow at y. It must be parent exit: parent entry would trap a deeper child, as in the upper cap argument. Its incoming spoke is either an actual child Z, of depth d(f)+2, or a non-tree spoke from a face G whose equal-depth hull is before l_f.

The latter exception is too early for a preceding C anchor. For a preceding L at x, write the unused borrowed chords as a-b and c-d, where a is at x, b,c are at f, and d is at y. Lemma 8.1 gives

$$a < b < c < d.$$

The incoming G-y chord ends after d; its source precedes l_f. Noncrossing with a-b puts that source before a. Yet its visit follows the earlier borrow. Depth and contour position force G to be the earlier wall's actual used child H, immediately before a. A non-tree outgoing face after l_f could not reach it. For an earlier E, the corresponding child H is immediately before l_f, and again G=H. Higher-cleanness makes the visits immediately after the first wall and immediately before the last wall coincide. The raw window is exactly B_f,V_H,B_f, which the strict interface puts in one block or rejects. It cannot supply the two distinct blocks.

The later incoming spoke is therefore the actual child spoke from Z, with l_Z immediately after d. Its entry chord must start before c; returning along its parent edge would repeat Z, while a source between c,d would cross the unused spoke.

The earlier L or E must also enter an actual higher-priority child cap H. Here the absence of an actual visit to f matters. If the outgoing spoke went to a non-tree face G, its equal-depth hull would be after l_f. The later unused chord c-d forces its target past d; but a depth-d(f) face port cannot lie inside the deeper wall hull l_y. Whether G is the parent of y or is unrelated, that target is after all of l_y, too late for the selected borrow. Parallel alternatives violate the inward gate. Thus the child cap is actual, and has priority above f.

For L, the exit chord from I_H ends after b; the entry chord of I_Z starts before c. Either one chord goes directly from I_H to I_Z, or their intervening endpoints and all intervening atoms are inside I_f. For E, the same argument uses the absence of another vertex between its cap I_H and I_f. For C, the first anchor is already in I_f. Hence the strict intermediate atoms lie in

L-R or E-R: $I_H \cup I_f \cup I_Z$, C-R: $I_f \cup I_Z$.

All extra caps are higher than f. Every paired middle must therefore emit f, exposing a forbidden edge. This excludes all three cases. The full two-spoke proof [5] includes the literal parallel-spoke and distinct-block checks.

8.3 The pure late-parent cases

Let the later borrow be S at u=parent(f). It uses the actual tree spoke to f, and has local order P,f,Q,h with parent exit. We must first prove that its incoming Q incidence is a tree child; the earlier mixed proof had an actual f visit that is no longer assumed.

Suppose instead the incoming spoke is non-tree from a face G of depth d(f)-2, before I_u. An earlier C or R is too late to reach that visit. An earlier E repeats the same borrowing wall and is impossible. Thus the earlier borrow is L at a wall x. Its unused chord is a-b, with a at x and b at f. The proposed G-u chord ends after I_f; noncrossing puts its source $g<a$. Since d(G)=d(x)-1, G is either the parent of x or has a hull disjoint from I_x. All its ports are outside I_x, so $g<a \in I_x$ places it before the entire I_x. It cannot follow the earlier borrow. This contradiction makes Q an actual child, consecutive after f.

An earlier R before this S lies in I_Q, and Lemma 7.3 excludes it. For an earlier C, all strict middle atoms lie in $I_f \cup I_Q$. For an earlier L, parent exit is still impossible by the lower barrier. Its parent-entry outgoing spoke has two possibilities:

- An actual child H before I_f, giving intermediate atoms in $I_H \cup I_f \cup I_Q$. Its exit chord ends after the unused port at f, and must still precede the final visit to Q.
- A non-tree outgoing face after I_f. By equal depth and the need to reach the final Q visit, that face must be Q itself. The prefix is the direct raw step B_f,V_Q, and every following middle atom is in I_Q.

There is no intervening extra turn at u: pure localization makes its borrowing occurrence unique. The child hulls are consecutive, so there are no other intervening contour regions. The direct-Q case is real and cannot be omitted by assuming a child cap at every earlier wall.

For a direct-Q prefix, Lemma 7.3 excludes every surviving middle f. Only Q can remain active. A direct endpoint edge is forbidden, and more than one middle vertex gives a forbidden Q-Q edge. Hence the path is A→X→B, with one Q block and endpoints not emitting Q. The visits immediately after A and before B cannot have been bypassed separately from X. They coincide in its single actual face visit, forcing B_f,V_Q,B_f across distinct blocks, contrary to the strict interface.

For the higher-cap L prefix and the C prefix, a middle f anchor in I_f is excluded by the lower-to-inside cap or Corollary 7.2, respectively. A middle anchor in I_Q is excluded by Lemma 7.3; one in I_H cannot emit the lower-priority label f. Only Q remains active in the middle. Thus again A→X→B, and provenance places the actual Q visit before B in X.

Apply the full sibling-mate argument of Section 7.5. Its additional assumption holds: an f anchor in I_Q after V_Q is an R borrow relative to f. Prepending A gives the L-R cap just proved, or the C-R inside-anchor cap. Neither can contain the actual visit to the distinct equal-depth later sibling Q. The earlier mate direction uses X→B and Lemma 7.3; the later direction uses A→X and this cap. For an outside Q mate, its extra cap is again higher than f, and its parent cannot be u. Thus every possible mate path exposes a forbidden Q-Q edge,

contradicting pairedness. This excludes L-S and C-S. The full expanded argument is also recorded in the late-parent technical note [6].

