hexagon:2611.00633v1

Parameterised complexity of graph homomorphism to odd cycles

Jason Davies*

November 18, 2026

Abstract

A nonbipartite graph has an odd-cycle core precisely when it maps to its shortest odd cycle. Core recognition and homomorphism to an input odd cycle are both XNLP-complete for the width of a supplied path decomposition, XALP-complete for the width of a supplied tree decomposition, and W[SAT]-hard and in W[P] for feedback vertex number. Hardness persists on connected, 2-degenerate N-vertex sources of girth $\Omega(N^{1-\varepsilon})$ for any fixed $0 < \varepsilon < 1$. Our reduction encodes lists by cycles sharing an edge. On inputs with an edge it increases treewidth and feedback vertex number by exactly two. Lean 4 checks this structural transfer. For feedback *edge* number r, we compute the exact circular chromatic number in $2^{O(r)}$ time up to polynomial factors, which gives the same upper bound for both decision problems.

1 Introduction

Does a nonbipartite graph G map to its shortest odd cycle C_h? Gerards studied this question for graph homomorphisms [13]. It is also a recognition problem. The *core* of a finite graph is its smallest homomorphically equivalent subgraph, unique up to isomorphism. If $G \to C_h$, the inclusion $C_h \to G$ gives equivalence, and C_h is a core because every endomorphism of an odd cycle is an automorphism (Lemma 3). Conversely, if the core is C_ℓ for some odd ℓ, then $C_\ell \to G$ gives an odd closed walk containing an odd cycle, so $h \le \ell$. Composing $C_h \to G \to C_\ell$ gives $\ell \le h$ by the same argument in C_ℓ. Thus $\ell = h$. Bipartite graphs have no odd-cycle core.

Theorem 1. *Consider two decision problems on a finite simple undirected graph G.*

- *Does G have an odd cycle as its core?*
- *Given also an odd $n \ge 5$ in binary, does $G \to C_n$?*

Each problem is XNLP-complete for the width of a supplied path decomposition, XALP-complete for the width of a supplied tree decomposition, and W[SAT]-hard and in W[P] for $\mathrm{fvs}(G)$*. All three hardness bounds hold on connected, 2-degenerate graphs of girth $\Omega(|V(G)|^{1-\varepsilon})$ for every fixed $0 < \varepsilon < 1$.*

Deleting edges leads to a different algorithm. If deleting r edges leaves a forest, both problems are solvable in $2^{O(r)}$ time up to polynomial factors. In fact we can compute the exact circular chromatic number $\chi_c(G)$ within this bound (Theorem 15). For odd $h \ge 3$, the test $G \to C_h$ is the circular-colouring threshold $\chi_c(G) \le 2h/(h-1)$ [21]. If h is the odd girth of G, inclusion of C_h gives the reverse inequality. Core recognition therefore tests equality in this lower bound. Corollary 17 also gives polynomial time for both problems at girth $\Omega(N \log\log N / \log N)$.

*jason@jasondavies.com

The hardness proofs encode List Colouring in the geometry of an odd cycle. All reference cycles share an ordered edge AB. Its image determines the coordinates in every copy. A path from an input vertex to a forbidden coordinate then excludes exactly that colour. Each forbidden occurrence gets its own copy and path. Deleting A, B exposes a subdivision of the input graph with trees attached. Contracting selected paths instead gives the input graph joined to the edge AB. These two views of the same graph give matching upper and lower bounds.

Theorem 2. *There is a polynomial-size, logspace reduction from List Colouring on a graph F to $G \to C_n$, with n odd and $n \geq 5$, such that*

$$\mathrm{tw}(G) \leq \max\{\mathrm{tw}(F) + 2, 3\}, \qquad \mathrm{pw}(G) \leq \mathrm{pw}(F) + 6, \qquad \mathrm{fvs}(G) \leq \mathrm{fvs}(F) + 2.$$

Supplied tree or path decompositions can be transformed within the same resource bounds. For every feedback vertex set S of F, the construction specifies a feedback vertex set of G of size $|S| + 2$. If F contains an edge, then the treewidth and feedback vertex number bounds are exact: $\mathrm{tw}(G) = \mathrm{tw}(F) + 2$ *and* $\mathrm{fvs}(G) = \mathrm{fvs}(F) + 2$.

Theorem 1 also gives hardness for minimum width by forgetting the supplied decomposition. Amarilli asked whether bounding the treewidths of both input graphs makes homomorphism fixed-parameter tractable [2]. Here the target always has treewidth and pathwidth two. Since XNLP-hardness implies W[t]-hardness for every fixed $t \geq 1$ [4, Lemma 2], the reduction gives W[1]-hardness for the maximum of the two treewidths even on these targets.

Prior work and attribution. In May 2025 Amarilli reported workshop reductions indicating W[1]-hardness for the parameter $\max\{\mathrm{tw}(G), \mathrm{tw}(H)\}$ [2]. Amarilli later clarified (personal communication) that the reduction from k-Clique produces a homomorphism instance $G \to H$ with source pathwidth $\mathrm{pw}(G) = k$ and target pathwidth $\mathrm{pw}(H) = 2$. Thus the workshop reduction already gives W[1]-hardness for source pathwidth with a target of pathwidth two. Our reduction from List Colouring takes H to be an odd cycle and yields the completeness results for supplied decompositions in Theorem 1.

We use the XNLP-completeness of List Colouring for pathwidth [4, Theorem 15] and its XALP-completeness for treewidth [3, Theorem 8]. For feedback vertex number, Bodlaender, Groenland and Pilipczuk obtained W[3]-hardness for List Colouring and W[SAT]-hardness for Binary CSP [5, Table 1]. Their Binary CSP reduction certifies the truth of subformulas by propagating choices down a formula tree [6, Lemmas 11 and 14]. In Theorem 10 we express that propagation with colour lists and ordinary edges, strengthening the W[3] lower bound for List Colouring in [6, Corollary 34].

