FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Josef Minařík

Unfolding some classes of polycubes

Department of Applied Mathematics

Supervisor of the bachelor thesis: Mgr. Jan Kynčl, Ph.D.
Study programme: Computer Science
Study branch: General Computer Science

Prague 2022

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to thank my supervisor Mgr. Jan Kynčl, Ph.D. for introducing me to this topic and for his assistance on this thesis.

Title: Unfolding some classes of polycubes

Author: Josef Minařík

Department: Department of Applied Mathematics

Supervisor: Mgr. Jan Kynčl, Ph.D., Department of Applied Mathematics

Abstract: An *unfolding* of a polyhedron is a cutting along its surface such that the surface remains connected and it can be flattened to the plane without any overlap. An *edge-unfolding* is a restricted kind of unfolding, we are only allowed to cut along the edges of the faces of the polyhedron. A *polycube* is a special case of orthogonal polyhedron formed by glueing several unit cubes together face-to-face. In the case of polycubes, the edges of all cubes are available for cuts in edge-unfolding. We focus on one-layer polycubes and present several algorithms to unfold some classes of them. We show that it is possible to edge-unfold any one-layer polycube with cubic holes, thin horizontal holes and separable rectangular holes. The question of edge-unfolding general one-layer polycubes remains open. We also briefly study some classes of multi-layer polycubes.

Keywords: polycube, planar net, unfolding polyhedra, cutting and folding

Contents

Introduction

An *orthogonal polyhedron* is a polyhedron whose edges are parallel to the Cartesian axes and whose faces meet at right angles. Each face of an orthogonal polyhedron is parallel to one of the Cartesian coordinate planes. A *polycube* is a special case of an orthogonal polyhedron. It is formed by glueing several unit cubes together by whole faces. Polycubes are three-dimensional generalizations of planar polyominoes. A *one-layer* polycube is a polycube of height 1. In other words, the centers of all unit cubes are in one plane. One-layer polycubes with non-zero genus have some *holes* in them. If the hole consists of only one missing unit cube, we call this hole *cubic*.

An *unfolding* of a polyhedron is a cutting along its surface such that the surface remains connected and it can be flattened to the plane without any overlap. We usually only care about interior overlap and there may be touching edges after unfolding to the plane. An *edge-unfolding* is a restricted kind of unfolding. In this case, we can only cut along the edges of the faces of the polyhedron. It is quite easy to show that there exist non-convex orthogonal polyhedra that cannot be edge-unfolded [8]. We are mostly interested in edge-unfolding of polycubes. In the case of polycubes, the edges of all cubes are available for cutting. This means that we can cut the faces of our polyhedron along the edges of the 1×1 grid. Different kinds of unfolding are discussed in greater detail by O'Rourke [7] [9].

Definition. Let $\mathcal{P}$ be a polycube. We define the *face graph* $F(\mathcal{P})$ as a graph whose vertices are the squares of the surface of $\mathcal{P}$. Two vertices are connected by an edge if the corresponding squares share an edge.

Unfolding a polycube $\mathcal{P}$ is equivalent to finding a suitable spanning tree of $F(\mathcal{P})$. Given a spanning tree of $F(\mathcal{P})$, there is a unique way to flatten $\mathcal{P}$ to the plane without cutting the spanning tree.

Definition. Let $\mathcal{P}$ be a polycube. We define the *cube graph* $C(\mathcal{P})$ as a graph whose vertices are the cubes of $\mathcal{P}$. Two vertices are connected by an edge if the corresponding cubes share a face.

Unfoldings of many classes of orthogonal polyhedra have been studied; for example rectangle-faced orthostacks [2], orthotubes [1] or Manhattan towers [5]. There are also known edge-unfoldings of special cases of polycubes, such as well-separated orthotrees [4]. Otrhotrees are polycubes whose cube graph forms a tree. We briefly discuss one-layer orthotrees in Section 1.8 and show that it is possible to unfold them into a polygon of height 3.