8.4 Exhaustion and termination

For two distinct pure f blocks choose an actual borrow anchor in each. The possible ordered types and their exclusions are

Pair type	Reason
C-C	Highest anchored pair, Corollary 7.2
L-C	Lower-to-inside cap
E-C	Early-parent cap
L-R, E-R, C-R	Section 8.2
R-S	Actual sibling entry and Lemma 7.3
L-S, C-S	Section 8.3 and the full sibling-mate argument

All other types are forbidden by the fixed incidence order, the single lower/upper borrower restrictions, or the incompatibilities of the parent entry. Macros may contain two actual borrows of f; selecting anchors in distinct blocks does not change any sandwich or case in this table.

There is therefore no highest active resource in a fully reduced nontrivial core. Such a core would have a paired interior vertex and hence an active resource, a contradiction. The core contains only σ,τ, or no terminal path exists. Every application above concerns an expanded path in the normalized core; it can be extended to a complete root-to-root raw path, since the core is source-to-sink relevant. Global provenance supplies its cleanliness and purity properties. The argument continues to hold after budget-table pruning.

Theorem 8.2 (structured reduction termination). The reductions of Section 6 completely solve every endpoint-normalized strict block DAG constructed from the aligned radial ribbon of a preprocessed square board, including the tables and provenances produced by the stated reductions and nonnegative-cost truncation. Their final profiles are the exact maximum values of resource-valid paths at every cost.

This is a geometric theorem about these constructed DAGs. No assertion that all nonalternating or reachability-hierarchical forbidden-pair DAGs reduce completely is used; that broader assertion is false.

9. The complete algorithm and a polynomial bound

The decision algorithm is explicit:

1. Handle zero targets, impossible targets, boundary horses, and the initially enclosed case; cap the budget by the number of wallable cells and test all wall sets of size at most two.
2. Construct the retained coordinate graph, its finite rotation maps, radial components, and literal cell anchors.
3. For each permitted root and corner, construct the prescribed BFS/DFS frame, the actual wall gates, and the cell-prize cochain. Materialize its spoke values once. Enumerate the turns, real arcs, singleton and pair pieces, and cache each piece's endpoints, resource block, wall cost, and signed reward.
4. Enumerate the first piece, last piece, terminal automaton state, and real root-boundary arcs. For each valid branch, impose the root and endpoint resource conditions and build its numerical profile table.

5. Compute the reduction schedule and exact-cost terminal profile. Add the first piece's cost and reward once, then accept if any total cost at most the capped budget has value at least M+K.

Soundness follows from distinct emissions, simplicity, the independent local-ear theorem, the corrected horse bonus, and exact numerical accounting. Completeness follows by choosing an important optimum, its uniform maximal noose, a root outside the paired windows, an exterior corner, and the corresponding strict endpoint-normalized path. Theorem 8.2 makes every branch optimization total.

The following deliberately generous bounds match the literal finite compiler in Lean. They do not rely on deduplicating its catalogues or on an optimized representation. Set

```
U = 8m                  upper bound for faces, walls, and spokes
S = 4U(U+1)               contour length
T = 2U(U+1)²              number of turns
P = T + T³              number of pieces
A = 4U² + US²             number of real arcs
Q = 4U + 2              number of automaton states
N = SPQ                 number of compiler vertices
J = P²QA²               number of branches for one root/corner
```

There are at most U roots and at most U corners at each root. In particular, $N=O(m^{12})$ and $J=O(m^{29})$. The repeated-index copies and the triple piece enumeration are intentionally included in these bounds.

For a branch with at most N vertices, Q states, and A arcs, the measured primitive-work bound is

$$
\begin{aligned}
&R(N,Q,A,B) = \\
&\quad N^2(18Q + 5A + 11B + 976) + N(3Q + 210) \\
&\quad + (N+1)(9N^3 + (14B^2 + 62B + 209)N^2 + 140N + 63) \\
&\quad + 28Q + 35B + 167.
\end{aligned}
$$

This includes endpoint encoding, initial edge/resource/profile tables, reachability recomputation, trimming, bypasses, profile convolutions, the source-piece shift, and the final threshold scan. Each nonterminal pass removes a live vertex, so the schedule has at most N such passes. This bound holds without assuming that the search succeeds.

Preparation is also counted. The frame has measured place, endpoint, and gate queries with bounds 64m+6, 2, and 10, respectively. The local score computation uses those queries and an array lookup. The polynomial preparation bound counts turn generation, tree/non-tree arc generation, the complete piece enumeration, summary caching, and the complete branch list. All of its definitions are finite sums, products, and fixed powers of the five arguments. The fully measured compiler satisfies

$$
\begin{aligned}
&\text{work} \le \text{preparationBound}(U, \text{place}, \text{endpoint}, \text{gate}, \text{score}) \\
&\qquad + J \cdot (R(N,Q,A,B) + 2) + 13.
\end{aligned}
$$

The final 13 includes the explicit dimension wrapper. Its stored faces/walls/edges arguments control the loops; the implementation does not recover those integers by rerunning the earlier orbit construction. Lists, orbit tables, canonical-tree locations, prize rows, routes, cochain values, and piece summaries are shared once prepared.

The remaining frontend bounds account for shared board preparation, per-root component and wall preparation, at most U corner calls per root, and the measured compiler above. Each frontend bound is again an explicit polynomial in m. The direct small-wall branch has a separate polynomial bound, including

its actual finite reachability evaluator. Thus the whole decision procedure uses polynomially many primitive operations. The large polynomial is a complexity upper bound, not a claim that this proof implementation is practical.

The polynomial coefficient and exponent are uniform over the side length, board, budget, and target. The counted solver has no supplied geometric certificate or unbounded callback assumption.

