Analysis of Walkability Less Than or Equal to 2 in Graphs

Author: **Noelia Falczuk and Demián Fraiman**

Group name: Perfect Matching

Date: December 10, 2025

Abstract

This project presents a novel approach to the graph recognition problem proposed in Investigathon 2025, which asks whether a given graph admits a 2-colored vertex ordering that avoids two specific forbidden patterns. We introduce the concept of path-likeness (caminitud) to characterize such graphs, proving that they are precisely those where every edge connects vertices that are either consecutive in the global ordering or consecutive within their own color class.

Our analysis reveals fundamental structural properties: all admissible graphs are planar, have maximum degree at most 4, and decompose into a linear chain of induced cycles forming an F_2-basis of the cycle space. Building on this characterization, we develop a cubic-time algorithm that not only decides recognition but also constructs valid colorings and orderings when they exist. The algorithm employs a cycle hypergraph representation and enforces strict connection constraints between cycles and attached trees.

This work provides both theoretical insights into pattern-avoiding graph classes and practical algorithmic solutions, demonstrating how structural graph theory can yield efficient recognition algorithms for complex combinatorial problems.

1 General Remarks

First, we realized that if a graph contains a subgraph that is not of walkability k, then the graph itself cannot be of walkability k, as seen below:

Proposition 1. *Let G be a graph. If $U \subset G$ is a subgraph that is not of walkability k, then G is not of walkability k either.*

Proof. For any ordering, the graph G does not avoid the pattern because if it did, by restricting the ordering to U, we would avoid it, leading to a contradiction. □

Another useful property is that we can separate the problem by connected components:

Proposition 2. *Let G be a graph and $k \in N$. Then:*

$$\textit{walkability}(G) \leq k \iff \textit{each connected component } U \subset G \textit{ satisfies walkability}(U) \leq k.$$

Proof. $\Rightarrow$) Given $U \subset G$ a connected component, it suffices to take the ordering restricted to the vertices of U.

$\Leftarrow$) Construct the ordering by concatenating the orderings of each connected component arbitrarily. □

Henceforth, without loss of generality, we can assume graphs are connected.

Our next step was to work with the definition of graphs of walkability k to obtain a more manageable characterization.

Let G be a graph, $\phi : V \to \{1, .., n\}$ an ordering of its vertices, and $c : V \to \{1, .., k\}$ a coloring with k colors. We denote $\phi_k : V_k \to \{1, ..n_k\}$ as the ordering restricted to color k, where $V_k = \{v \in V : c(v) = k\}$ are the vertices of color k and n_k is the total number of vertices of that color.

Theorem 1 (Characterization of graphs with walkability$(G) \leq k$)**.** *The following characterization holds:*

$$\textit{walkability}(G) \leq k \iff \exists \phi, c : (|\phi(v_1) - \phi(v_2)| = 1) \vee (c(v_1) = c(v_2) \wedge |\phi_{c(v_1)}(v_1) - \phi_{c(v_2)}(v_1)| = 1) \quad \forall v_1 v_2 \in$$

i.e., each vertex can only be connected to a vertex that is adjacent in the overall ordering or adjacent in the ordering restricted to its color.

Proof. A graph is not of walkability k $\iff$ for every ordering, one of the patterns is found $\iff$ for every ordering there exists an edge between two nodes that are not adjacent in the ordering nor in their own color (moreover, they do not satisfy it overall) $\iff$ there exists an edge that does not meet the conditions. □

This characterization allows us to generate any walkability 2 graph by representing one color below, the other above, and the ordering from left to right.

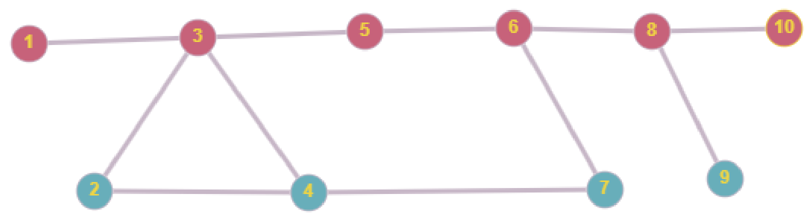

Figure 1: Representation of a walkability 2 graph.