Piecyk and Rzążewski studied cutwidth and feedback vertex number for list homomorphism. Their conditional cutwidth bounds also apply to ordinary homomorphism for odd-cycle targets [19]. For each fixed odd cycle our reduction transfers algorithms in the opposite direction, from ordinary homomorphism to List Colouring with a fixed palette (Remark 13). Carbonnel gives single-exponential-time, polynomial-space algorithms measured in the number of source vertices for target families excluding a topological minor, together with improved randomised bounds for fixed odd-cycle targets [7, Theorems 1 and 2]. These bounds concern source order rather than the structural parameters studied here.

Disequality and list gadgets for projective cores are developed by Okrasa and Rzążewski [18, preprint version, Lemma 17 and Proposition 19]; their fixed-target width overhead may depend on the target. Kostochka, Nešetřil and Smolíková prove NP-completeness of fixed-H colouring on 2-degenerate sources for nonbipartite triangle-free H [15, Theorem 1.3]. Here the elimination of lists is uniform and polynomial-size, with additive structural overhead independent of the target length

for all three parameters. The single-link reference copies provide the feedback vertex guarantee as well as connected sources of prescribed large girth.

2 Properties of odd cycles

Conventions. All graphs are finite, simple, and undirected. Path length counts edges. A tree decomposition consists of a nonempty tree T and bags $(B_t)_{t\in V(T)}$ covering the graph's vertices and edges, such that the bags containing any given vertex induce a connected subtree of T. Its width is $\max\{0, \max_t |B_t| - 1\}$. A path decomposition has a path as its index tree. The parameters tw and pw are the minimum widths over tree and path decompositions, respectively. Thus the empty graph has width zero, witnessed by a single empty bag. A feedback vertex set is a set whose deletion leaves a forest, and fvs denotes its minimum cardinality.

We identify $V(C_n)$ with $\mathbb{Z}/n\mathbb{Z}$, with consecutive residues adjacent. Throughout this section $n \geq 3$ is odd.

Lemma 3 (Rigidity)**.** *Every homomorphism $f : C_n \to C_n$ has the form $f(x) = r + \varepsilon x$, where $r \in \mathbb{Z}/n\mathbb{Z}$ and $\varepsilon \in \{1, -1\}$. In particular, fixing the ordered edge $(0, 1)$ fixes every vertex.*

Proof. Choose the integer increment $+1$ or -1 for each of the n consecutive steps of the image walk. Their sum is a multiple of n, lies in $[-n, n]$, and is odd. It is therefore n or $-n$, forcing all increments to have the same sign. □

Lemma 4 (Inequality link)**.** *A path of length $n - 2$ has a homomorphism to C_n with endpoint images x, y if and only if $x \neq y$.*

Proof. The possible integer displacements of a walk of that length are

$$-(n-2),\ -(n-4),\ \ldots,\ n-4,\ n-2.$$

These $n - 1$ integers give distinct residues modulo n: a difference is $2j$ with $0 < |j| < n$, and n is odd. None is zero modulo n, since each is odd and has absolute value less than n. Thus they give exactly the nonzero residues, and every listed displacement is realised by an appropriate sequence of forward and backward steps. □

A *link* below means a path of length $n - 2$ whose internal vertices are private. Different links may share endpoints, but no internal vertices.

3 The reduction

An instance of List Colouring consists of a graph F and lists $L(v) \subseteq \{0, \ldots, q - 1\}$. It asks for a map c with $c(v) \in L(v)$ for every vertex and $c(u) \neq c(v)$ for every edge uv. Relabelling the palette is harmless. Choose the least odd integer $n \geq \max\{5, q+1, |V(F)|\}$, and regard the lists as subsets of $\mathbb{Z}/n\mathbb{Z}$.

Construct G as follows.

1. Introduce a vertex h_v for every $v \in V(F)$ and two further vertices A, B, with edge AB.
2. For every pair $(v, d) \in V(F) \times \mathbb{Z}/n\mathbb{Z}$, introduce a copy $R_{v,d}$ of C_n, identifying its coordinates $0, 1$ with A, B, respectively. All other vertices of these cycles are private.

3. If $d \notin L(v)$, add a link from coordinate d of $R_{v,d}$ to h_v. This is the only link belonging to that reference copy.

4. For every edge $uv \in E(F)$, add one link from h_u to h_v.

There are no other edges. In particular, original edges of F become links, not edges of G. Copies $R_{v,d}$ with $d \in L(v)$ have no domain link; retaining these harmless copies makes the finite vertex type in Lean uniform.

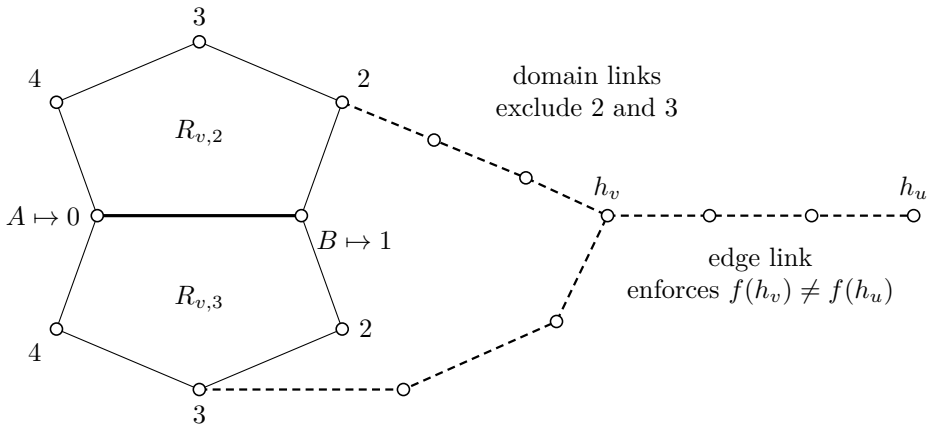


Figure 1: Part of the construction for target C_5. Solid edges form two reference cycles sharing AB. After normalising A, B to $0, 1$, their private vertices have the indicated images. Each dashed link has three edges and two private internal vertices; it forces its endpoint images to differ. The remaining reference copies and domain links are omitted.