This is also a bit-model bound. A spoke coefficient has magnitude at most $L=(2m+1)16m$; a piece's signed reward has magnitude at most $4L+2$. Every finite profile entry has an original forward-path witness. That path contains at most N vertices, which bounds both retained values and finite convolution candidates by a polynomial in m. The source shift adds one more bounded piece. Array indices, costs, loop counters, and all signed values therefore have $O(\log m)$ bits after preprocessing. The formal operand analysis gives the explicit signed bound $54 + 14\cdot\mathrm{bitlength}(m+1)$ for the actual prepared profile's values and their arithmetic intermediates, as well as its measured scorer and threshold. Ordinary binary addition, comparison, multiplication, and division take polynomial time in that bit length. Reading and comparing the original binary B,K adds polynomial time in their encoded lengths; an enormous binary budget never controls a loop, because it is capped first.

The maximum score follows by binary search over $0,\dots,m$. A wall witness can be recovered with at most m more decision calls. While the remaining budget is positive, consider each remaining grass cell x and test the board with x changed to water and the remaining budget reduced by one; commit that wall exactly when the target remains feasible. If the budget reaches zero, feasibility already says that the committed walls suffice. A rejected cell cannot belong to any later feasible completion: such a completion, together with the walls subsequently committed, would have certified the earlier test. At the end, any still-needed wall would contradict its rejected test, so the committed walls themselves attain the target. This proves the optimization and witness corollaries using only the decision procedure. The exported Lean interface exports the decision Boolean; these corollaries do not require an executable provenance-extraction interface.

Part II. Arbitrary plane graphs

Let a finite undirected graph be given explicitly with a plane embedding, a designated exterior face, and a distinguished vertex h. Let T be the vertices incident to that face in the input embedding. A wall set $W \subseteq V \setminus \{h\}$ has unit cost $|W|$. Its score is zero if the component of h in $G - W$ meets T, and otherwise is the number of vertices in that component, including h.

Theorem. Given nonnegative binary integers B, K, deciding whether at most B walls attain score at least K is polynomial-time solvable. There is no restriction on vertex degree. The algorithm also recovers an optimal wall set. Literal parallel incidences can be retained during the plane-map constructions; loops and redundant original parallel edges can be removed after fixing T.

1. Preserve the objective while simplifying neutral regions

The escape set remains fixed throughout preprocessing. If $K = 0$, the empty wall set succeeds. If $h \in T$, every legal placement has score zero. If the initial horse component misses T, that entire component is the optimum and needs no walls.

In the remaining case put

$$Q = C_h(G - T), \qquad E = T \cap N_G(Q).$$

Keep $Q \cup E$ and the edges with at least one endpoint in Q. Projecting any wall set onto the retained vertices preserves its score. Every successful horse component lies in Q, and the retained adjacent escape vertices detect exactly the first possible escape. The marks E remain cofacial. If $B \geq |E|$, walling all of E saves all of Q and is optimal. The marked-instance formalization [7] proves these statements directly from finite reachability.

Give retained vertices reward one. During the following operations a vertex may acquire a larger positive integer reward; wall costs stay one. A neutral region contains neither the horse nor a marked escape and has one or two outside neighbors, called its poles. At a given pole pair, combine all neutral components having exactly those two poles.

Every successful wall placement can be normalized to leave such a region unwalled, without increasing cost or losing a previously saved vertex. For one pole, simply restore its internal walls. For two poles, either restoration is safe or it joins the old horse side through one pole to an escape side through the other. Wall the latter pole and restore the region. At least one internal wall was removed, and the new pole wall lies outside the old horse component. This pole is wallable even when it is a marked escape; the artificial protected target has not yet been added.

It follows that a one-pole region can be folded into its pole's reward. A two-pole bundle can be replaced by one degree-two vertex of the bundle's total reward. A direct edge between its poles can also be deleted, since the restored bundle supplies the same connection. Normalization in both the original and replacement graphs proves equality of the optimum at every upper-bound budget, including against walls on the replacement vertex. Reverse expansion uses no greater cost and saves no smaller reward.

Apply only vertex- or edge-decreasing rewrites: a singleton two-pole bundle with no direct pole edge is already saturated. Repeat until none applies. All candidate components can be found by trying every one- or two-vertex pole set and running graph search. Positive rewards are sums of disjoint original vertex bags, so their total stays at most the original vertex count. Replacements follow old embedded paths and retain cofacial escape marks. The complete graph and embedding arguments are in the saturation proof [8]; the Lean compression layer [9] proves the graph-level optimum equivalences.

Finally place a protected target t in the common escape face and join each marked escape to it by two parallel edges. Call this plane host Γ. Successful enclosure is exactly separation of h from t, with the same wall cost and positive reward.

Two properties of this saturated host will govern every local candidate, including candidates later rejected by global constraints:

1. A wall that alone does not separate h,t has one corner at each incident face.
2. Two walls that together do not separate h,t have at most two common face labels. If there are two, their incidences are consecutive at both walls.

For the second statement, q common faces give an embedded $K_{2,q}$ in the vertex-face radial graph. Its q complementary disks separate primal paths avoiding the two poles. The connected h,t component, and all surviving marked escapes, occupy one disk. Every other disk must contain a neutral two-pole vertex or a direct pole edge: without either, its two face centers would belong to the same primal face. Saturation permits at most one such object in total. Thus $q \le 2$. The same argument identifies the two-face disk as one edge or one two-edge representative path, proving consecutiveness. The one-pole version proves the unique-corner statement.

2. Represent an optimum by a forward radial walk

The radial graph has a vertex for each primal vertex and face and a spoke for each literal corner incidence. Delete the radial stars at h,t. A separating noose alternates wall vertices and face centers. Zero-, one-, and two-wall solutions are checked directly; the remaining search uses at least three wall events and rejects every singleton or consecutive wall pair already separating h,t.

Choose an optimal inclusion-minimal separator and, among its representing nooses, one with a maximal horse-side disk. Strictly positive rewards ensure that enlarging the horse component strictly improves its reward. The shortest-tree completeness argument [10] shows that radial shortest paths from a suitable noose face root to the noose remain in this disk. An exterior shortest excursion could replace a noose arc with no greater wall cost, either improving the horse component or enlarging the chosen disk.