Theorem 1. *It is possible to edge-unfold any one-layer orthotree such that the produced polygon has height 3.*

One-layer orthogonal polyhedra with arbitrary genus g can be edge-unfolded using only $2(g-1)$ additional cuts [3]. Kiou, Poon and Wei proved that it is possible to edge-unfold one-layer polycubes with sparse cubic holes [6], which are one-layer polycubes with cubic holes such that each connected component in a column contains at most one hole. We generalize this result in Section 1.5 and present an algorithm for unfolding general one-layer polycubes with cubic holes.

Theorem 2. *It is possible to edge-unfold any one-layer polycube with cubic holes.*

In Sections 1.6 and 1.7 we further generalize this approach to other classes of one-layer polycubes.

Definition. A hole is called *thin horizontal* if it is a rectangle of height 1.

Theorem 3. *It is possible to edge-unfold any one-layer polycube with thin horizontal holes.*

Definition. We call a set of rectangles *separable* if it satisfies the following property. If we extend any edge of any rectangle to a line, it does not cut any other rectangle.

Theorem 4. *It is possible to edge-unfold any one-layer polycube with separable rectangular holes.*

Note that cubic holes are both thin horizontal and separable rectangular. We study quite an opposite kind of polycubes in Section 1.4.

Definition. We say that a hole is *wide* if it satisfies the following property. If there is a missing cube with a center at coordinates $[x, y]$, then there is at least one missing cube at coordinates $[x+1, y], [x-1, y]$ and at least one missing cube at coordinates $[x, y-1], [x, y+1]$.

Theorem 5. *It is possible to edge-unfold any one-layer polycube with wide holes.*

Definition. A set of points in the plane is called *y-convex* if its intersection with any vertical line forms a line segment.

In Section 1.9 we present an algorithm to unfold any polycubes with y-convex holes and at most one hole in every column. This time we use a different approach than for the other classes. Internal boundary of every hole remains connected even after the unfolding.

Theorem 6. *It is possible to edge-unfold any one-layer polycube with y-convex holes and at most one hole in every column.*

Multi-layer polycubes

We also describe some algorithms for multi-layer polycubes. In particular, we show that it is possible to unfold stacks of one-layer orthotrees or paths given some additional constraints.

Definition. A *treestack* is a polycube whose every layer (set of cubes with the same z-coordinate) forms an orthotree.

Definition. A *pathstack* is a polycube whose every layer forms an orthopath.

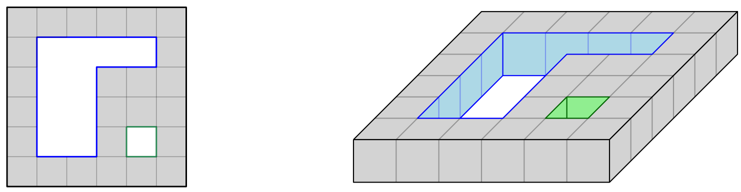
We will discuss this in more detail in Sections 2.1 and 2.2. The classes of polycubes we unfold contain polyhedra of arbitrary genus. Edge-unfolding polyhedra with high genus is in general a very difficult problem. An example of high genus multi-layer polycubes that can be edge-unfolded are level 1 Menger polycubes [10].

1. One-layer polycubes

1.1 Definitions

Let us consider a one-layer polycube $\mathcal{P}$ placed in the xy plane such that the centers of all cubes have integer coordinates. The exact position of the polycube is not important; we only need to be able to index the cubes by coordinates. By a cube with coordinates x, y we mean a cube whose center has such coordinates. Let us denote the set of holes $\mathcal{H}$. A polycube $\mathcal{P}$ has a top base T and a bottom base B. There is also an *external boundary* E and several *internal boundaries* $\mathcal{I} = \{I_h \mid h \in \mathcal{H}\}$, each corresponding to some hole h. The boundaries are formed by cyclic stripes of unit squares.

Since we are only interested in one-layer polycubes, we will display them as 2-dimensional objects. In all of the figures, we are looking at the polycube from above, which means that we see the top base, see Figure 1.1. With respect to that, we will be using terms such as "left", "right", "up" and "down" to describe directions. For example, the boundary of a hole consists of four not necessarily connected parts: left, right, top, and bottom.