In Figure 1, take $L(v) = \{0, 1\}$ and $L(u) = \{1, 2\}$. The link from coordinate 2 allows image 0 at h_v via the walk $2, 3, 4, 0$. It excludes image 2 because C_5 has no closed walk of length three. The other domain links enforce the remaining exclusions, and the link between the hubs requires different colours.

Lemma 5. *The instance (F, L) is list-colourable if and only if $G \to C_n$.*

Proof. Given a homomorphism, compose it with a target automorphism to place A, B at $0, 1$. By Lemma 3, every reference cycle then maps coordinatewise to C_n. Lemma 4 says that the domain links exclude exactly the residues outside $L(v)$ from the image of h_v. The links corresponding to edges of F force adjacent vertices to have different images. Thus $v \mapsto f(h_v)$ is a list colouring. Conversely, given a list colouring, map each h_v to its colour and every reference cycle coordinatewise. Each link has distinct endpoint images, so Lemma 4 extends the map over it. Private interiors make these extensions independent. □

3.1 Treewidth

Deleting A, B from G exposes a subdivision of F, trees attached at single hubs, and separate tree components. Subdivision preserves treewidth, and adding these trees raises it to at most $\max\{\mathrm{tw}(F), 1\}$, giving $\mathrm{tw}(G - \{A, B\}) \leq \max\{\mathrm{tw}(F), 1\}$. Adding the two anchors yields $\mathrm{tw}(G) \leq \max\{\mathrm{tw}(F)+2, 3\}$. We give an explicit certificate construction so the same bound holds for supplied decompositions within the claimed resources.

Let J contain the reference cycles, the hubs with the original edges of F, and a direct edge from coordinate d of $R_{v,d}$ to h_v for every pair (v, d), including allowed colours. Root the private

reference path $2, \ldots, n-1$ at coordinate d when $d \geq 2$, and at coordinate 2 otherwise. Attach this root to h_v, allowing an extra edge when needed. The resulting graph of hubs and private paths contains $J - \{A, B\}$ and is obtained from F by attaching rooted trees at single vertices.

Keep the input bags. For each new vertex x, add the bag $\{x, \mathrm{parent}(x)\}$ below its parent's bag, attaching a root bag to an input bag containing its hub. These are bags of size at most two. Add A, B to every bag. The resulting decomposition of J has width at most $\max\{w+2, 3\}$: old bags have size at most $w+3$, and new bags have size at most four.

Delete unused direct domain edges and replace each required domain or original edge ab by its link. Subdivision uses bags of size three: for internal vertices $x_1, \ldots, x_t$, attach the chain $\{b, a, x_1\}, \{b, x_1, x_2\}, \ldots, \{b, x_{t-1}, x_t\}$ at a bag containing a, b. Vertex occurrences stay connected and all edges are covered. The width remains at most $\max\{w+2, 3\}$.

3.2 Pathwidth

Use the same auxiliary graph J as in Section 3.1. Its direct domain edges for allowed colours can only help provide a decomposition for the eventual source.

Given a path decomposition of F of width p, add A, B to every bag. At the first bag containing h_v, process all its reference cycles serially, retaining that enlarged bag. Traverse each reference cycle using a window of two consecutive private vertices. Every direct domain edge is covered when its reference vertex is in the window, since h_v is retained. This gives a path decomposition of J of width at most $p+4$.

When the endpoints of an edge share a bag, subdividing that edge into an arbitrarily long path raises pathwidth by at most two: retain that bag and sweep the new path, with at most two consecutive internal vertices present at once. Process all subdivisions serially at their owning bags. Delete the unused direct domain edges and subdivide the required domain and original edges into links. Vertex intervals and edge coverage are preserved, giving $\mathrm{pw}(G) \leq p+6$. This conservative bound is the one formalised in Lean.

3.3 Feedback vertex sets

Let S be a feedback vertex set of F. Delete

$$S' = \{h_v : v \in S\} \cup \{A, B\}$$

from G. The original-edge links form a subdivision of $F - S$, together with pendant or detached path fragments from edges incident with S. This is a forest. Every reference cycle minus A, B is a path. Its single optional domain link attaches it to at most one surviving hub, or leaves pendant or detached fragments when an endpoint was deleted. Such attachments cannot create a cycle. Thus S' is a feedback vertex set and $|S'| = |S| + 2$.

3.4 Exact parameter transfer

Proposition 6. *For the padded construction, $K_2 \vee F$ is a minor of G, where the join adds two adjacent vertices adjacent to every vertex of F. If F contains an edge, then*

$$\mathrm{tw}(G) = \mathrm{tw}(F) + 2, \qquad \mathrm{fvs}(G) = \mathrm{fvs}(F) + 2.$$

For every nonempty F, also $\mathrm{pw}(F) + 2 \leq \mathrm{pw}(G) \leq \mathrm{pw}(F) + 6$.

Proof. Since $q < n$, residue $n-1$ is forbidden at every hub. For each v, take the copy $R_{v,n-1}$ and contract its entire private path and domain link into h_v. Its edges to coordinates 0 and 1 become h_vA and h_vB. These private paths and link interiors are disjoint for distinct hubs. Retain AB, contract the original-edge links to edges, and discard the unused pieces. This produces $K_2 \vee F$ without identifying hubs.

For nonempty F, the standard join identity gives $\mathrm{tw}(K_2 \vee F) = \mathrm{tw}(F) + 2$. Every chordal completion of the join restricts to one of F, and every chordal completion of F extends to the join; adjoining the two universal vertices increases the maximum clique size by two. Minor monotonicity supplies the lower bound, and the upper bound in Section 3.1 matches it when F has an edge.