For each radial face root and corner, compute ordinary unweighted distances and run DFS along increasing-distance spokes, scanning in the order opposite to the clockwise contour. Cutting along this tree gives a disk with a linearly ordered contour. Tree spokes have their two boundary uses; every non-tree spoke is forward in its increasing-distance direction. Non-tree chords have no strictly alternating endpoints. The selected optimal noose appears as a forward path in one such ribbon. All allowed radial components are considered separately.

At a wall event i, let J_i contain its incoming and outgoing face labels and every label in its actual closed inward fan. Labels are deduplicated, but an inward incidence remains included even if another event later uses that spoke. With cyclic indices define

$$E_i = J_i \setminus J_{(i-1)}.$$

Require the local guard $J_{(i-1)} \cap J_i \cap J_{(i+1)} = \varnothing$ and pairwise disjoint emissions E_i. A label's emissions count its cyclic runs of fan support. These tests therefore say that each face occurs at one wall event or at two consecutive events. Every minimal separating noose passes them.

These same conditions force any accepted forward walk to be simple. Two visits to a face would give it at least three supporting wall events. If a wall repeats, the literal contour order puts the incoming face of its later run inside its earlier inward fan. That face is also present at the two walls adjacent to the later visit,

again giving three support events. The local different-neighbor tests make these events distinct. The Lean path and ownership proofs [11] derive simplicity without a degree bound or an inward-incidence cap.

3. Compute the exact score with fixed local values

Activate every selected wall's complete star inside an accepted simple noose. Closed-inward ownership makes the resulting graph consist of the noose together with pendant spokes and paths between consecutive wall pairs. Its inside has one central region incident to every selected wall. Each other region has one pole or two consecutive poles. Inserting the extra paths successively proves this: each cuts off a cap containing no other boundary wall, and repeated spokes create one-pole bigons.

For a candidate whose chosen disk contains h and excludes t, the horse lies in the central region. Otherwise a side region's one or two poles would be a separator rejected by the local filters. Each selected wall has a primal neighbor in the central horse component A: a radial region with both endpoints of its primal edge walled is an isolated quadrangle with at most two wall poles, so it cannot be this central region of a noose with at least three walls. Consequently $W=N_\Gamma(A)$, even for an accepted nonminimal separator.

Every other inside component is an orphan with one or two poles. It is already a component after deleting just those poles and is therefore present in a precomputed component table. Conversely, a charged table component cannot contain another selected wall: that wall's edge to A would connect it to the horse after deleting only the listed poles. Thus the table subtracts exactly the inside orphans, once each. Literal attachment darts determine their side, including the cyclic last/first two-pole charge.

Let total retained reward be R and put $M=R+1$. Give the horse an extra charge M and put the negative total charge at t. A dual-tree cochain converts disk charge into an additive boundary sum. Fixed boundary-anchor corrections remove selected-wall rewards, and the component tables remove orphan rewards. Every correctly oriented accepted candidate has value

$$M + \sum_{v\in C_h(\Gamma-W)} \mathrm{reward}(v).$$

A horse-outside disk cannot reach M, and a target-inside disk cannot pass the positive threshold. All cochain, anchor, and module values are computed before optimization. The weighted implementation and audit [12] include negative local terms and simultaneous one- and two-pole charges.

4. Optimize the global resource constraints in polynomial time

A state remembers the preceding, current, and following actual wall turns. Their literal real arcs determine the intervening face turns. This suffices to compute the local guard, fan difference, and score. Fix the first, second, penultimate, and last turns in a branch, allowing the second and penultimate to coincide for three-wall paths. The two endpoint emissions are now fixed. Reject their overlap and remove other states meeting their union.

Write $I_f=[L_f,U_f]$ for a face's tree-subtree contour interval and order face resources lexicographically by

$$\rho(f) = (\mathrm{depth}(f), -L_f).$$

For the face g_i visited after wall i, transfer

$$F_i = \{f\in J_i\cap J_{i+1} : \rho(f)\geq\rho(g_i)\}$$

to the following wall block. Disable the transfer at the closing cut. The block's new emission is $E'_i=(E_i\setminus F_i)\cup F_{i-1}$. The triple guard implies $F_i\subseteq E_i$; retained and incoming parts are disjoint. Every

original occurrence is kept or transferred exactly once, preserving the complete multiset of resource occurrences on valid and invalid paths. Group each incoming face with its following wall. Its moved labels use that actual face as their canonical anchor; kept labels use the wall.

Create a forbidden pair between every two graph vertices sharing a resource, and add empty source and sink dummies. Apply the following sound reductions: trim irrelevant vertices, discard unreachable pairs, delete forbidden arcs, and bypass an unpaired interior vertex. For each arc and exact cost retain the best signed value and a representing original path. A bypass composes profiles by max-plus convolution.

The termination theorem in Appendix A proves that these rules cannot leave a nontrivial relevant core. Its argument chooses a highest-priority active face f, meaning that two surviving owners are comparable. Every retained arc expands to an original path. Such an expansion cannot contain a forbidden pair with an eliminated endpoint: that endpoint could not have been unpaired while its comparable mate survived. Hence the expansion through the chosen pair repeats no higher resource.

There are five possible locations for an owner of f: a canonical anchor inside I_f; the parent wall before or after I_f; or a non-tree neighbor at the lower or higher adjacent depth. They exhaust all owners, including labels moved from incoming faces. The proof checks all 25 ordered endpoint combinations.

Inside anchors trap every intermediate block in I_f, where its labels have priority at least f. Outside chords confine intermediate walls to that interval and appropriate bands of child subtrees. Saturation also makes a returning wall own every child face through its incoming child. A return to a wall whose children all outrank f would therefore repeat a higher label. Parent-band cases either obtain the same rank bound or show that the later endpoint owns every possible active middle label. These statements use arbitrarily large child bands and emission sets.

