Strict upward planar GRID DRAWINGS

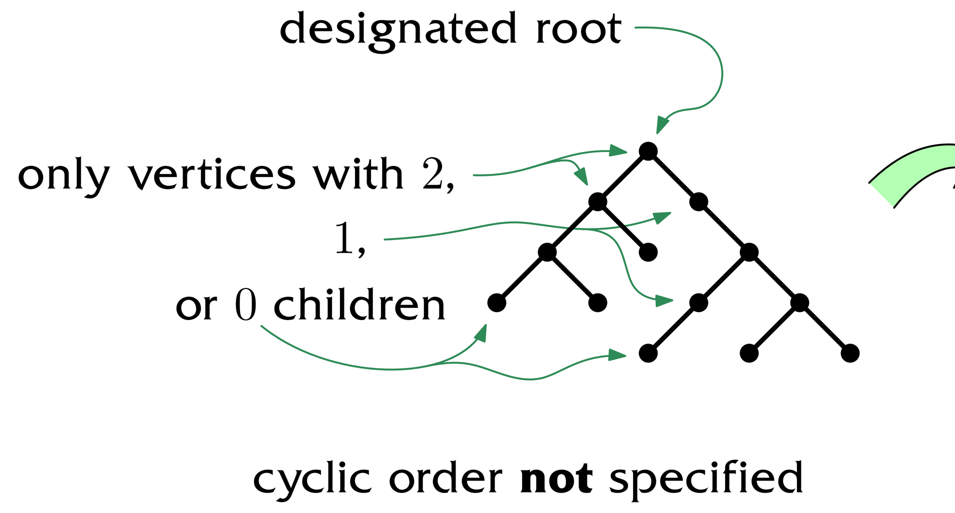
of binary trees with minimal area

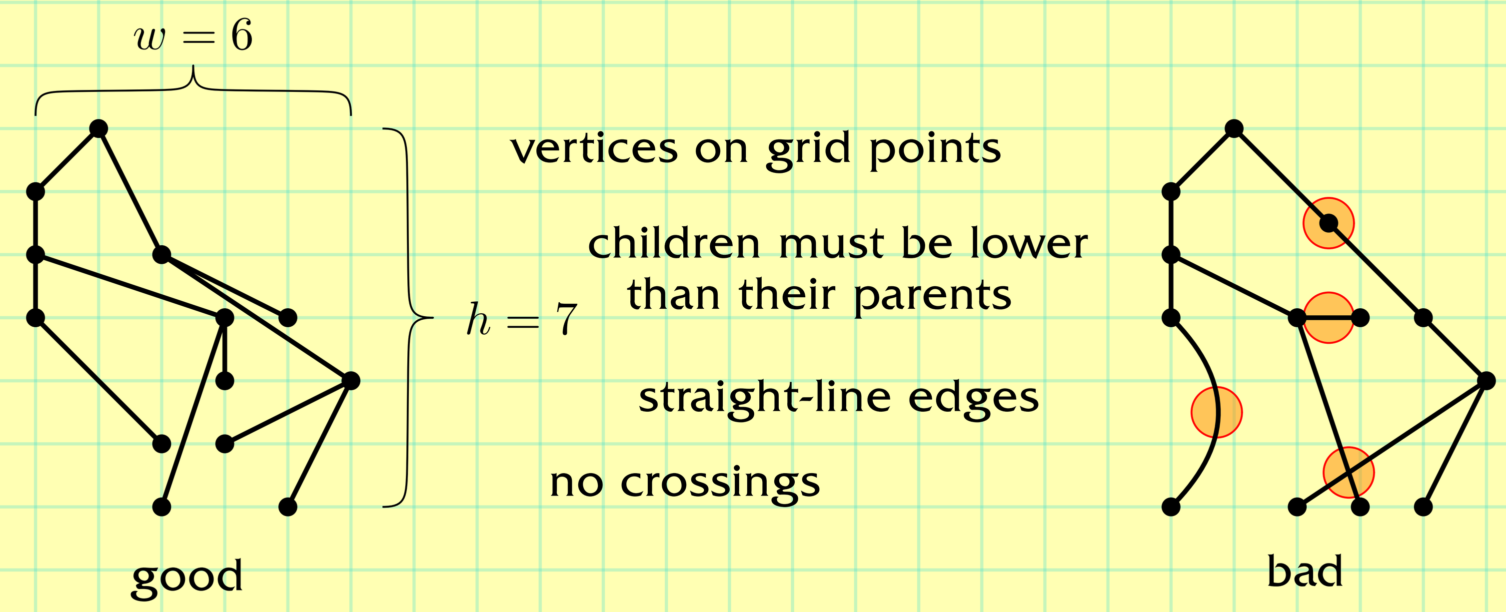
Maarten Löffler ∘ Utrecht University

Problem

Problem statement. Given a rooted binary tree T and a tuple (w, h), test whether there exists a strict upward drawing of T on a $w \times h$ grid.