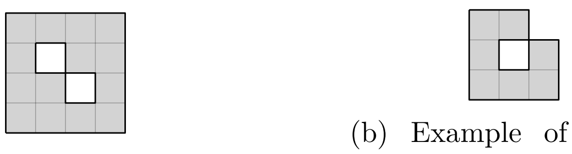


(a) Polycube seen from above.

(b) Polycube as a 3-dimensional object.

Figure 1.1: Example of a polycube.

We require the surface of $\mathcal{P}$ to be *simple*, that is, every edge of $\mathcal{P}$ is incident with exactly two 1×1 squares on the surface of $\mathcal{P}$. The holes are not allowed to "touch" each other by corners nor to "touch" the external boundary, examples of such disallowed configurations are in Figure 1.2.



(a) Example of holes touching.

(b) Example of a hole touching the external boundary.

Figure 1.2: Examples of polycubes that are **not** allowed.

1.2 Algorithms

We will describe several algorithms for unfolding one-layer polycubes. Let n denote the number of unit cubes that form $\mathcal{P}$. All of the presented algorithms can be implemented in $\mathcal{O}(n)$ time if we are provided with a reasonable representation of the polycube as input (for example a sorted list of all unit cubes). We mainly focus on the existence of the unfolding and the existence of such an algorithm is more important to us than the exact implementation. However, an implementation of all the presented algorithms should be mostly straightforward.

In many of the algorithms, it might be more natural to focus at the spanning tree of $F(\mathcal{P})$ we are constructing rather than at the actual cuts.

1.3 No holes

Let us start with a simple example to get familiar with the techniques we will be using. Without holes, we only need to unfold B, T and E. The algorithm starts with the external boundary E. The external boundary can be unfolded into a single stripe of height 1. Let us place this stripe horizontally in the plane. We do not cut B and T. We simply connect them to the unfolded E, each being placed in a different half-plane. They are connected to E by the cube with the lowest y coordinate (if there are more of them, we can choose one arbitrarily). The resulting shape is connected, and it is easy to see that there are no overlaps. See Figure 1.3 for an example of an unfolding of a polycube without holes.

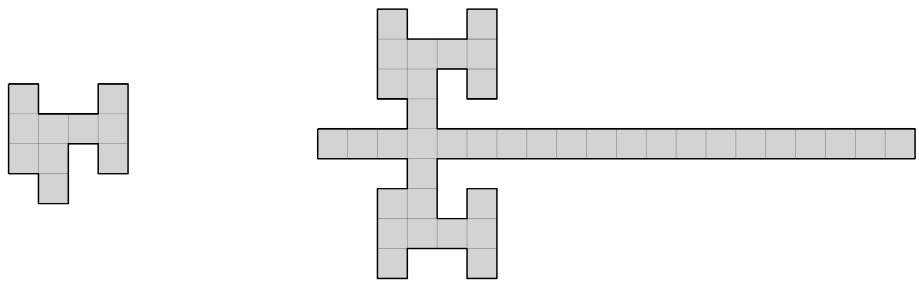


Figure 1.3: Unfolding of a one-layer polycube without holes.

Note that this is an edge-unfolding in the standard sense, we only used cuts along the faces of the polyhedron. We did not use any additional cuts along the edges of the unit cubes.

1.4 Wide holes

Definition. We say that a hole is *wide* if it satisfies the following property. If there is a missing cube with a center at the coordinates $[x, y]$, then there is at least one missing cube at the coordinates $[x + 1, y], [x - 1, y]$ and at least one missing cube at coordinates $[x, y - 1], [x, y + 1]$. In other words, there are no interior points of two parallel faces of the same hole with distance 1.