Every surviving middle vertex is paired through some active label. The rank cases force it to own f, exposing a forbidden first arc. The coverage cases expose a forbidden last arc, even when the middle vertex's mate is on another branch. The remaining endpoint orders are impossible by contour order or repeat a higher label. Thus every nontrivial core admits another sound reduction. The root resource is inactive: endpoint normalization leaves it only at the fixed last wall, and its low priority prevents an internal transfer.

5. Size, bit complexity, and recovery

Let q be the number of vertices plus edges in the normalized augmented graph. Then there are $O(q)$ root/corner ribbons and $O(q^2)$ actual turns. A branch has $O(q^6)$ grouped states, and there are $O(q^8)$ endpoint branches per ribbon. Face sets are stored explicitly; the construction never enumerates arbitrary subsets of a wall's faces.

Unit wall costs permit capping B by the number of current wallable vertices. Each ordered state pair has at most $B+1$ exact-cost entries. There are at most as many bypasses as states; other changing operations remove vertices, pairs, or whole arcs. Truncation occurs when entries are offered. Reachability recomputation, convolution, and explicit witness copying give, for example, the conservative $O(q^{45})$ arithmetic-operation bound in the compiler analysis [13].

Retained rewards sum to at most the original vertex count. Cochain, component-table, local, and path values therefore have polynomial bit length. Neither a binary budget nor a binary threshold requires a pseudopolynomial loop. Preprocessing and all reverse-recovery records also have polynomial size.

The maximum terminal profile over all branches and the direct small-wall cases gives the optimum. Expand its stored path to a noose, read off its wall set, and reverse the neutral-region rewrites. Each reverse step uses no greater cost and loses no saved reward. The resulting original wall set attains the computed optimum, completing the mathematical algorithm.

Appendix A. Planar resource termination

Every endpoint-normalized branch of the saturated ranked incoming compiler is completely reduced by the sound forbidden-pair rules. A nontrivial source-to-sink core with no applicable rule cannot remain. The result has no maximum-degree hypothesis and does not use the important-wall inward gate. It also holds for the retained-arc graph of exact-budget signed-value profiles.

This is a structural optimizer theorem. The plane preprocessing, enclosure completeness, and exact weighted score enter through the construction and scoring arguments in Part II.

1. Structural input and the compiled branch

Let Gamma be the normalized augmented plane multigraph, with protected horse h and target t. Its face labels are the actual faces after target-fan insertion and before deleting terminal radial stars. They are not labels for the original unsplit outer face.

Use the vertex-face radial multigraph with the stars at h,t deleted. Fix one connected component and a face root r in that component. Terminal-star deletion may disconnect the allowed radial graph; the theorem is applied separately to each root component. An isolated component with no wall turn supplies no branch.

The following are the exact structural hypotheses.

1. **Saturated eligible incidences.** A selected wall is neither a protected terminal nor a singleton h-t separator. Every face has a unique corner at such a wall. For every allowed pair of distinct consecutive walls, any two common faces have consecutive actual incidences at both walls. The pair is allowed only if it does not separate h,t.
2. **Canonical ribbon.** Compute ordinary unweighted distances from r in the chosen radial component. Run DFS on the increasing-level spokes, scanning counterclockwise in the order exactly opposite to the clockwise ribbon order, with the corresponding fixed root corner. Cut along this tree. Every non-tree spoke is forward exactly in its increasing-distance direction. Non-tree chords do not have strictly alternating ports.
3. **Literal local paths.** A complete path is a forward alternating sequence of actual wall and face turns, beginning and ending at r, with at least three wall events. Consecutive turns use the same real spoke. Immediate reversals are excluded. Every wall passes the singleton filter, every cyclic consecutive wall pair passes the distinct-wall and pair-separator filters, and the closed inward fans satisfy the cyclic three-fan guard below.
4. **Exact finite-state compilation.** States remember the preceding, current, and following actual wall turns and the outgoing actual face turn. Fix the endpoint data and perform the endpoint normalization in Section 2. Ranked incoming grouping preserves exactly the complete paths of that normalized state graph.
5. **Resource-pair semantics.** Every pair of distinct interior graph vertices sharing an emitted face label is forbidden. The two graph terminals are empty dummy vertices. Reductions use the actual sound trace and path-profile operations of Section 4.

The first hypothesis follows from saturation [14]: neutral one-pole regions are absent; neutral two-pole regions contain at most one degree-two representative; and its poles have no redundant direct edge. The connected h-t component lies on one side of each common-face radial four-cycle. Its other side contains either that representative path or the one direct edge. This gives the consecutive incidences. An eligible wall cannot be an articulation, which gives its unique face corners. These are properties of the host and the local pair filter, including on paths that fail global resource disjointness.

The second hypothesis is the degree-independent canonical orientation theorem [15]. The remaining hypotheses are explicit compiler operations; they do not assume a globally valid noose, resource uniqueness, a laminar resource family, or reduction termination.

Write I_v=[L_v,U_v] for a tree-subtree contour hull. Equal-depth vertex hulls are disjoint. A vertex port lies in I_v exactly when its original vertex is v or a tree descendant. The face priority is the strict finite order

```
rho(f) = (depth(f), -L_f),
```

maximized lexicographically. Radial adjacency always joins successive distance levels.

2. Ownership, endpoint normalization, and the root label

Let the cyclic wall events be B_0,...,B_(k-1), with k>=3. Their closed inward face sets J_i contain the incoming and outgoing visited faces and every face incidence in the literal inward fan, deduplicating labels. The local guard is

```
J_(i-1) intersection J_i intersection J_(i+1) = empty
```

at every cyclic index. Put

```
E_i = J_i minus J_(i-1).
```

The first and last actual wall turns, and the neighboring actual wall-turn data used by their states, are fixed in a branch. Their sets E_0,E_(k-1) are therefore fixed. Reject their intersection, prepay their union, remove every interior state whose old E meets that union, and trim to the selected endpoints.