History. Upward drawings of binary trees with fixed combinatorial embeddings [Akatiya et al 2018] and upward drawings of trees with free combinatorial embeddings [Biedl et al 2017] are both known to be hard.





Result & Approach

Theorem. Testing whether a tree T can be embedded on a $w \times h$ grid is NP-complete.

Corollary. This implies that minimizing w (width), h (height), $w + h$ (perimeter), or wh (area) is also hard.

Reduction. We reduce from monotone not-all-equal-3SAT.

Given a n variables and m "clauses" (triples of variables) ...

(x_1, x_2, x_3) (x_1, x_2, x_4) (x_1, x_3, x_4)

...find an assignment such that all variables in each clause are neither all true nor all false.

(x_1, x_2, x_3) (x_1, x_2, x_4) (x_1, x_3, x_4)

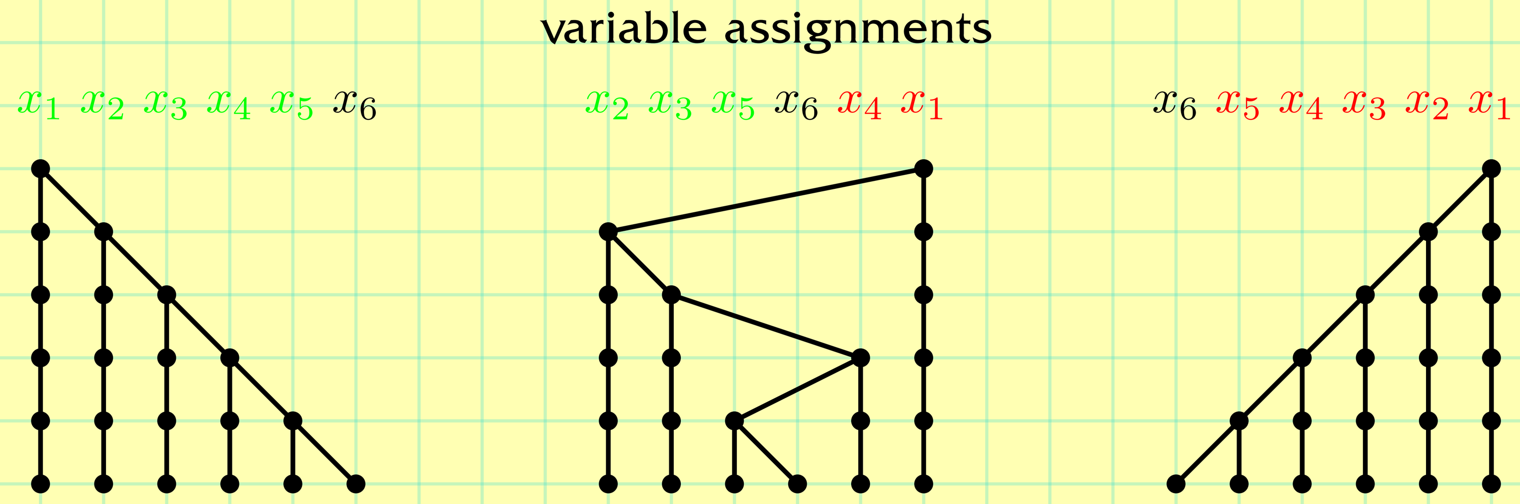
Variable Gadget

Variables are represented by paths in T of length h. In a strict upward drawing, these paths must have one vertex on every row.

The top $n + 1$ rows contain a permutation gadget. In the ith row, we can choose to place the path representing x_i either on the left or on the right, which encodes the true and false states.

We introduce a "dummy variable" x_{n+1}, which has no truth assignment but separates the true and false variables.

The vertices in the $n + 1$st row, and thus the paths hanging from them, have 2^n possible permutations. Specifically, these permutations have all true variables on the left with increasing indices, and all false variables on the right with decreasing indices.