Write $k = \mathrm{fvs}(F)$, $c = \mathrm{vc}(F)$ for its vertex cover number, and $m = |V(F)|$. Deleting both, exactly one, or neither of the two new vertices gives, respectively,

$$\mathrm{fvs}(K_2 \vee F) = \min\{k+2, c+1, m\}.$$

Indeed, with one new vertex surviving, the remaining vertices of F must be independent; with both surviving, no vertex of F may survive. When F has an edge, $k \le c-1$: retain one vertex of a minimum vertex cover, leaving a star and isolated vertices after deleting the others. Also $k \le m-2$, by retaining the endpoints of an edge. Thus the displayed minimum is $k+2$. Feedback vertex number is minor-monotone, so the existing upper bound $\mathrm{fvs}(G) \le k+2$ is exact.

Finally, $\mathrm{pw}(K_2 \vee F) = \mathrm{pw}(F) + 2$ for nonempty F. Indeed, restricting any path decomposition of the join to F gives a bag with at least $\mathrm{pw}(F)+1$ input vertices. Their occurrence intervals and those of the two universal vertices pairwise intersect. By the Helly property for intervals, some bag contains all these vertices, giving the lower bound. Adding the universal vertices to every input bag gives equality. Minor monotonicity and Section 3.2 now give the bounds for G. □

The minor argument uses a common missing colour, supplied by palette padding. The upper bounds also cover edgeless and empty inputs.

3.5 Size and construction resources

Write $m = |V(F)|$, $e = |E(F)|$, and $f = \sum_v (n - |L(v)|)$. The exact source size is

$$|V(G)| = m + 2 + mn(n-2) + (e+f)(n-3) \le 2 + m + mn^2 + (mn+m^2)n. \tag{1}$$

Since $n = O(q+m+1)$, the output size is polynomial in the explicit input size. Here the palette may be taken to contain only colours occurring in the lists: order the distinct labels by first occurrence in the input and replace each by its rank. Repeated scans compute these ranks in logarithmic space, so q is bounded by the input length even if the original labels are large binary integers.

Vertices and bags are named by a constant number of polynomially bounded indices. Each gadget is processed at the first input bag covering its attachment, found by rescanning the input. In the path construction the intermediate bag stream can likewise be regenerated on demand. Appendix A gives the enumeration and its bag counts. The transducer uses logarithmic workspace and polynomial time, completing the proof of Theorem 2.

3.6 Sparse sources and large girth

Proposition 7. *The source G is 2-degenerate. If F is nonempty, its girth is exactly n.*

Proof. Repeatedly remove internal vertices of links, all of which have degree at most two in the remaining graph. Then remove the private reference vertices, again of degree at most two. Only AB and isolated hubs remain. This gives a degeneracy ordering.

Before adding links, the nontrivial component consists of n-cycles sharing only the edge AB; its shortest cycle has length n. Every link joins different components of this base graph, so a cycle using a link must use at least two links. Their private interiors force it to traverse each link in full, giving length at least $2(n-2) \geq n$. A reference cycle of length n exists whenever F has a vertex. □

The choice $m \leq n$ gives a quantitative girth bound. By (1), $|V(G)| \leq 2+n+n^3+2n^3 \leq 4n^3$ for $n \geq 5$. Thus for every nonempty input the girth is at least $(|V(G)|/4)^{1/3}$. To impose any additional fixed lower bound on the girth, include it in the maximum defining n. The target length remains polynomial in the input size, and all width and feedback bounds continue to hold.

The output is also connected. Since $n \geq q+1$, residue q is outside every input list. Each hub therefore has a domain link to the reference component containing AB. Every remaining vertex lies in a reference copy or a link. For empty F, the output is simply AB. This explains the strict palette padding in the choice of n.

3.7 Amplifying the girth by odd subdivision

Write $S_t(G)$ for the graph obtained by replacing each edge of G by a path of t edges with private internal vertices. The following equivalence also follows from the circular-colouring theory of odd subdivisions [14]. We give an explicit projection because the reduction changes both the source and the target.

Lemma 8. *For odd $n \geq 3$ and positive odd t,*

$$S_t(G) \to C_{tn} \quad \Longleftrightarrow \quad G \to C_n.$$

Proof. Given $G \to C_n$, send an original vertex of colour i to ti in C_{tn}. Consecutive colours in C_n give endpoints of a t-edge arc in C_{tn}. Map each replacement path along this arc.

Conversely, define a map from the residues modulo tn to those modulo n by writing an integer lift as $qt+r$, $0 \leq r < t$, and putting

$$\pi(qt+r) = q + (r \bmod 2) \pmod{n}.$$

Changing the lift by tn changes q by n. The endpoints of a t-step walk have lifts at odd distance $d \leq t$. Order these lifts increasingly. If they are in the same block of length t, their remainders have opposite parities and their images differ by one. If they lie in successive blocks, their remainders differ by $d-t$, which is even. Now the block indices differ by one, so again the images are adjacent. Applying π to the images of the original vertices of G proves the reverse implication. □

Proposition 9. *For every fixed $0 < \varepsilon < 1$, the reduction on nonempty inputs can output a connected, 2-degenerate graph of order N and girth $\Omega(N^{1-\varepsilon})$. Its shortest cycle has odd length equal to the new target length. Subdivision preserves the feedback vertex number and treewidth of the reduction graph and adds at most two to the width of its supplied path decomposition.*

Proof. Let $M = |V(G)|$ before subdivision. Fix an integer $a \geq 1$ with $a/(a+1) \geq 1-\varepsilon$, and set $t = 2M^a + 1$. Each cycle traverses its replacement paths in full, so subdivision multiplies every cycle length by t. Proposition 7 thus gives girth tn, witnessed by an odd cycle. A 2-degenerate graph has at most $2M$ edges, whence

$$N = M + (t-1)|E(G)| \leq 2tM = O(M^{a+1}), \qquad tn \geq t = \Omega(N^{a/(a+1)}).$$

The new graph is connected. Remove the new degree-two vertices first, then the original vertices, which are now isolated. This gives a 2-degeneracy ordering.