For the actual nonroot face g_i visited from B_i to B_(i+1), define

```
F_i = {f in J_i intersection J_(i+1) : rho(f) >= rho(g_i)}.
```

Set F_(k-1)=empty at the cut, leaving the closing root-face ownership with the last wall. The ranked incoming emission is

```
E'_i = (E_i minus F_i) union F_(i-1),
```

where the incoming transfer to the first wall is empty. The local triple guard makes F_i a subset of E_i; the two parts of E'_i are disjoint. The rewrite moves an existing label occurrence across one face interface and preserves the complete resource multiset on every path.

The compiler construction [16] first splits off the face node carrying F_i, merges copies with the same two neighboring actual wall turns and actual face turn, then duplicates that face into each following wall-state copy. The merge is exact because all incoming copies have identical successor sets. The resulting block is its actual incoming face followed by its current wall. Every original wall-state path has exactly one such grouped path and conversely. This uses local stored turns, with no choice of an arbitrary subset of a wall's incident faces.

Add empty dummy terminals around the fixed first and last wall states. Keep every reassigned label on its wall block when constructing the pair relation. Thus the proof can track resources through all wall blocks, including the fixed boundary blocks.

The root-face label cannot be active. The closing visit puts r in J_(k-1) and J_0. The triple guard excludes it from J_(k-2), so r belongs to the prepaid old set E_(k-1). Every interior old emission of r was removed by endpoint normalization; E_0 does not contain it. Moreover every internal visited face has positive depth and outranks r, so no internal F_i contains r. The closing F_(k-1) is empty. Consequently the new graph emits r only at its one fixed last-wall vertex. It has no resource pair and cannot become active during reductions.

3. Literal anchors and the exhaustive endpoint classes

A kept label in E_i minus F_i uses the current wall turn as its canonical anchor. A moved label in F_(i-1) uses the actual incoming face turn g_(i-1) as its canonical anchor. Its priority is at least the priority of that face. Every emitted label is also incident with the block's current wall.

The block span begins at the incoming port of its first actual turn and ends at the outgoing port of its wall turn. The first wall block has no incoming nonroot face. All canonical anchors and current-wall ports lie in the span. On every original transition,

$$\mathrm{hi}(A) < \mathrm{lo}(B).$$

The same order holds for surviving blocks joined by shortcuts. A block has only one incoming face turn, so it cannot contain two different actual visits to a particular face.

Local rank. If a canonical anchor of an emitted label g lies inside I_f, then g=f or rho(g)>rho(f).

For a wall anchor, adjacency gives depth(g)>=depth(f). Equality makes the wall a direct child of f; any other equal-depth incident face has an earlier hull by the non-tree orientation theorem. For a moved face anchor at q inside I_f, its priority is at least that of f, and the transfer rule gives rho(g)>=rho(q). This proves the statement for arbitrarily many labels in one block.

Fix a nonroot emitted label f. Classify one of its owning blocks as follows, checking the inside class first.

Class	Exact condition
I	Its canonical f anchor lies in I_f.
L	Its current wall is a non-tree neighbor of f at depth depth(f)-1.
E	Its current wall is u=parent(f), with its wall turn before I_f.
R	Its current wall is a non-tree neighbor of f at depth depth(f)+1.
S	Its current wall is u=parent(f), with its wall turn after I_f.

These conditions are exhaustive. In particular, every owner at a wall inside I_f is of class I. Such a wall must be a direct child of f, since it is adjacent to f. A kept label has its wall anchor inside I_f. If f was moved from an incoming face q, then rho(q)<=rho(f). Adjacency at that child wall gives depth(q)=depth(f) or depth(f)+2. The larger depth is impossible. At equal depth, q!=f would put I_q before I_f and make it outrank f. Therefore q=f, and the moved canonical anchor is the actual visit to f inside I_f.

If the current wall is neither parent nor child of f, its hull is disjoint from I_f, and adjacency gives the two depths in the table. The canonical non-tree orientation puts the entire L hull before I_f and the entire R hull after it.

At the parent wall, the unique f incidence is the child-tree spoke. A turn spanning across I_f puts this incidence strictly in its outward interval, so it cannot own f. An incoming visit to f is already of class I. An

outgoing visit moves f to the following block and cannot leave ownership at the present block; the closing root exception is inactive by Section 2. Every remaining parent occurrence is therefore E or S, and its f spoke is strictly inward.

The same used-spoke argument shows that the L,R occurrences use their f spoke strictly inward. These are the pure outside occurrences needed by the chord-band lemmas. Endpoint blocks may straddle hulls; the classification does not assign a moved label to its current wall in place of its canonical face anchor.

4. Reduction traces and higher-resource cleanliness

Use the following sound operations on the retained directed acyclic graph:

- trim vertices outside a source-to-sink path;
- discard forbidden pairs whose endpoints are no longer reachable;
- delete an arc whose endpoints form a forbidden pair;
- bypass an unpaired interior vertex by composing its incoming and outgoing profiles, retaining only allowed endpoint pairs.

The last operation uses exact nonnegative costs and arbitrary signed values. It retains a maximizing original-path witness at each exact cost. Skipping a composed shortcut whose endpoints are forbidden is the same legal operation as forming and immediately deleting it. Budget truncation removes entries and possibly arcs; it cannot add reachability between surviving vertices.

The actual trace supplies two provenance invariants.