Next, we observe a couple of properties easily deduced from this representation:

Proposition 3. *Let G be a graph satisfying walkability$(G) \leq 2$. Then:*

1. *G is planar,*
2. *$d(v) \leq 4 \quad \forall v \in V$,*
3. *the first and last vertices in the ordering have degree at most 2.*

Proof.

1. If G is not planar, for any ordering, there are always two edges that cross. Under our representation, we know that means the pattern exists.
2. A vertex can be connected at most to its two adjacent vertices of the same color and its two adjacent vertices in the overall ordering.
3. Being first or last, they have no previous or next vertices respectively, so they can only be connected to at most two other vertices.

□

Looking at examples, we noticed cycles are very relevant for walkability 2 graphs, as they can determine the ordering.

2 Cycles

Cycles are of walkability 2 and have several possible orderings, as seen in Figure 2. Therefore, they will be our first step in the construction.

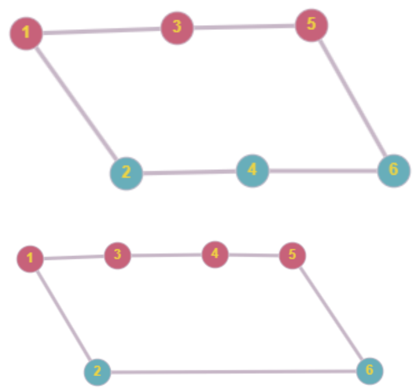

Figure 2: Two possible orderings of a six-vertex cycle.

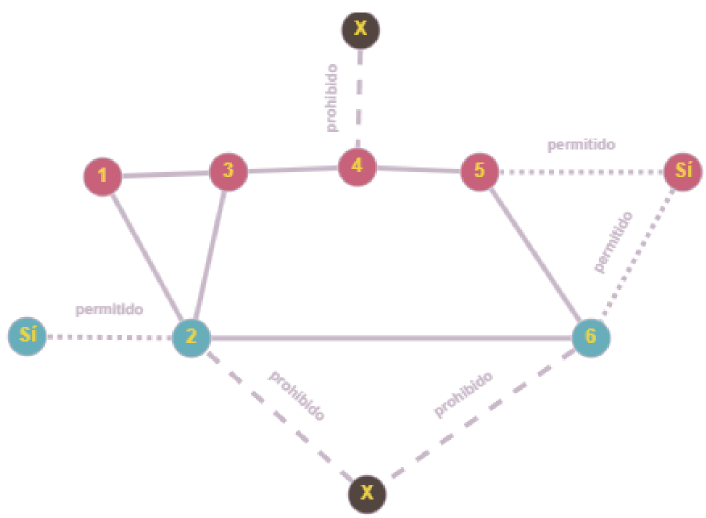

Figure 3: Allowed and forbidden configurations when two cycles share vertices.

In Figure 3, we see how having two cycles joined prohibits certain configurations. It seems the structure must remain linear.

Proposition 4. *Let G be a graph. If two cycles share more than one edge, then walkability(G) ≥ 3.*

Proof. If two cycles share more than one edge, for any possible ordering of one of the cycles, there will be a different path between two non-adjacent vertices of that cycle, so at some point an edge will cross the ordering. $\square$

Just as (connected) graphs of walkability 1 are paths, we realized that graphs of walkability 2 are also paths, but this time of two-dimensional objects, i.e., cycles. This perspective clarifies the problem. However, not all cycles are of interest; we are interested in the set of shortest cycles that can generate any other cycle. To make this rigorous, we need the following definition.

Definition 1 (Cycle Vector Space)**.** Given a graph G, define its cycle vector space $H_1(G, F_2)$ as the vector space where each coordinate is 1 if the edge is present and 0 otherwise, with addition over the field F_2, i.e., modulo 2.

Note that this addition is equivalent to taking the symmetric difference of cycles, which clearly yields another cycle.