Every original feedback vertex set still hits every cycle. Conversely, replace each selected internal vertex by an endpoint of its original edge. Every original cycle lifts to a cycle and is hit by the resulting set. Contracting the replacement paths recovers G as a minor. For the treewidth upper bound, attach a chain of bags of size three for each subdivided edge as in Section 3.1. Here G contains a cycle and hence has treewidth at least two. For pathwidth, retain the first bag containing the two endpoints and sweep the replacement paths assigned to that bag one at a time, keeping at most two internal vertices at once. For fixed a, counters for edges, bags and positions along a path all use logarithmic space, and the output has polynomial length. Lemma 8 transfers the decision equivalence. □

4 Complexity consequences

XNLP consists of problems decidable nondeterministically in time $f(k)N^{O(1)}$ and space $f(k)\log N$, where N is the input length. XALP permits, in addition, an auxiliary stack with access only to its top; it has an equivalent characterisation by alternation with bounded total computation-tree size [4, 3]. We use parameterised logspace reductions, with workspace $f(k) + O(\log N)$.

4.1 List Colouring with a feedback vertex set

Theorem 10. *List Colouring is W[SAT]-hard parameterised by feedback vertex number, even when the input includes a set of k vertices whose deletion leaves a tree and the parameter is k.*

Proof. Weighted Antimonotone Satisfiability asks whether an AND/OR formula with leaves $\neg x$, for x in a set X of variables, is satisfied by setting exactly k variables true. It is W[SAT]-complete [1, Theorem 4.1(ii)]. Use its formula tree, with a separate vertex for every literal occurrence, including repeated occurrences of the same variable. For $k > |X|$, output one vertex with an empty list and an empty deletion set. Henceforth assume $k \leq |X|$.

Take the tree edges and add a clique S of k selectors, each with list X. Join every literal vertex to every selector. A colouring of the clique chooses a set $U \subseteq X$ of exactly k true variables. Assign the following *active* colours to the tree vertices.

- an AND gate u has one colour a_u;
- an OR gate u has a colour $a_{u,v}$ for each child v;
- a leaf $\neg x$ has colour x.

All gate colours are distinct and disjoint from X. The list at the root contains only its active colours. At every other vertex v add one *inactive* colour, namely a_u when its parent u is an AND gate and $a_{u,v}$ when its parent is an OR gate.

Fix the selector colours. At a leaf $\neg x$, the active choice is available precisely when $x \notin U$. On a tree edge uv from parent to child, the only colour shared by the two lists is the child's inactive colour. Hence an active AND gate forces every child to be active, and an OR gate coloured $a_{u,v}$ forces child v to be active. Induction from the leaves shows that every active vertex represents a true subformula. The root must be active, including when the formula consists of one literal.

Conversely, start at the root of a true formula. At a required AND gate use its active colour and require every child. At a required OR gate choose a true child, use its corresponding colour, and require that child alone. Colour each required leaf $\neg x$ with x and all unrequired vertices with their

inactive colours. The required leaves avoid all selector colours; on each tree edge the parent uses the child's inactive colour exactly when it requires that child. This gives a proper list colouring.

Deleting the selectors leaves the original formula tree. For a tree of t nodes with ℓ leaves, the graph has $t + k$ vertices and $t - 1 + \binom{k}{2} + k\ell$ edges. Gate colours can be named by their node and child indices. Given the formula as a node table with parent pointers, counters and input scans enumerate the lists and edges in logarithmic space. The supplied deletion set has size k, so the minimum feedback vertex number is also bounded by the input weight. □

Transferring hardness to cycles. Apply Theorem 2 to the two published width lower bounds and to Theorem 10. The transformed decompositions have the claimed widths. The k selector hubs and the two anchors give a supplied feedback set of size $k + 2$. We may adjoin a single isolated vertex with a singleton list to ensure a nonempty input, extending a supplied decomposition by a singleton bag. Proposition 9 gives the asserted girth restriction with the same feedback set and at most two more vertices per path bag. The resulting graph has a shortest cycle of odd length equal to the target length. By the equivalence in the introduction, discarding the target proves the same hardness for core recognition. These graphs have chromatic number three, by 2-degeneracy and the presence of an odd cycle.

4.2 Membership

We first bound the target length so that binary encoding does not increase the resources needed by the algorithms.

Lemma 11 (Capping the target length)**.** *Let $m = |V(G)|$ and let $n \geq 5$ be odd. If $n > m$, then $G \to C_n$ if and only if G is bipartite. Setting $n' = \min\{n, 2m+5\}$ preserves whether $G \to C_n$. The binary encoding of n can be transformed to the unary encoding of n' in logarithmic space, leaving G unchanged.*

Proof. An odd closed walk of length t in C_n has an odd integer displacement that is a multiple of n, so $t \geq n$. A nonbipartite graph on m vertices contains an odd cycle of length at most m, precluding a homomorphism to C_n when $n > m$. Conversely, a bipartition maps to any edge of C_n.

If the cap changes the target, both n and n' exceed m, so the preceding equivalence applies to both. The cap is odd and at least five. Scan the binary digits of n with an accumulator saturated at $2m + 5$; then emit its value in unary. The counter uses $O(\log(m + 2))$ bits and the output has $O(m + 1)$ symbols. Input validity, including oddness, can also be checked by scans. No source parameter or supplied decomposition changes. □

By Lemma 11, first replace a binary target length by its linear-size cap. In the membership arguments below, n denotes this capped length, so target vertices have $O(\log N)$-bit identifiers and all loops over the target take polynomial time.

We first isolate the common correctness argument. A bag assignment is a map $a_t : B_t \to V(H)$ that preserves every graph edge with both endpoints in B_t.