1. Every retained arc entry has an original forward-path expansion. Surviving block spans retain their original strict order.
2. No expansion of a current full path contains an original forbidden pair with an eliminated endpoint. At the first elimination of either endpoint, its mate would still be present and comparable on the trace expansion, so it would not be unpaired. Prior deletion of an unreachable pair cannot help, since reachability between surviving vertices never increases. Full-path trimming cannot remove a vertex needed on that retained expansion either.

These are trace statements, not an assumption that a reducible interior vertex always exists. The generic finite-resource versions are audited in Resource-rank invariants [17].

Call f active when two distinct surviving occurrences are comparable. The trace keeps the original shared-resource pair semantics on comparable surviving vertices. If f has highest active priority, every full expansion through its selected pair is higher-clean for f: two original owners of a higher label would both still be alive by the retired-pair invariant and would form a higher active pair.

The multiset-preserving ownership rewrite also gives a useful single-label consequence. A label owned at most once has exactly one cyclic run of J support, of length at most two if nonempty. The old emission count is the number of support runs; the triple guard forbids longer runs and the all-one word. Hence a higher-clean label has at most two consecutive supporting wall events and at most one actual face visit. The latter uses the fact that every internal visit is owned by its following block and a block contains only one face visit. The selective ownership lemma [18] establishes the same implication for one chosen label without requiring other labels to be feasible.

5. The geometric progress statements

The following statements apply to the full, possibly resource-invalid expansions just described. Their hypotheses are precisely Sections 1-4.

Anchor rank. Two comparable class-I owners of a highest active f are impossible in a reduced core. Every strict middle block's whole span lies in I_f. It must have an active resource, and the local rank statement makes that resource f. The first arc is therefore forbidden. A direct endpoint arc is forbidden too.

Child-prefix ownership. A block at wall u returning through an actual child Q owns every child face of u at or before Q. The returning closed fan contains that prefix, and incoming rank transfer assigns all predecessor-shared prefix labels to this block. Onward sharing is impossible: the incoming child is excluded by the triple guard; an earlier child and a later outgoing face would violate consecutive common-face incidences; an outgoing parent face would require a backward increasing-level next-wall spoke. See Section 16 of companion note [16].

No return to a higher-child wall. If every child face of a wall x outranks f, a higher-clean complete path has at most one event at x. Its first departure, if it were to return, would visit a child H. A later return would own H by the prefix lemma, as would the distinct block following the earlier visit. Every L,R wall has the required higher-child property. This makes its unused f spoke a global chord barrier.

Non-tree mixed rank. For L-I or I-R, every strict middle owned label has priority at least f. The unused non-tree chord confines the middle wall anchors to I_f and a band of higher child hulls of the outside wall. The proof allows any number of child hulls and both endpoint-span orientations; it is in the non-tree mixed-cap theorem [19].

Parent rank and coverage. An early parent owner followed by an inside anchor enters a child H before f, of higher priority. It cannot return to the parent because that would own H again. All middle labels have rank at least f. For a late parent return through Q, the endpoint owns all child faces from f through Q. A middle wall in those child hulls has only higher labels or labels owned by that endpoint; an intermediate parent run owns f. These are the parent-band conclusions [16].

Pure non-tree progress. The orders L-L, L-E, R-R, and S-R are impossible on a higher-clean complete path. The one-sided chord bands cannot contain an endpoint wall of the required depth. For L-R, all strict middle labels have rank at least f. Its exceptional incoming face would create three supports of a higher label, or reduce to one immediate two-wall ownership interface. The outside-pair proof [20] uses the single-label support statement, without assuming that the possibly nonconsecutive endpoint walls pass a pair-separator test.

Parent/non-tree progress. For E-R, all strict middle labels have rank at least f. For L-S, each strict middle block has only higher labels or labels owned by the later endpoint, except that a middle parent run owns f directly. Both statements, including their direct-face exceptions, are proved in the parent/non-tree cap theorem [21].

None of these proofs uses a bound on a wall's degree, its number of unused inward incidences, or the size of an emitted resource set.

6. Complete ordered endpoint matrix

Suppose a nontrivial reduced relevant core exists. Every interior vertex is paired, so some resource is active. Choose a highest one f from the finite face order and a directed path between two of its surviving owners. The root label is unavailable by Section 2. Classify the two blocks by Section 3, with the earlier owner as the row and the later owner as the column.

Earlier / later	L	E	I	R	S
L	Chord order exclusion	Chord order exclusion	Mixed rank	Outside rank	Late-endpoint coverage
E	Same-depth chronology	Higher child repeats	Early-parent rank	Parent/upper rank	Higher child repeats
I	Chronology	Chronology	Anchor rank	Mixed rank	Late-endpoint coverage
R	Chronology	Chronology	Chronology	Chord order exclusion	Late-endpoint coverage
S	Chronology	Chronology	Chronology	Chord order exclusion	Late-endpoint coverage

The entries mean the following exact contradictions.

- A rank entry puts every strict middle label at f or above it. Every surviving middle vertex is paired through an active label. Highest priority makes it an f owner, exposing a forbidden first arc. A direct endpoint arc is already forbidden by f.
- A coverage entry makes an active label of every strict middle vertex belong to the later endpoint. Its last arc is forbidden. This remains true when the active mate that establishes pairedness lies on another branch. The generic endpoint-coverage argument does not require that mate on the chosen path.
- A higher-child entry puts the same higher label at two distinct blocks of the full expansion, contradicting higher-cleanliness.
- A chord-order entry contradicts the corresponding one-sided band and depth argument before reduction is considered.

Three bridges in the matrix deserve explicit verification.

E-L **chronology.** An L wall x and u=parent(f) both have depth depth(f)-1. They are different, so their hulls are disjoint. Since I_x precedes I_f and I_f lies in I_u, the whole I_x precedes the whole I_u. A parent-wall block cannot reach a later L wall. The other chronology entries compare a current-wall port before or after I_f with a canonical inside anchor, or compare two opposite sides. Strict block-span order justifies those comparisons even when an endpoint block straddles the hull.