Within this vector space, one could choose different bases, but let's see that induced cycles form a basis for our case of interest:

Proposition 5. *Let G be a graph with walkability$(G) \leq 2$. The set of induced cycles $\mathcal{U} \subset H_1(G, F_2)$ is a basis.*

Proof. Suppose it is not a basis. Then there exist two induced cycles whose sum yields another induced cycle. We know cycles cannot share more than one edge, otherwise the graph would not be of walkability 2, so they share exactly one edge. However, that means the cycle generated by the sum is not induced, because the induced subgraph contains that edge, leading to a contradiction. □

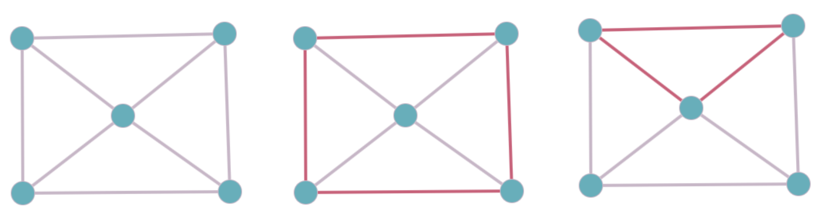

Figure 4: Example of generating cycles of $H_1(G, F_2)$

Figure 4 shows that, for example, if we consider all 3-cycles and a 4-cycle, they do not form a basis because adding all triangles yields the square. Moreover, the basis generated by the square and three triangles has more edges than the one generated by four triangles, which is minimal.

Furthermore, the induced cycle basis is minimal:

Proposition 6. *Let G be a graph with walkability$(G) \leq 2$. The induced cycle basis minimizes the sum of the lengths of its cycles over all possible bases.*

Proof. Let $\mathcal{G} \subset H_1(G, F_2)$ be a basis different from the induced one. Then it contains at least one non-induced cycle, meaning there exists an edge between two non-adjacent vertices in the cycle, forming two induced cycles each shorter than the original. Since bases have the same cardinality and this argument holds for any cycle in $\mathcal{G}$, the proof is complete. □

Moreover, there are at most $|V|$ induced cycles.

Proposition 7. *Let G be a graph with walkability(G) ≤ 2. Then the number of induced cycles is less than $|V|$.*

Proof. Using the representation, the maximum number of cycles occurs when all triangles share edges, as in Figure 5, where the property clearly holds.

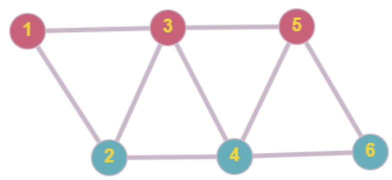

Figure 5: Maximum number of induced cycles.

$\square$

Also, note the converse: if the induced cycles form a basis, the graph is planar:

Lemma 1. *Let G be a graph. If the set of induced cycles is a basis of $H_1(G, F_2)$, then the graph is planar.*

Proof. By Kuratowski's theorem, it suffices to check that K_5 and $K_{3,3}$ do not have their induced cycles as a basis of their first homology. This is true because $\dim H_1(K_5, F_2) = \beta_1 = 10 - 5 + 1 = 6$ and the induced cycles of K_5 are all 3-subsets, i.e., $\binom{5}{3} = 10$; and $\dim H_1(K_{3,3}, F_2) = \beta_1 = 96 + 1 = 4$, while the number of induced cycles (4-cycles in $K_{3,3}$) is $\binom{3}{2}\binom{3}{2} = 9$. $\square$

Definition 2 (Cycle Hypergraph)**.** Given a graph G and a subset of cycles $\mathcal{U} \subset H_1(G, F_2)$, define the hypergraph $\mathcal{U}(G)$ as the graph whose vertices are the cycles in $\mathcal{U}$ and edges represent if two cycles satisfy any of the following:

- share an edge,
- share a hinge vertex,
- there exists a path between them that does not pass through any other cycle (excluding vertices belonging to either cycle).

See some examples:

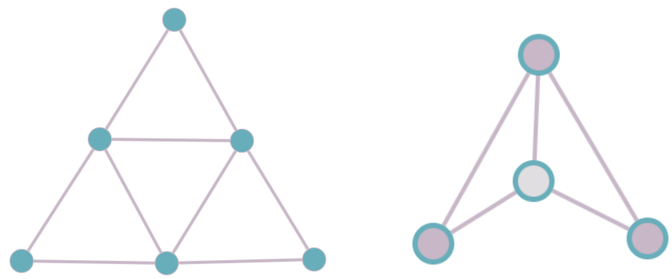

Figure 6: Pyramid graph and its hypergraph.