Clause Gadget

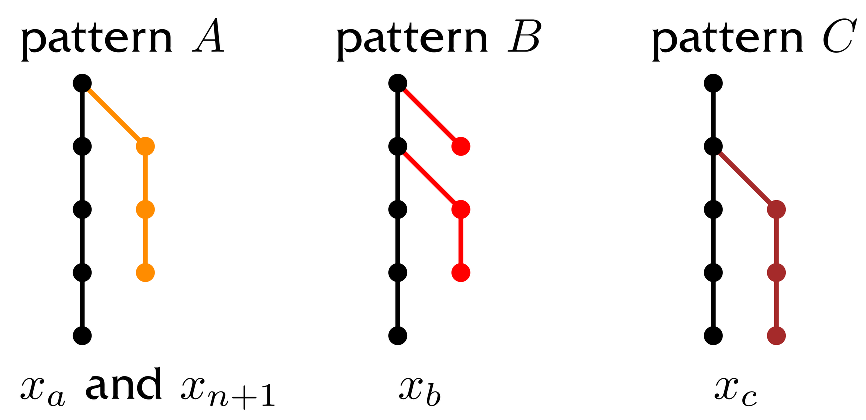
Each clause is represented by four rows.

A clause (x_a, x_b, x_c) with $a < b < c$ is encoded using the paths for x_a, x_b, x_c, and the dummy variable x_{n+1}.

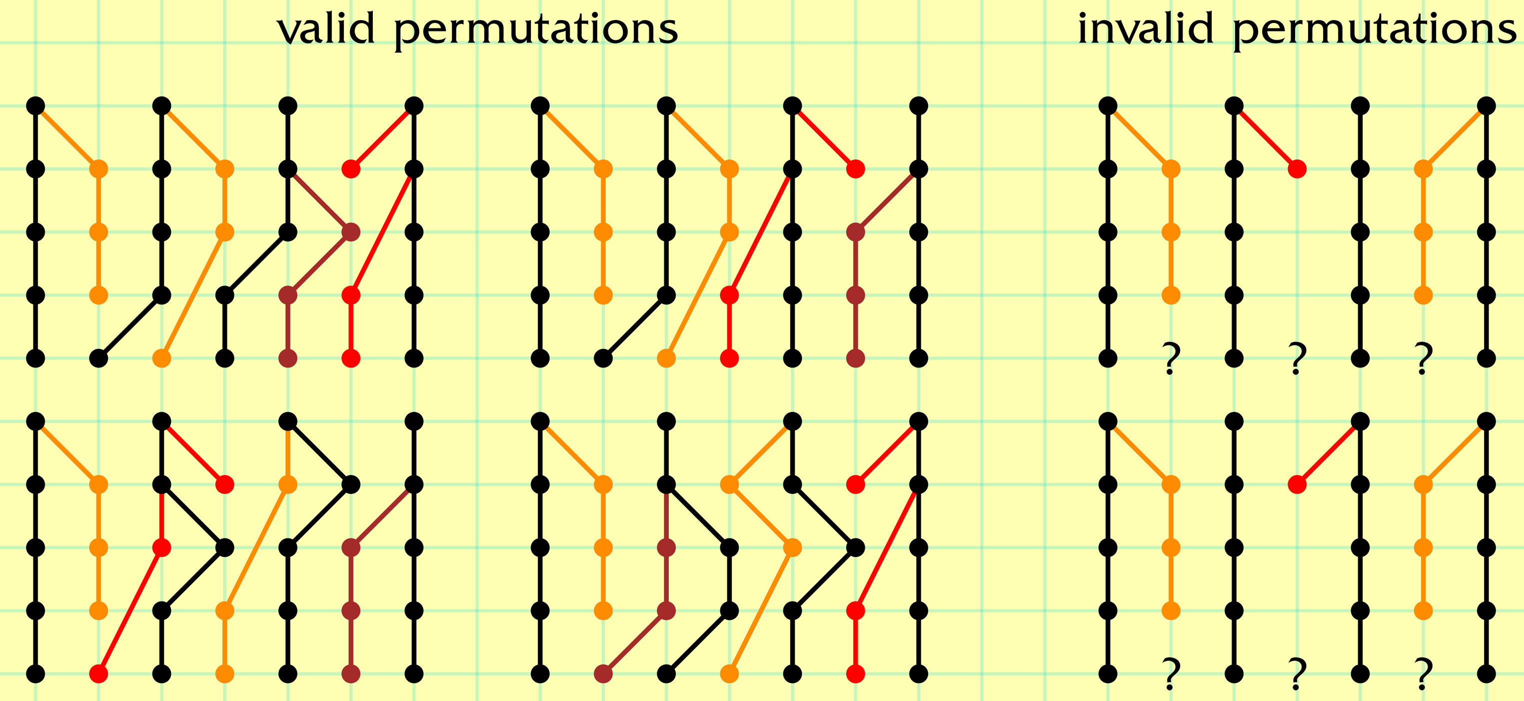
Each of these has 3 additional vertices in one or two subtrees, so 12 additional vertices in total.