R-S **localization.** Let the earlier R wall be y, with a later parent-wall event at u. A port of y follows I_f and precedes that later port inside I_u; tree-hull nesting forces y to be inside I_u. Its depth is two greater than u, so its parent is an actual child face H of u. Since y is not a child of f, H!=f; the contour order gives H after f. If f was kept at y, its canonical wall anchor is in I_H. If it was moved from the incoming face q, adjacency at y and rho(q)<=rho(f) give depth(q)=depth(f). The face q is not f, since that would make the block class I. Thus transfer gives L_f<L_q. Adjacency at the child wall y of H gives L_q<=L_H: equality means q=H, and otherwise the non-tree orientation puts q before H. Consequently q is itself a later child face of u, between f and H. Its actual incoming-face anchor lies in I_q. The broader late-parent coverage theorem applies to the canonical anchor in either case.

Repeated parent-wall owners. If the first parent occurrence is early, any departure that can reach that wall again is an actual child H before f; visiting f would move ownership away, and a departure outside I_u could not return. The visit to H and the later return both own this higher label. This excludes E-E and E-S.

If both parent turns are late, the first must depart through an actual child H after f to reach the second. Every intermediate wall is in a child hull from H through the second incoming child Q, or is another parent run. The later endpoint owns the entire prefix through Q, so endpoint coverage applies. This does not

assume that the first late turn itself arrived through a child: its incoming face may be earlier while its wall port is late. For any subsequent return, an incoming non-tree source-face anchor before l_u would be too early to follow the first block.

All 25 ordered cells are excluded. Therefore no highest active resource exists in a nontrivial reduced relevant core, contradicting its paired interior vertex.

7. Termination theorem

Theorem. For every branch satisfying Sections 1-4, exhaustive application of the sound forbidden-pair reductions ends either with no source-to-sink path or with only the two dummy terminals and their exact-budget profiles. It cannot stop at a nontrivial relevant core. The theorem holds for every saturated ranked incoming branch, at every vertex degree, with or without the important-wall gate.

The preceding matrix excludes the only nonterminal fixed point. A bypass decreases the number of vertices. Between bypasses the other operations only delete vertices, pairs, arcs, or profile entries; only finitely many are present. There can therefore be only finitely many changing steps, and any exhaustive run reaches one of the two stated terminal forms.

Signed values do not enter the structural argument. With budget truncation, each remaining arc still has a forward original-path entry and the same resource-trace semantics. Truncation only decreases reachability, so all rank, coverage, and cleanliness arguments remain valid. The theorem applies to each such retained graph, including a graph obtained after removing additional local turns by the important-wall gate.

This proves termination of the resource optimizer. Bounding a binary input budget by the relevant unit-wall count and accounting for the number of compiler states, branches, and profile operations are given in the compiler bounds [22]. No numerical budget bound is used to exclude a structural core.

Appendix B. Verification and reading notes

The square-grid and arbitrary-plane formal developments connect the original input semantics to the executable decision Boolean. The result is an equivalence: the procedure accepts exactly when a legal wall placement within the budget reaches the requested score.

For both procedures, the polynomial coefficient and exponent are fixed before the input is chosen. The work theorem measures explicit source-level primitives, including preparation and search. Separate bounds cover the integers and finite labels consumed by the actual search. Large binary budgets are capped before profile allocation; thresholds are read and compared without a loop proportional to their numerical value.

The recorded principal axiom checks use only Lean's standard foundational axioms. They report no unfinished proof or problem-specific axiom in the principal claims. Recorded exhaustive comparisons and witness flood fills test implementations and particular instances. They supplement the proofs rather than determine the complexity classification.

The mathematical running-time conclusion uses ordinary binary representations and implementations of the stated primitives. The formal project does not verify the compiler, garbage collector, or an instruction-by-instruction machine trace. Source and generated-code reviews examine whether the implementation retains and shares expensive prepared data as the analysis assumes.

The $O(q^{45})$ bound in Part II is a conservative arithmetic-operation bound for the mathematical compiler, where q counts normalized augmented vertices and edges. It is not stated as the exponent of the instrumented whole-program implementation. Neither this bound nor the larger square-grid bounds imply that these prototypes are practical large-board solvers.

Optimization and wall recovery follow from the mathematical arguments. The exported formal interface returns a decision Boolean and, in the measured versions, counted work. This document does not claim a formally exported end-to-end wall-returning optimizer.

This edition was prepared from the completed consolidated proof notes and the relevant research transcript, with source reviews of both theorem statements. The historical build and audit results are reported from that record; the full formal project was not rebuilt for this edition. The PDF contains the consolidated arguments and an expanded planar termination appendix. It does not embed the formal source or every companion incidence-level proof note.

Companion proof notes

Numbered references below identify the project's unpublished technical notes cited in the exposition. They preserve the mathematical dependency trail. They are not external hyperlinks, and the notes themselves are not bundled in this PDF. References to numbered sections within those notes belong to the named note.

1. Non-tree cap technical note.
2. Anchor extension.
3. Early-parent proof.
4. The repeated-borrow lemma.
5. Full two-spoke proof.
6. Late-parent technical note.
7. Marked-instance formalization.
8. Saturation proof.
9. Lean compression layer.
10. Shortest-tree completeness argument.
11. The Lean path and ownership proofs.
12. Weighted implementation and audit.
13. Compiler analysis.
14. Saturation.
15. Canonical orientation theorem.
16. Compiler construction.
17. Resource-rank invariants.
18. Selective ownership lemma.
19. Non-tree mixed-cap theorem.
20. Outside-pair proof.
21. Parent/non-tree cap theorem.
22. Compiler bounds.