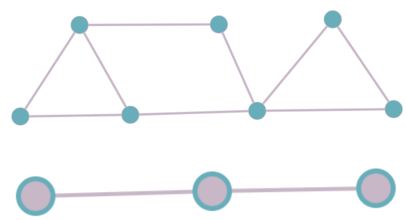

Figure 7: Joined cycles and their hypergraph.

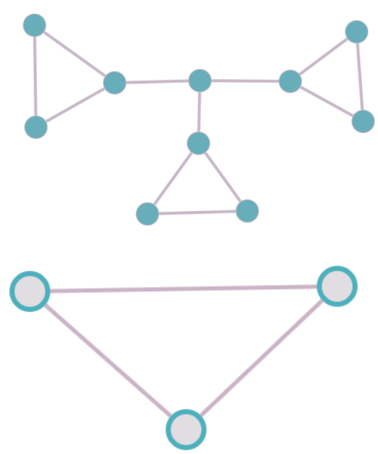

Figure 8: Union of triangles via paths and its hypergraph.

Lemma 2. *Let G be a graph with walkability$(G) \leq 2$ and let the hypergraph of its induced cycles $\mathcal{C}(G)$ be a path. Then in any valid ordering, the cycles must follow the path order either directly or inversely.*

Proof. By contradiction: if there are three cycles in the path c_1, c_2, c_3 and, without loss of generality, an ordering places them as c_1, c_3, c_2, then there exists a path between c_1 and c_2, so at some point a forbidden pattern will appear. $\square$

Theorem 2. *Let G be a graph that is a union of cycles, connected as specified in Definition 2. Then walkability$(G) \leq 2$ if and only if:*

- *the induced cycles form a basis of $H_1(G, F_2)$,*
- *the hypergraph $\mathcal{C}(G)$ is a path,*
- *if $c_1, c_2, c_3 \in \mathcal{C}(G)$ with c_1c_2 and c_2c_3 connected such that $c_1 \cap c_2 \neq \emptyset$, then c_2 must be a triangle (cycle of length 3).*

Proof. $\Rightarrow$) We already know induced cycles form a basis by Proposition 5. If the hypergraph is not a path, it is either a cycle or has a vertex of degree 3. In both these cases, or if condition 3) fails, we reach a contradiction: take a path of 3 vertices in the hypergraph; by Lemma 2 a valid ordering must respect the cycle order. Thus, we can find a path between two non-adjacent cycles for any valid ordering, a contradiction.

$\Leftarrow$) Take the path of cycles and order each so they join properly, then concatenate all orderings. (The ordering method is detailed in **??**) $\square$

Below, we show why the conditions are necessary via examples where they fail and thus the graph is not of walkability 2.

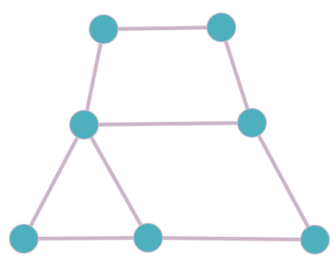

Figure 9: Graph that does not satisfy condition 3).

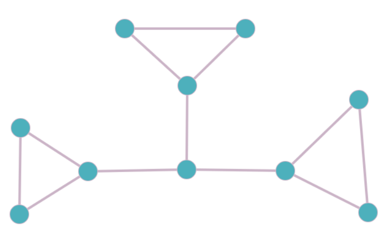

Figure 10: Hypergraph that is not a path.

Having covered cycles, to handle any graph it suffices to see what happens when we attach trees.

3 Trees

Throughout this section, assume the graph is of walkability 2; otherwise, attaching trees won't make it so. Depending on where in the hypergraph path the tree is attached, different conditions must be met, ordered from strongest to weakest.