Lemma 12 (Gluing bag assignments)**.** *For a tree decomposition $(T, (B_t))$ of G, a homomorphism $G \to H$ exists if and only if there are bag assignments that agree on $B_s \cap B_t$ for every edge st of T.*

Proof. A homomorphism restricts to such assignments. Conversely, if a vertex v occurs in two bags, every bag on the unique path between them also contains v. Agreement across adjacent bags therefore gives a single image for v, independent of the chosen occurrence. Vertex coverage defines this map on all of G, and edge coverage makes it a homomorphism. □

Supplied width. On a path, guess one bag assignment at a time and check its internal edges and agreement with the preceding bag. On a rooted tree, visit children in depth-first order. Save the parent assignment, bag index and next-child index on the auxiliary stack before descending, then restore them on return. Every sibling thus sees the same parent assignment. Lemma 12 proves correctness in both cases. At most two assignments occupy the work tape, using $O((w+1)\log N)$ bits for width w and input length N. Each bag and tree edge requires only polynomially many input scans. This gives XNLP membership on paths and XALP membership on trees. Appendix A gives the preliminary logspace certificate check.

Feedback vertex number. A parameterised problem is in W[P] if a nondeterministic Turing machine decides it in time $f(k)N^{O(1)}$ with at most $f(k)\log N$ nondeterministic steps [12, Chapter 3]. Let $k = \mathrm{fvs}(G)$. Compute a minimum feedback vertex set S deterministically in time $f(k)N^{O(1)}$ [8]. Guess an image in $\mathbb{Z}/n\mathbb{Z}$ for every vertex of S, using $k\lceil\log_2 n\rceil$ nondeterministic bits, and reject unless the guess preserves the edges inside S. The graph $G - S$ is a forest; root each of its trees. For a vertex x of $G - S$ and a residue a, call (x, a) *feasible* if a is adjacent in C_n to the guessed image of every neighbour of x in S, and every child y of x has a feasible pair (y, b) with b adjacent to a. Feasibility is computed from the leaves upwards in polynomial time, and the guess extends to a homomorphism exactly when every root has a feasible pair. This proves membership in W[P].

This membership result also follows from the Binary CSP upper bound [5, Theorem 2], since the standard encoding preserves the constraint graph. For both List Colouring and the input-cycle problem, the remaining feedback-number gap is between W[SAT]-hardness and W[P]-membership.

Core recognition. For the width algorithms, guess an odd $3 \le \ell \le |V(G)|$ and verify $C_\ell \to G$ by guessing a closed walk of length ℓ in logarithmic space. Then check $G \to C_\ell$ with the corresponding bag algorithm. For W[P], reject bipartite graphs and compute the odd girth deterministically in polynomial time before guessing the feedback-set images. The odd girth is the minimum distance between the two copies of any one vertex in the bipartite double cover, since this is the minimum length of an odd closed walk. This completes the proof of Theorem 1.

Remark 13 (Fixed target). For fixed odd $q \ge 5$, enumerating all assignments on a supplied feedback set of size s takes $O^*(q^s)$ time, where O^* suppresses polynomial factors. Conversely, an $O^*(b^s)$ algorithm for C_q yields the same bound for List Colouring on q colours. At $n = q$ the decision equivalence holds for arbitrary lists, and the supplied set grows by two, costing a factor b^2 for fixed $b > 1$. This transfer is unconditional.

Remark 14 (Forest targets). If a target forest has an edge, its sources are exactly the bipartite graphs. A nonempty edgeless target accepts precisely edgeless sources, and an empty target accepts only the empty source. Thus at target treewidth one the problem is polynomial-time decidable.

5 Circular colouring by feedback edge number

For $p \ge 2q \ge 2$, the circular clique $K_{p/q}$ has vertex set $\mathbb{Z}/p\mathbb{Z}$. Two colours are adjacent when their difference lies in the integer interval $[q, p-q]$ modulo p. Thus $K_{(2k+1)/k}$ is isomorphic to C_{2k+1}, since its two steps are $\pm k$ and k is coprime to $2k+1$. Write $N = |V(G)|$ and $r = |E(G)| - N + c$, where c is the number of connected components. This *feedback edge number* counts the edges outside a spanning forest.

Theorem 15. *For positive p, q given in binary with $p \ge 2q$, deciding $G \to K_{p/q}$ takes $2^{O(r)}|I|^{O(1)}$ time and polynomial space, where $|I|$ is the input length. The exact circular chromatic number of G can be computed within the same bounds.*

Proof. On an oriented chain of L edges, every step contributes an arbitrary integer in $[q, p-q]$. The possible sums are all integers in

$$[Lq, L(p-q)].$$

Indeed, start each step at q and distribute any desired excess of at most $L(p-2q)$ among the steps. Reduction modulo p gives exactly the feasible endpoint differences. In particular a component that is a cycle of length L is feasible if and only if this interval contains a multiple of p.

Repeatedly remove vertices of degree at most one. Any colouring extends back across these deletions, since every target colour has a neighbour. Test pure cycle components as above. In each other remaining component, suppress the maximal degree-two chains, retaining their lengths and any resulting loops or parallel edges. The multigraph has minimum degree three and its original feedback edge number r_i. With b vertices and e edges it satisfies $3b \le 2e$ and $e = b + r_i - 1$, hence $e \le 3(r_i - 1)$. Altogether there are at most $3r$ chains.

For a chain directed from u to v, put $w = L(p-2q)$ and let $a \in [0, p-1]$ represent $Lq \bmod p$. If $w \ge p-1$ every residue is possible. Otherwise, using representatives $0 \le x_v < p$, the permitted integer differences $x_v - x_u$ form the nonempty intersections

$$[1-p, p-1] \ \cap \ [a+jp, a+w+jp], \qquad j \in \{-2, -1, 0\}.$$

For a loop the difference is zero. Enumerate one interval per constrained chain. Each chosen interval $[A, B]$ gives two difference constraints $x_v - x_u \le B$ and $x_u - x_v \le -A$. Express the coordinate bounds relative to a common variable, which may be translated to zero. A branch is feasible exactly when the constraint digraph has no negative cycle. In that case shortest distances from a new source of zero-weight arcs give integral coordinates. The chain criterion extends the endpoint colours independently across their private interiors. Conversely any colouring selects a feasible branch. The 3^{3r} branches can be processed sequentially. All weights, including $L(p-q)$, have polynomial bit length.