See Figure 1.4 for an example of wide holes. We can unfold one-layer polycubes with wide holes using the following algorithm. We start by unfolding B, T and E in the same way as above in Section 1.3. Due to the wideness of the holes, there is a lot of space inside B and T. For every hole h, we will unfold I_h in two steps. In the first step, we unfold the top and the bottom faces of I_h. In the second step, we unfold the left and the right faces of I_h. In the first step, we unfold parts

(a) Examples of wide holes. (b) Examples of **non**-wide holes.

Figure 1.4: Examples of wide and non-wide holes.

of I_h into the top base T, inside the holes. There will be no overlap because the holes are wide. The second step is almost the same, the only difference is that we use the bottom base B instead. Figure 1.5 shows an example of an unfolding produce by this algorithm.

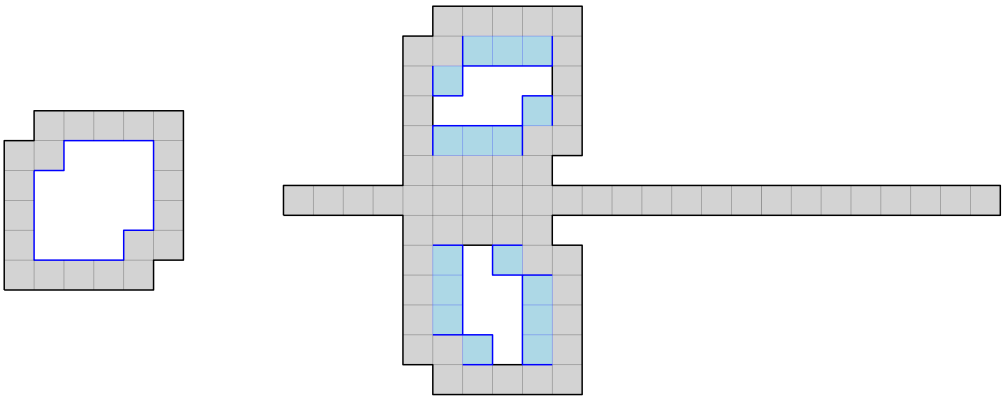


Figure 1.5: Unfolding of a one-layer polycube with a wide hole.

Again, we used only cuts along the edges of $\mathcal{P}$.

1.5 Cubic holes

The algorithm for unfolding one-layer polycubes with cubic holes is slightly more complicated; we will need to cut T and B. Note that cutting T or B is necessary to unfold even a single cubic hole. The idea is similar to the algorithm in Section 1.4, we will unfold some parts (the top and the bottom faces) of the internal boundaries by connecting them to T and some of them (the left and the right faces) by connecting them to B.

The beginning is still the same, we unfold the external boundary E. Now, let us color the squares of T using orange and red. The squares whose y-coordinate is 0 or 1 modulo 4 will be orange, the remaining ones will be red. In other words, we are coloring pairs of rows orange and red. Example of such coloring can be seen in Figure 1.7. Consider the connected components formed by orange or red squares, that would be formed by cutting edges separating squares of different colors. The leftmost and the rightmost square of every connected component must be incident to E. This is because the holes are cubic; they are not large enough to separate the components.

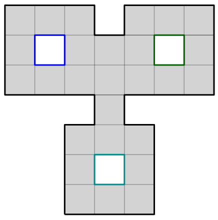


Figure 1.6: Example of a one-layer polycube with cubic holes.

(a) Coloring of the top base.

(b) Coloring of the bottom base.

Figure 1.7: Coloring of the one-layer polycube with cubic holes in Figure 1.6.

