The undecidability of the joint embedding property for finitely-constrained hereditary graph classes

Samuel Braunfeld

University of Maryland, College Park

July 13, 2018

Atomicity and the JEP

- A permutation class is *atomic* if it cannot be expressed as a union of two proper subclasses.

Lemma

Suppose $\mathcal{K}$ is a permutation class, with no infinite antichain in the containment order. Then $\mathcal{K}$ can be expressed as a finite union of atomic subclasses. Furthermore, the upper growth rate of $\mathcal{K}$ is equal to the maximum upper growth rate among its atomic subclasses.

- We view permutations as structures in a language of two linear orders, and so embeddings correspond to containment.
- A hereditary class of structures $\mathcal{C}$ has the *joint embedding property (JEP)* if for every $A, B \in \mathcal{C}$, there is a $C \in \mathcal{C}$ embedding both.

Lemma

A permutation class is atomic iff it has the JEP.

- The JEP is equivalent to the existence of a weak universal limit.

The decidability of the JEP

Question (Ruškuc, '05 [3])

Is there an algorithm that, given finite set of forbidden permutations, decides whether the corresponding permutation class has the JEP?

- Positive answer in some special cases, such as grid classes [4].
- Positive answer for the stronger property of being a natural class [2].

Theorem (B., '18 [1])

There is no algorithm that, given a finite set of forbidden induced subgraphs, decides whether the corresponding hereditary graph class has the JEP.

- The 2-dimensional nature of permutations seems to be an obstruction to carrying out the argument.
- 3-dimensional permutations? permutation graphs?

The tiling problem

- Proof by reduction from the tiling problem.
- A tiling problem consists of
  - A collection of tile-types $\{t_1, \ldots, t_n\}$
  - Constraints of the form "tiles of type t_i cannot be above (or right of) tiles of type t_j"
- The question is whether tiles can be assigned to cover the grid $\mathbb{N}^2$, respecting the constraints

Theorem (Berger, '66)

There is no algorithm that, given a tiling problem, decides whether it has a solution.

Proof sketch

1. Construct two graphs: A^* representing a grid, and B^* representing a suitable collection of tiles.
2. Choose a finite set of constraints to ensure that successfully joint embedding A^* and B^* requires producing a valid tiling of the grid points in A^* with the tiles from B^*
3. Show that if the tiling problem admits a solution, then the chosen class admits a joint embedding procedure.

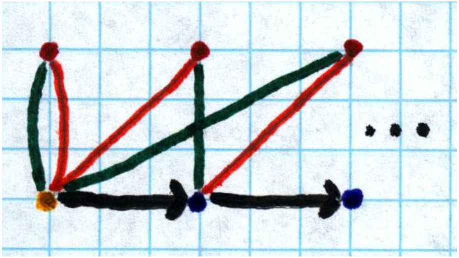
- Steps 1 and 2 ensure that the tiling problem can be solved iff we can joint embed two particular graphs.
- Step 3 ensures that the JEP for the whole class is equivalent to joint embedding for those two graphs.

The language

- We don't work directly with graphs, but in an enriched language.
    1. Ordinary edges E
    2. Directed edges $\vec{E}$
    3. Colored edges E_x, E_y
    4. Colors for vertices $C_1, \ldots, C_k$
- To translate to graphs
    1. Break up special edges using colored vertices.
    2. Attach decorations to vertices to get rid of colors.
    3. Ensure no forbidden subgraphs are created.

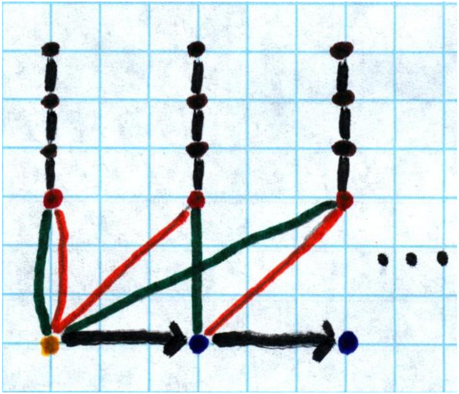
Canonical models (1)

- Recall A^* is supposed to represent the grid $\mathbb{N}^2$. To construct A^*:
  1. Construct a directed path $p_0 \to p_1 \to \dots$
  2. For every pair (p_i, p_j), add a grid point $g_{i,j}$
  3. Add colored edges $g_{i,j} E_x p_i, g_{i,j} E_y p_j$
  4. Every type of vertex (origin, path, grid) gets its own color.