For a graph with an edge, $\chi_c(G)$ is the least feasible ratio p/q. Vince's finite-attainment theorem gives a minimiser with $p \le N$ for connected graphs [20, Theorem 3]. For a disjoint union take the maximum over components, since circular colourability is monotone in the ratio [21]. Test the $O(N^2)$ pairs with $1 \le q \le p/2$ and $p \le N$ and return the least feasible ratio by integer comparison. We use the usual chromatic conventions 1 for a nonempty edgeless graph and 0 for the empty graph. □

The theorem includes homomorphism to an input odd cycle. For recognition, reject bipartite graphs and apply the threshold test at $p = h,\ q = (h-1)/2$, where h is the odd girth.

Corollary 16. *Computing $\chi_c(G)$ and deciding both problems of Theorem 1 are fixed-parameter tractable for feedback vertex number plus maximum degree.*

Proof. Deleting all edges incident with a minimum feedback vertex set leaves a forest. Thus $r \le \Delta(G)\,\mathrm{fvs}(G)$, and Theorem 15 needs only the graph as input. □

Corollary 17. *On graphs of order N and finite girth g, computing $\chi_c(G)$ and solving both problems of Theorem 1 take $2^{O((N/g)^2)}|I|^{O(1)}$ time. The running time is polynomial whenever $g = \Omega(N \log\log N / \log N)$.*

Proof. Write $D = N/g$. We first bound r for every D. Discard tree components, which contribute zero to r. Put $s = \lfloor (g-3)/8 \rfloor$. In each remaining component take a maximal set of centres pairwise at distance greater than $2s$. The radius-s balls about all centres are disjoint. Each contains at least

$s+1$ vertices, since the component is connected and has at least g vertices. Thus the total number of centres is at most $N/(s+1) = O(N/g)$.

Maximality puts every vertex within distance $2s$ of a centre. Simultaneous breadth-first search from the centres partitions the graph into rooted trees of radius at most $2s$. Contract the tree edges. An extra edge internal to a tree would close a cycle of length at most $4s+1$. Two edges joining the same pair of trees would, with the paths in the two trees, form a cycle of length at most $8s+2$. Both lengths are less than g. Hence the contraction gives a simple graph on $O(N/g)$ vertices. Each tree-edge contraction preserves r, so $r = O(D^2)$. Theorem 15 gives the first time bound.

To obtain the polynomial range, suppose $D = O(\log N/\log\log N)$. Then $r = O(D^2) = O(\log^2 N)$. Connect the original components by bridges, giving a simple graph of the same girth with $N + r - 1$ edges. For $r \geq 3$, the girth bound of Bollobás and Szemerédi [10] gives $g = O((N+r)\log r/r)$. Since $N + r = O(N)$, we obtain $r = O(D\log r) = O(\log N)$. Theorem 15 now takes polynomial time, as it also does for $r < 3$. □

The girth bound counts all cycles. For core recognition a companion algorithm [16] gives polynomial time already at *odd* girth $\Omega(N/\log N)$, by using the odd girth as the target length. For an independently supplied target, its length can be much smaller than the source girth. The case of girth $\Omega(N/\log N)$ and unrestricted odd target length remains open in these bounds.

A Encoding and checking decompositions

Emitting the graph. Let N denote the length of the explicitly encoded input, including any supplied decomposition. Give output vertices identifiers consisting of a type tag and a constant number of indices: an original vertex, a residue, a reference position, or an original edge and a link position. All indices are polynomially bounded in N, so an identifier occupies $O(\log N)$ bits. Nested counters enumerate the vertices and edges. List membership and original adjacency are decided by rescanning the input; original edges are emitted once, in increasing endpoint order. If consecutive integer identifiers are required, the rank of a tuple can be recomputed by enumerating its predecessors. Output n in binary; the corresponding cycle can also be enumerated explicitly. Neither an adjacency table nor a list of output vertices is stored.

Emitting a tree decomposition. Fix an order on the input bags. A reference gadget belongs to the first bag containing v; an original-edge link belongs to the first bag containing both endpoints. Such an owning bag exists by vertex or edge coverage and can be found by scans.

The reference-path root has coordinate $r = \max\{2, d\}$. Its bag has parent the owning input bag; the bag at any other private coordinate j has parent $j+1$ if $j < r$, and $j-1$ if $j > r$, within the same copy. Its vertex parent follows the same rule, except that the root's vertex parent is h_v. Emit the input bags with A, B added, and the reference bags $\{A, B, x, \text{parent}(x)\}$. These formulas require only coordinate comparisons and give the decomposition of J directly.

A required direct domain edge belongs to its copy's root bag when $d \geq 2$, and to its owning input bag when $d = 0$ or $d = 1$. An original edge belongs to its owning input bag. Emit each subdivision chain of size-three bags from Section 3.1 at that bag. Parents within a chain are consecutive indices, so both bags and decomposition edges can be emitted without a general forest traversal.

Emitting a path decomposition. Implement the two stages of Section 3.2 in order. For each input bag, emit its enlargement by A, B, followed by the reference windows for vertices whose first

occurrence is that bag. Order these windows by vertex, residue, and reference position. This is an explicit stream of bags for J. For each required link, find the first bag of this stream containing its endpoints by regenerating the stream. At that bag insert the link's consecutive internal-vertex windows, retaining the whole bag of J. Order links with the same owning bag by their identifiers. The intermediate stream need not be stored: its bags and their contents can be recomputed by counters and input scans.

If the input decomposition has b bags, either construction uses $O(b + mn^2 + (e + f)n)$ output bags. Every bag is emitted one vertex at a time, and each owning-bag query stores only indices and membership flags. The enumerations have a fixed number of nested loops over polynomially bounded ranges. Consequently the transducer uses $O(\log N)$ workspace, polynomial time, and polynomial output length, independently of the input width.