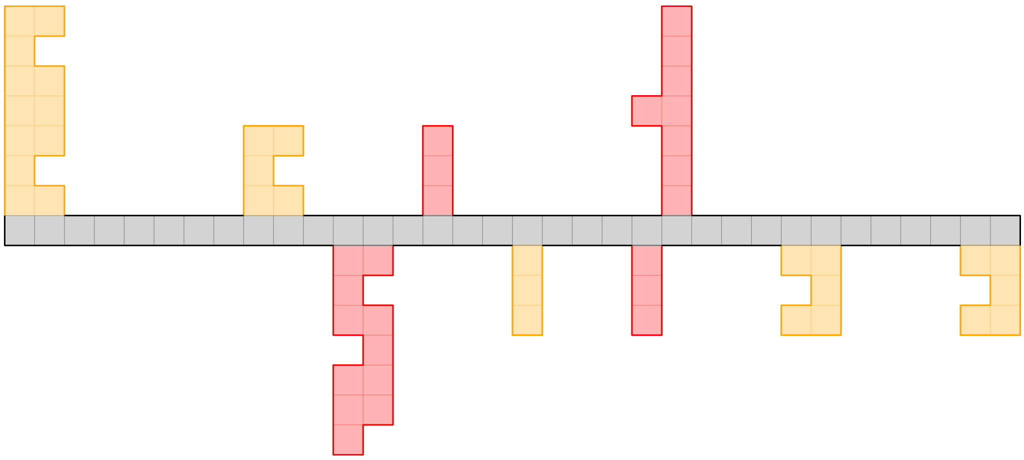


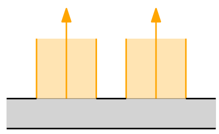
Figure 1.8: The first step of unfolding the polycube in Figure 1.6.

We will connect all the orange components to the external boundary on their left side by their leftmost square. Analogously every red component will be connected to the boundary by its rightmost square. An example of the current stage of unfolding is shown in Figure 1.8. Quite simple casework shows that there is a distance of at least 2 between any pair of connected components after placing them in the plane next to unfolded E. Suppose that there are two stripes that have a distance of less than two. There are two cases:

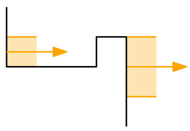
1. Both of the stripes have the same color. We can suppose without loss of generality they are orange. Now consider where these stripes come from in the polycube. They either come from the same pair of rows or a different pair of rows. In the first case, the distance would have to be at least 3, in

the second case, it would have to be at least 2, a contradiction. See Figure 1.9 for an illustration.

2. The stripes have different colors. Without loss of generality, we can assume that the left stripe is orange. Let us again consider where those stripes were before the unfolding. If they don't come from the neighboring pair of rows, the distance would obviously have to be at least 4. There are two remaining (symmetric) cases: the red rows could be either above or below the orange rows. In both of those cases, the distance is at least 2, contradiction again, see Figure 1.10.



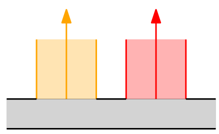
(a) Suppose the distance of two orange stripes is at most 1.



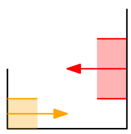
(b) In the first case, the distance after unfolding must be at least 3.

(c) In the second case, the distance after unfolding must be at least 2.

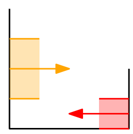
Figure 1.9: Two orange stripes cannot be too close to each other.



(a) Suppose the distance of an orange and red stripe is at most 1.



(b) In the first case, the distance after unfolding must be at least 2.



(c) In the second case, the distance after unfolding must be at least 2.

Figure 1.10: Two stripes of different colors cannot be too close to each other.

Now, we will take every left or right face of the internal boundaries and connect it to the only square of the already unfolded top base to which it is incident. There are no overlaps because the connected components have a distance of at least 2 and there is enough space for two unit squares between them. See Figure 1.11.

We repeat the process for the bottom base B. This time, we color pairs of columns instead of rows. This base and parts of holes are unfolded to the opposite half-plane, so there will be no overlaps with previously placed parts.

1.6 Thin horizontal holes

The approach in Section 1.5 can be quite easily generalized to holes of dimensions $1 \times k$, but only if all of them are oriented in the same way (either all horizontal or all vertical). In this section, we assume that all holes are horizontal.