Canonical models (2)

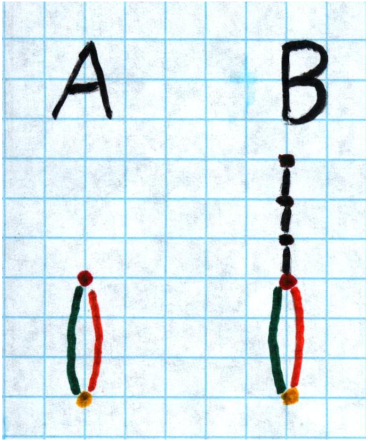
- For the case of graphs, B^* could just be a set of labeled tiles $t_1, \ldots, t_n$.
- For greater flexibility, B^* will be a copy of A^*, but with a full set of tiles attached to each grid point.
- Also, vertex colors in B^* are distinct from A^*.



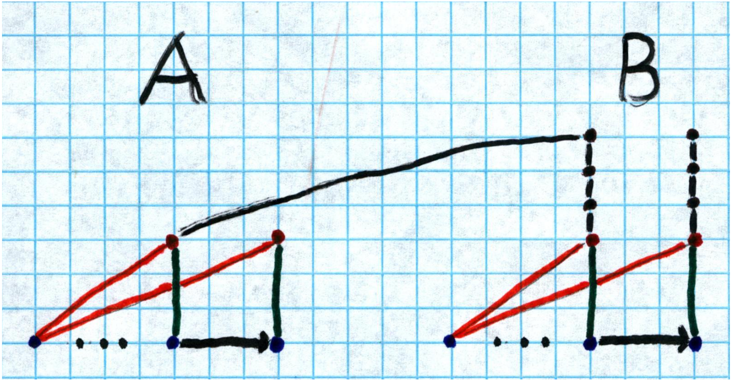
Constraints (1)

- We wish to ensure joint embedding A^* and B^* solves the tiling problem.
- A grid point in A^* is “tiled” if it is connected to a tile vertex from B^*.
- We want to force that given a grid point in A^*, it is tiled by a tile from B^* with the same coordinates.
- This is not a local condition.
- Instead, first force the origin to be tiled, then propagate the tiling.
- Also add constraints enforcing the tiling constraints.

Origin-tiling constraint



Propagation constraints



Constraints (2)

- We would like to force every graph to look like A^* or B^*.
- We cannot enforce "totality" conditions, so must allow for partial structures.
- The key property we need is every grid point has at most one set of coordinates.
- Other constraints include:
  1. Grid points have at most one E_x or E_y-neighbor.
  2. Origin vertices have at no $\rightarrow$-predecessor.
  3. Path vertices have at most 1 $\rightarrow$-predecessor.

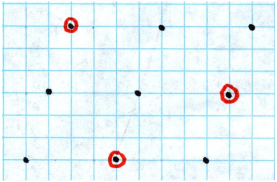
$$\vdots$$

Reducing from the tiling problem

- Given a tiling problem $\mathcal{T}$, create the corresponding graph class $\mathcal{G}_{\mathcal{T}}$.
- One direction is easy: If $\mathcal{G}_{\mathcal{T}}$ has the JEP, then can joint embed A^* and B^*, and read off a solution to the tiling problem.
- Now suppose $\mathcal{T}$ has a solution $T : \mathbb{N}^2 \to \{t_1, \ldots, t_n\}$. Given $A, B \in \mathcal{G}_{\mathcal{T}}$:
  1. Take the disjoint union $A \sqcup B$. (Note there is no uniqueness condition on origins, grids, etc.).
  2. If not finished, then a grid origin in A and one in B with a full set of tiles, so add an edge according to $T(0,0)$.
  3. If not finished, then need to propagate tiling. As every grid point has well-defined coordinates, we just use $T(x,y)$.
  4. Check that the new edges don't create any forbidden substructures.
- A key property of graphs is that placing an edge between two vertices has no effect on whether we can place an edge or not between other vertices.
- In contrast, suppose $x < x' \in A$ and $y \in B$. If place $x' < y$, then must place $x < y$.

Moving to permutations

- The 2-dimensional nature of permutations seems to makes it difficult to represent a grid (in classes other than the class of all permutations).
- In the straightforward representation of an $n \times n$ grid, it is easy to find any permutation of length $\leq n$.



Question

Is there a permutation class (other than the class of all permutations) that represents arbitrarily large grids such that the neighbor relation is:

1. *local*
2. *determined only by the presence (not absence) of a pattern*

References

[1] Samuel Braunfeld, *Infinite Limits of Finite-Dimensional Permutation Structures, and their Automorphism Groups: Between Model Theory and Combinatorics*, PhD Thesis, Rutgers University, New Brunswick, 2018.

[2] Mike D Atkinson, Maximillian M Murphy, and Nik Ruškuc, *Pattern avoidance classes and subpermutations*, The Electronic Journal of Combinatorics **12** (2005), no. 1, 60.

[3] Nik Ruškuc, *Decidability questions for pattern avoidance classes of permutations*, Third International Conference on Permutation Patterns, Gainesville, Fla, 2005.

[4] Stephen D Waton, *On permutation classes defined by token passing networks, gridding matrices and pictures: three flavours of involvement*, University of St Andrews, 2007.