Now consider internal vertices of a cycle as those that, if the connection between cycles is via an edge, are not part of the connection, or when the connection is via a hinge or path, are not adjacent to the connecting vertices. Example:

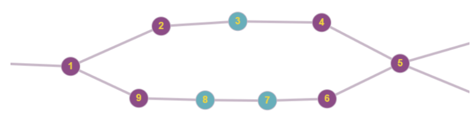

Figure 11: Internal and external vertices of a cycle

In this figure, violet vertices are external, green are internal. We identify them because this cycle connects to two others via vertex 1 (through a path) and vertex 5 (a hinge vertex). Let's call vertices like 1 and 2 connecting vertices. These have the peculiarity of being the only ones with degree 3 or 4 when restricted to vertices belonging to cycles or paths connecting cycles.

3.1 Within the Ordering of a Cycle

Consider a cycle in the middle of the hypergraph path. No tree can be attached to its internal vertices. However, it can have at most two paths attached to its external vertices, with the caveat that on each side, at most one path is allowed.

From connecting vertices, at most one path can emerge if the vertex has degree 3, making it degree 4 and preventing connecting vertices from having trees or paths.

3.2 Between Two Cycles

For the following, consider a cycle inside the hypergraph path.

If the connection is via a hinge vertex, then no tree or path can be attached at that vertex, because the degree would exceed four.

If the connection is via an edge, it depends on whether the cycle it connects to is a triangle or a longer cycle. If it is a triangle and that triangle is connected by an edge to another cycle, or if the cycle is longer, no extra edge can be added.

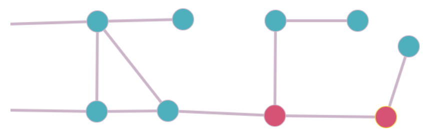

Figure 14: Allowed tree at the end of the hypergraph.

If it is a triangle and the other connection is not via an edge, a path can be added.

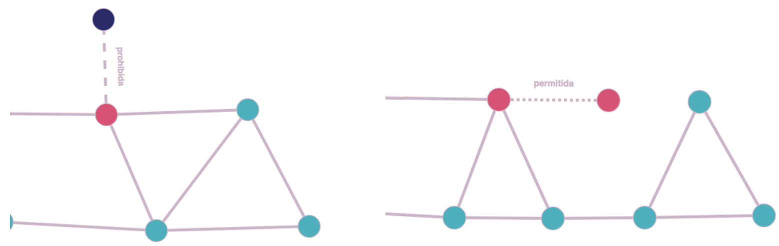

Figure 12: Possibilities for an edge connection with a triangle.

If the connection is via a path, that path can have paths attached to any internal vertex of the path in one of the following forms: a simple path, two simple paths emanating from the same vertex, or a path that bifurcates exactly once at the first vertex not part of the path. Examples:

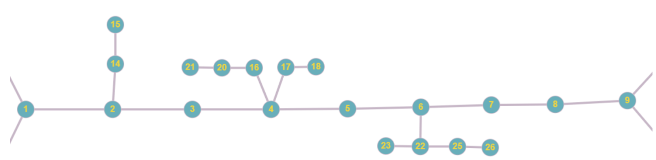

Figure 13: Possible configurations on a path between cycles.

3.3 At the End of the Path

Now consider a cycle at an endpoint of the hypergraph path. Then only one branch of the tree may contain all vertices of degree greater than 3; if there were two, they would overlap.

3.4 Tree Alone

Lemma 3. *A tree is of walkability two if and only if there exists a path connecting all vertices of degree ≥ 3.*

Proof. $\Rightarrow$) We know from earlier that if restricted to one side, to be walkability 2, there must be a path from the starting vertex connecting all vertices of degree greater than 3. Taking the first vertex in the ordering, this holds.

$\Leftarrow$) It suffices to place any vertex of the path in the middle and place each side of the path on each side of the ordering. $\square$

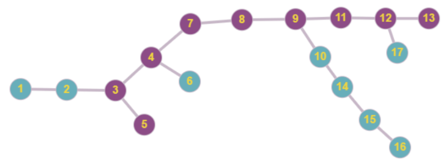

Figure 15: Tree of walkability 2.