The total width of the construction is $n + 4$, so there are three "empty columns", giving 12 empty spots in these four rows.

We use 3 different patterns for attaching the additional vertices.



Lemma. There is a valid embedding if and only if the two outermost patterns are not both A.



Full Example

We now present the full construction for a monotone NAE-SAT formula with 5 variables and 6 clauses.

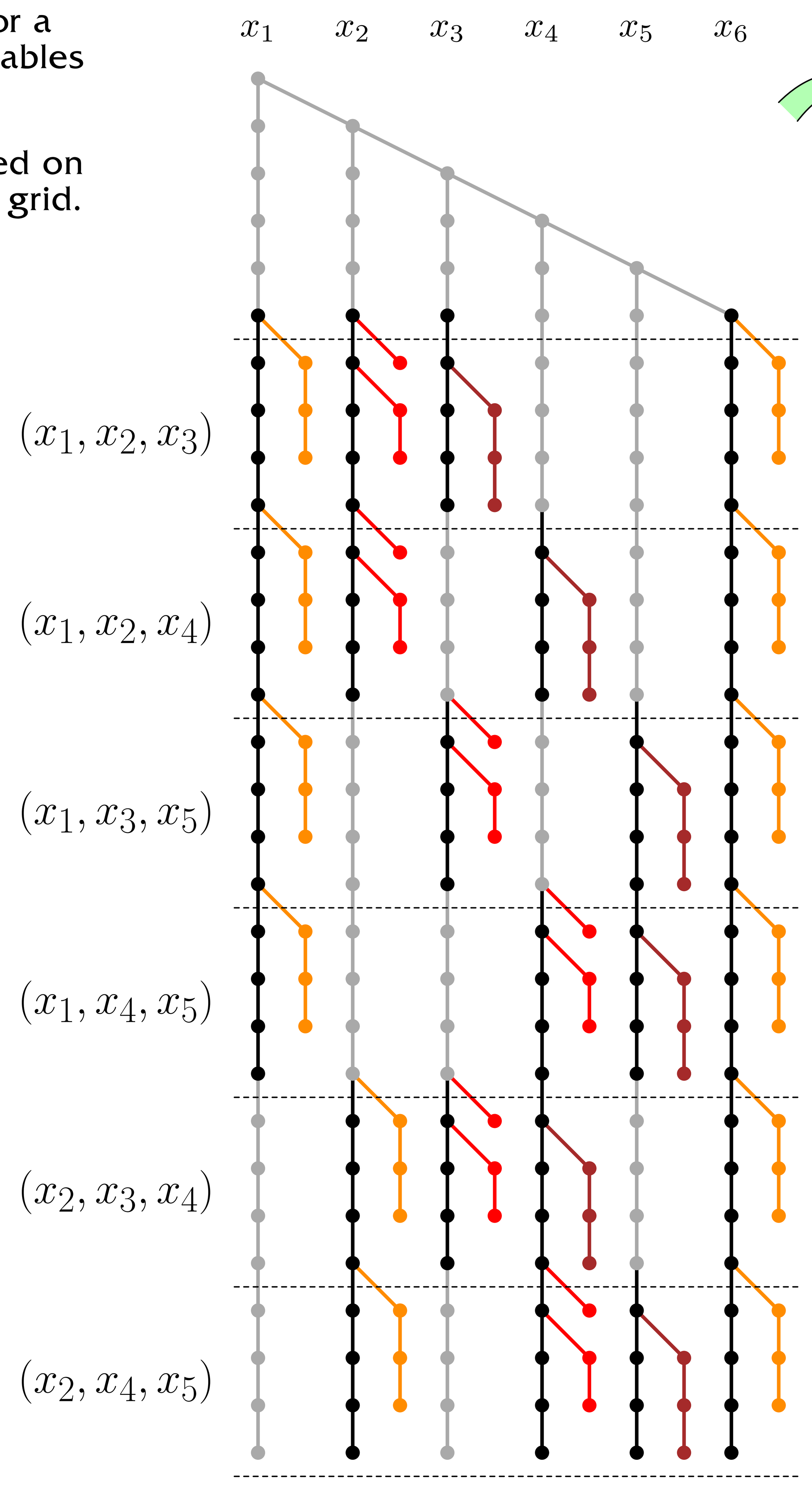
The constructed tree must be embedded on a $w = n + 4 = 9$ by $h = n + 4m + 1 = 30$ grid.