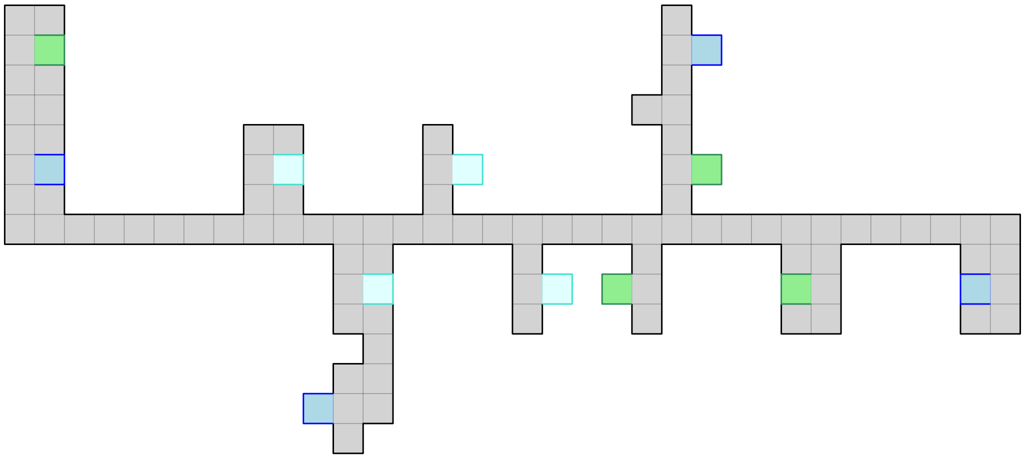
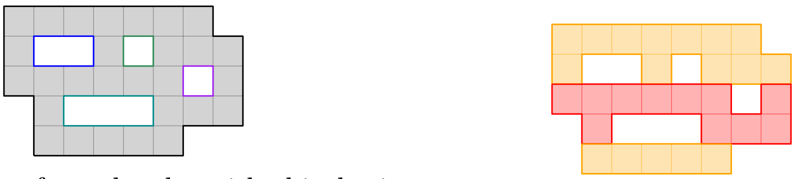


Figure 1.11: Unfolding of the one-layer polycube with cubic holes in Figure 1.6.

Definition. A hole is called *thin horizontal* if it is a rectangle of height 1.



(a) Example of a polycube with thin horizontal holes.

(b) Coloring of the top base.

Figure 1.12: Coloring of a one-layer polycube with thin horizontal holes.

Let us start by unfolding E, T and the longer (horizontal - top and bottom) faces of holes in the same way as in Section 1.5. The Figures 1.13 and 1.14 show the first two steps of the algorithm.

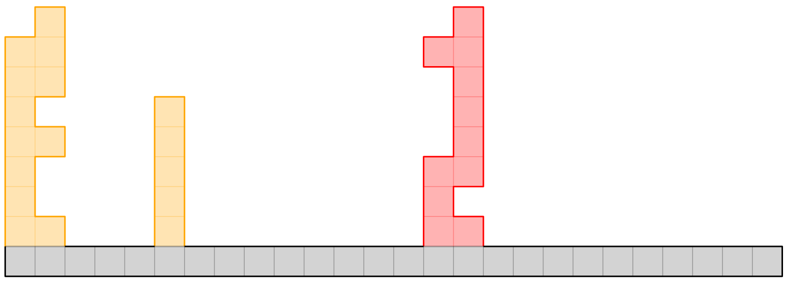


Figure 1.13: The first step of unfolding the polycube in Figure 1.12.

It remains to unfold the bottom base B and the short (left and right) faces of the holes. We cannot do that in the same way as before, because if we cut B into stripes of width 2, they would not necessarily be incident to the external boundary. We can instead connect one face of each hole $h \in \mathcal{H}$ to one of the two already unfolded faces of I_h. In case of holes in the orange stripes, we unfold the right face, in case of holes in the red stripes, we unfold the left face. Let us look at the already unfolded horizontal faces of I_h. One of the faces is unfolded "inside" of a stripe, but the other is "outside". For example, consider a hole in the lower row of a red stripe: the top face of this hole is unfolded "inside" the red