4 Defining the Coloring and Ordering

Let G be a graph of walkability ≤ 2, and let $\mathcal{C}(G)$ be the hypergraph. Then there is an initial and final cycle. We call extensions the edges that are not part of a cycle nor of a path connecting cycles.

- For the initial cycle:
  - If it has two extensions, then the cycle vertices with extensions will be the first and second with opposite colors. If it has one extension, the vertex with extension will be the first; if none, choose any as first.
  - The case of final vertices (at most 2): these are the vertices connecting to the next cycle and will be the last (or last and second last), each with different colors.

Color the first and last vertices always the same color. The shortest path between them that does not pass through an already colored vertex is colored the same. If paths are equal in length and no vertex on the path is already colored, choose either path and color all vertices on it the same color as the first and last.

If there were 2 final vertices, color them with different colors and the order is determined by the choice: the first and last always the same color. In this case, color them considering the coloring inside the cycle: Build the shortest path from each to a colored vertex that does not pass through the other final vertex. If both converge to the same color, it is irrelevant which color each gets. If they converge to different colors, the paths and their vertices, including the final vertices, are determined by the color they converge to.

Continue coloring for each cycle in the path.

4.1 Notes on Coloring Extensions

- Trees are colored following the path connecting vertices of degree ≥ 3. The rest are colored the other color. Consider the tree starting from the vertex where the extension occurs; i.e., the vertex belonging to the cycle.
- Each bifurcation from a cycle takes the color of the origin vertex if it occurs at external vertices of cycles, not at connecting vertices. At connecting vertices, it takes the opposite color.
- If the bifurcation occurs on a path, it takes the color of the origin vertex along the path.

After coloring, we proceed to order.

The overall order of vertices in cycles and paths connecting cycles is determined by the order of cycles in the hypergraph, except when two cycles are connected by an edge. In that case, sharing two vertices forces those vertices to appear consecutively in the order, breaking the strict condition that all vertices of one cycle appear before all vertices of the next cycle in the hypergraph. However, ignoring these shared vertex pairs, the property holds.

Thus, it suffices to order each cycle and note that if two cycles are connected by an edge, the order of the next cycle is conditioned by the previous one.

- The order of cycles is defined by the first cycle in the hypergraph. This first cycle defines its order based on the initial coloring described at the beginning of this section. All subsequent cycles are defined from their immediate predecessor.
- Consider two cycles A - B where A precedes B when looking left to right in the hypergraph. Suppose A is already ordered. Then the order of B is defined by A as follows:
  - If A and B are connected by a path or hinge vertex, then the first vertex of cycle B is determined by the last vertex of A (same vertex) and if connected by a path, the first vertex of B is determined by the vertex connected to the path linking to A. Also, the last vertex is the one that in the path of vertices of the same color as the first vertex is last in the path.
  - If A and B are connected by an edge, the order is strictly defined by cycle A because the first is already defined by the order A gave to the shared vertices (connections). Again, the last will be the last vertex of the path of the color of the first vertex.
- All remaining vertices, neither first nor last, can be ordered randomly provided that the next vertex of the same color as the first must come after the next vertex of the opposite color. See the following figures.

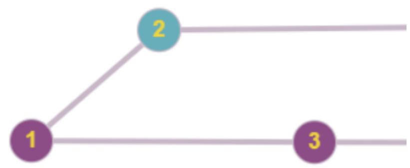

Figure 16: Correct Ordering

Figure 17: Incorrect Ordering

We are now ready to give the certified recognition algorithm for walkability 2:

Algorithm 1 Walkability 2 Recognition (Certified)

```
for each connected component U of G do
    Find the induced cycles of U
    if no cycles then
        if verifies tree conditions then
            return true, order
        else
            return false
        end if
    end if
    if U is a basis of H_1(G, F_2) then
        Build the hypergraph C(G)
        if C(G) is a path then
            for each cycle c in C(G) do
                if verifies the conditions then
                    concatenate to the total order the order of that cycle
                else
                    return false
                end if
            end for
        else
            return false
        end if
    else
        return false
    end if
    concatenate the orders
end for
return true, order
```

This algorithm has complexity $O(n^2(n+m))$.