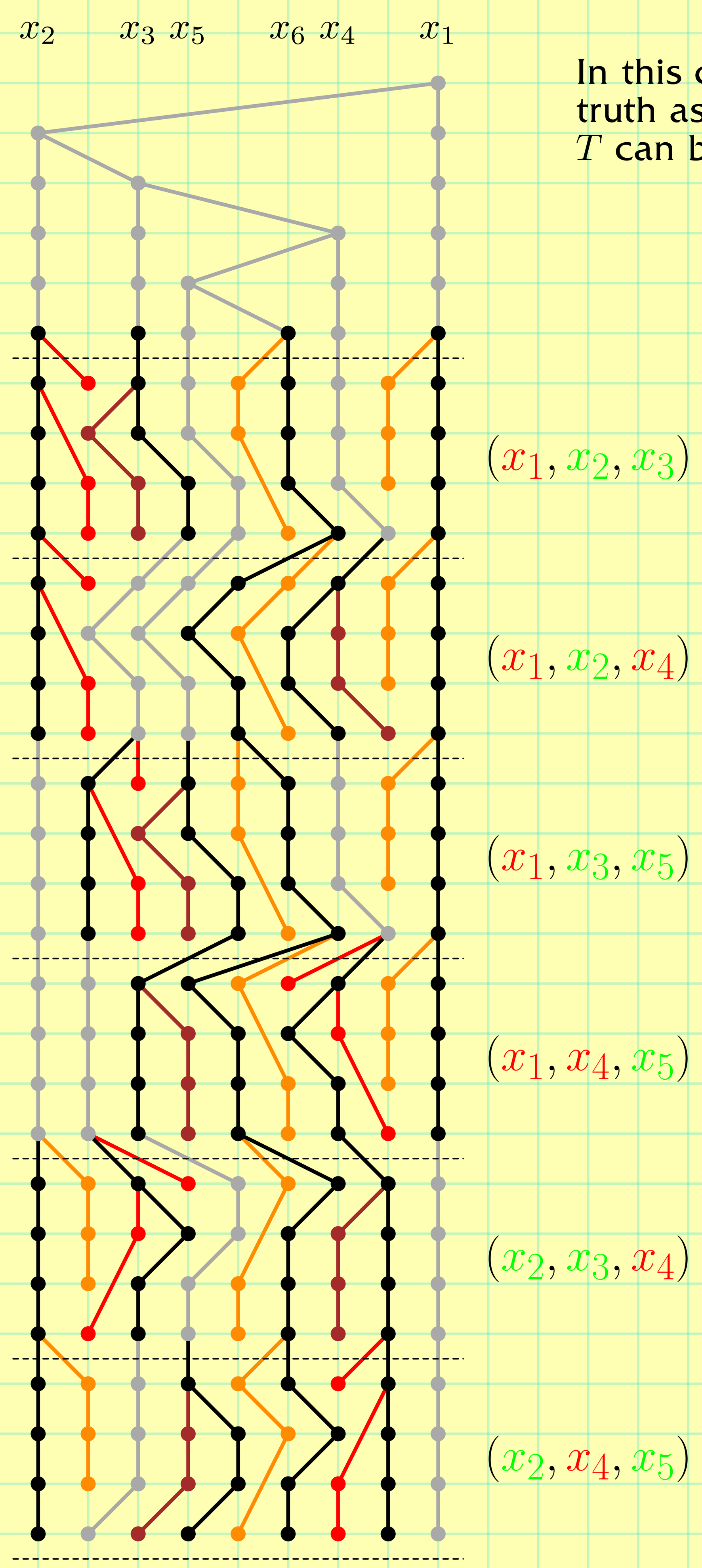
Active sections of variable chains in a clause are drawn in black; inactive sections are drawn in grey.

Observation. An edge between two free spots on consecutive rows can always be drawn.

This means that inactive chains can always be drawn inbetween active sections, independent of how they are embedded.

It also means attaching consecutive clause gadgets is no problem independent of where the black path ends.





In this case, there is a satisfying truth assignment, and therefore, T can be drawn in a 9×30 grid.

Future Work

Our result strengthens [Biedl et al., 2017] and complements [Akatiya et al., 2018] and settles the complexity of finding minimum-area strict upward drawings of trees.

Our construction relies critically on the strictness of the drawings. What is the complexity of finding **non-strict** upward planar embeddings of trees on a given grid?