Aperiodic Tilings

Thomas Fernique, Ilya Ivanov-Pogodaev, Alexei Kanel-Belov and Ivan Mitrofanov

Additional hints for the construction of hierarchial tiling decorations.

Let us consider the trapezoid substitution in more detail (Fig 1). We can see that four tiles form a macrotile, four macrotiles form a higher level macrotile and so on. We want to assign some local rules to enforce that tiles can only form hierarchical tilings associated with the trapezoid substitution We will use another method for local rules setting.

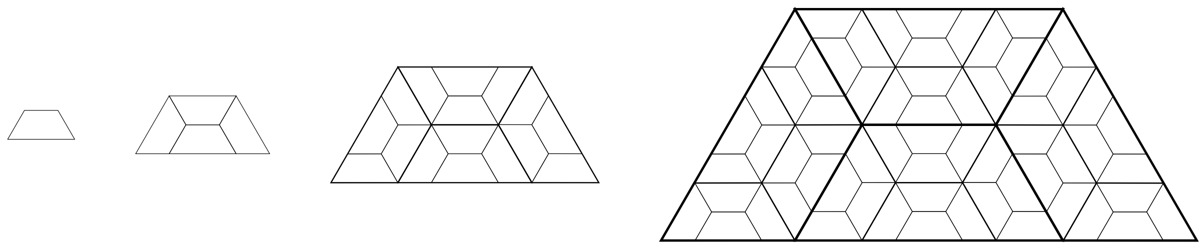

Рис. 1:

Let us set up a language that encodes the different tile types into the vertices and edges. We say that a tiling of the plane is *correct* if it hierarchical for the trapezoid substitution. We say that a pattern is *correct* if it appears in correct tiling. We assign a code for our substitution system and from the other side we prove (by rank induction) that our code enforce the correct patterns.

E.7 Show that every vertex on a correct pattern has one of the three types on Fig. 2. (Trapezoids around the vertices can be macrotiles of any level.) Show that every macrotile edge on a correct pattern has one of the five types on Fig. 3. Show that every macrotile edge is the base-edge of two (macro)tiles.

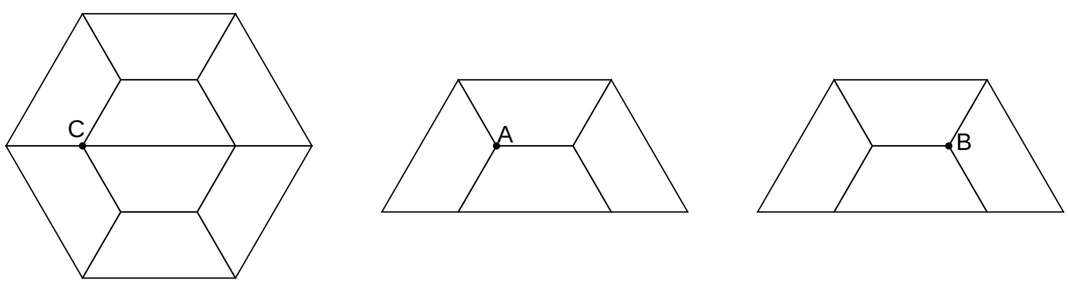

Рис. 2:

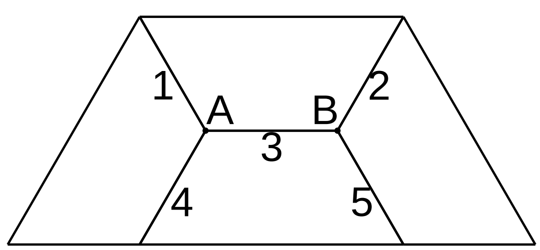

Рис. 3:

How can one code tile edges? Every edge has the same type as the maximal macrotile edge that can contain it. So, every tile has four vertices and four edges. The number of vertices (and edges) types is finite. The idea of forbidden words could be used here again – we can consider paths on a tiling encoded by sequences of vertices and edges types and construct lists of forbidden paths.

E.8 Find an example of sequence having the form XYZ, (there X, Z –are types of edges, Y is a type of vertex) which could not code any path on a correct pattern.

Our goal is to use forbidden paths to ensure that the only possible tilings are correct ones. But we need to work a bit.

Let us add a new type of vertex, D. This type codes vertices on a large base-edge that are part of the next lower hierarchy level of the tiling (Fig. 4). The vertices on this base-edge which are part of further lower levels of the hierarchy are of type C.

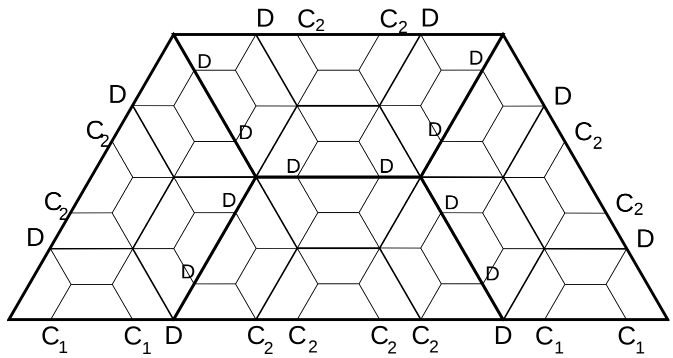

Рис. 4:

Also we assign new colors to C vertices. For every such vertex, we can find the tile on which base it is located We can say that the vertices located between an edge of the base and a D vertex are colored in a first color, while vertices between D vertices (they could be located far from them) are colored in a second color.

We also need additional colors for edges. Every edge is a side of a smallest tile or is situated on a big base of some macrotile. Let X,Y be types of vertices (with colors) in the ends of that big base (or the ends of the smallest tile side, if our edge is the side of smallest tile). We assign an ordered pair of "bosses"(X, Y) as the color of our edge. While we speak about paths, we have a direction on each edge, so we known which are the previous and the next bosses.

E.9 Look at the picture with four levels of substitutions and mark types of all vertices and edges with their colors.

E.10 Which types of edges start at vertices of each type?

So, we can number incoming edges for any type of vertex. Thus, we can know where are the incoming path from and where it goes. We will forbid paths which do not appear on a correct tiling by looking each path as a finite sequence of traveled vertices and edges and by memorizing through which edge we enter and exit each vertex.

E.11 Find the list of forbidden paths to obtain the following property: every path along the edges that form the big base of some macrotile contains exactly two vertices of type D (except the ends of that path).

E.12 Try to construct a finite list of forbidden paths to obtain the following properties: we can construct a tiling using our tiles, and every such tiling is correct.