Checking the certificate. The algorithms do not assume valid decompositions as a promise. We encode a tree by a designated root and one parent pointer for every other bag. Reject out-of-range pointers and require every bag to reach the root within b steps, where b is the number of bags. This checks that the index graph is a tree using counters and rescans.

For each graph vertex v, count its occurrence bags, say a_v, and the tree edges with both endpoints among those bags, say e_v. The induced index subgraph is a forest, so it is nonempty and connected exactly when $a_v > 0$ and $e_v = a_v - 1$. Thus these counts check vertex coverage and connected occurrences. Bag entries, duplicate entries, bag sizes, and coverage of each graph edge are checked by scans. For a path, check that every occurrence set is a nonempty interval. These preliminary checks take logarithmic workspace and polynomial time. No decomposition-finding algorithm is assumed.

B Lean verification

The artifact uses Lean 4.35.0 [11] and mathlib v4.35.0 [17]. It checks the weighted-formula and List Colouring decision reductions, the supplied decomposition and feedback-set transfers, their numerical bounds and matching lower bounds, and the connection with odd-cycle cores. It also checks the original source girth and size bounds, the endpoint projection for odd subdivision, and the walk interval for an odd-cycle target. The circular-clique generalisation, graph-level girth amplification, encodings and running times are proved in the manuscript.

To reproduce the audit at revision `848545a`, run `python3 scripts/check_formalization.py` from the repository root [9]. It builds the library, checks for proof admissions and audits the principal declarations. `FORMALIZATION.md` indexes the statements and the README gives commands for the finite checks. Mathematical development, formalisation and drafting used AI assistance under human direction.

References

[1] K. A. Abrahamson, R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic* 73(3):235–276, 1995. doi:10.1016/0168-0072(94)00034-Z.

[2] Antoine Amarilli. Complexity of the homomorphism problem parameterized by treewidth. *Theoretical Computer Science Stack Exchange*, question 34877, June 2, 2016, with a comment on workshop reductions, May 30, 2025. `https://cstheory.stackexchange.com/q/34877`.

[3] Hans L. Bodlaender, Carla Groenland, Hugo Jacob, Marcin Pilipczuk and Michał Pilipczuk. On the complexity of problems on tree-structured graphs. In *17th International Symposium on Parameterized and Exact Computation (IPEC 2022)*, LIPIcs, vol. 249, pp. 6:1–6:17. Schloss Dagstuhl, 2022. doi:10.4230/LIPIcs.IPEC.2022.6.

[4] Hans L. Bodlaender, Carla Groenland, Jesper Nederlof and Céline Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. *Information and Computation* 300:105195, 2024. doi:10.1016/j.ic.2024.105195.

[5] Hans L. Bodlaender, Carla Groenland and Michał Pilipczuk. Parameterized complexity of Binary CSP: vertex cover, treedepth, and related parameters. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, LIPIcs, vol. 261, pp. 27:1–27:20. Schloss Dagstuhl, 2023. doi:10.4230/LIPIcs.ICALP.2023.27.

[6] Hans L. Bodlaender, Carla Groenland and Michał Pilipczuk. Parameterized complexity of Binary CSP: vertex cover, treedepth, and related parameters. Full version, arXiv:2208.12543v5, 2023.

[7] Clément Carbonnel. Faster algorithms for graph homomorphism via tractable constraint satisfaction. arXiv:2602.23000v2, 2026.

[8] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

[9] Jason Davies. Odd-cycle homomorphism: Lean formalisation and executable checks. Software repository, 2026. Version `848545a`. `https://github.com/jasondavies/odd-cycle-homomorphism`.

[10] Béla Bollobás and Endre Szemerédi. Girth of sparse graphs. *Journal of Graph Theory* 39(3):194–200, 2002. doi:10.1002/jgt.10023.

[11] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, LNCS, vol. 12699, pp. 625–635. Springer, 2021. doi:10.1007/978-3-030-79876-5_37.

[12] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer, 2006. doi:10.1007/3-540-29953-X.

[13] A. M. H. Gerards. Homomorphisms of graphs into odd cycles. *Journal of Graph Theory* 12(1):73–83, 1988. doi:10.1002/jgt.3190120108.

[14] Hossein Hajiabolhassan and Ali Taherkhani. On the circular chromatic number of graph powers. *Journal of Graph Theory* 75(1):48–58, 2014. doi:10.1002/jgt.21722.

[15] A. Kostochka, J. Nešetřil and P. Smolíková. Colorings and homomorphisms of degenerate and bounded degree graphs. *Discrete Mathematics* 233:257–276, 2001. doi:10.1016/S0012-365X(00)00245-4.

[16] Jason Davies. Homomorphisms to long odd cycles in time exponential in the order-to-length ratio. Companion manuscript, 2026. Distributed alongside this paper as `long-odd-cycles.pdf`.

[17] The mathlib Community. The Lean mathematical library. In *Certified Programs and Proofs (CPP 2020)*, pp. 367–381. ACM, 2020. doi:10.1145/3372885.3373824.

[18] Karolina Okrasa and Paweł Rzążewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM Journal on Computing* 50(2):487–508, 2021. doi:10.1137/20M1320146. Preprint version: arXiv:1906.08371v3, 2020.

[19] Marta Piecyk and Paweł Rzążewski. Fine-grained complexity of the list homomorphism problem: feedback vertex set and cutwidth. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, LIPIcs, vol. 187, pp. 56:1–56:17. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.STACS.2021.56.

[20] Andrew Vince. Star chromatic number. *Journal of Graph Theory* 12(4):551–559, 1988. doi:10.1002/jgt.3190120411.

[21] Xuding Zhu. Circular chromatic number: a survey. *Discrete Mathematics* 229(1–3):371–410, 2001. doi:10.1016/S0012-365X(00)00217-X.