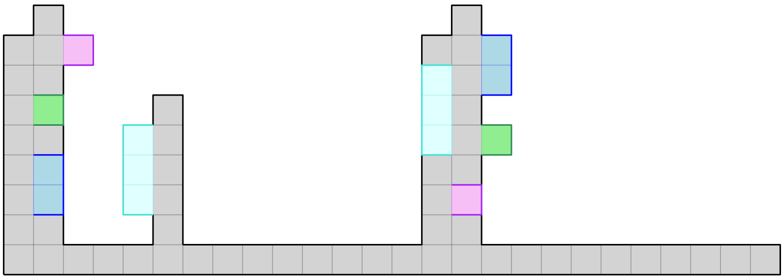


Figure 1.14: The second step of unfolding the polycube in Figure 1.12.

stripe while the bottom face is unfolded "outside" of an orange stripe. The face unfolded outside has empty space around it and we can connect the one face of I_h here (this face is only one 1×1 square). There cannot be an overlap - we are outside a stripe, so there could only be a face of some hole or external boundary. External boundary cannot be there because it has distance at least 1 from all holes (and it also lies in the opposite direction than the one in which we place the face). The same is true for holes, they are at a distance of at least 1 from each other, so the unfolded longer faces are not next to each other. Two faces unfolded in this step cannot overlap either because they are unfolded in the same direction.

The last part is the bottom base B and exactly one face of every hole. This is rather simple since all the remaining faces are just 1×1 squares. We can unfold the remaining 1×1 faces and B in a similar fashion to unfolding wide holes 1.4. See the Figure 1.15 for an example of the last steps.

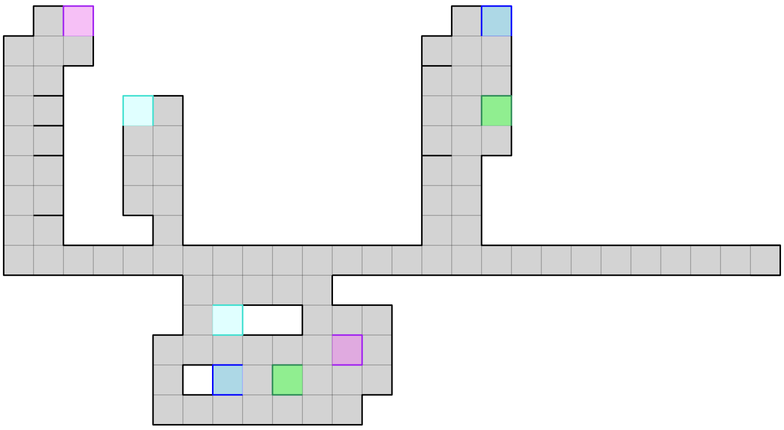


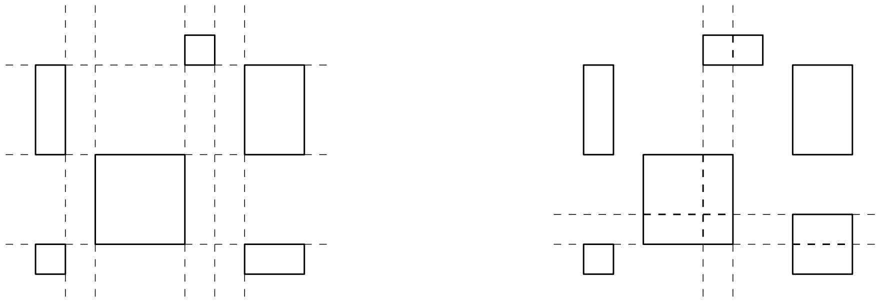
Figure 1.15: Unfolding of the polycube in Figure 1.12.

1.7 Separable rectangular holes

A slightly more general class of one-layer polycubes than the polycubes with cubic holes can also be unfolded using a similar algorithm.

Definition. We call a set of rectangles *separable* if it satisfies the following property. If we extend any edge of any rectangle to a line, it does not cut any other rectangle (it does not contain an interior point of any other rectangle).

See Figure 1.16 for an example of separable rectangles. One-layer polycubes whose holes are separable rectangles can be unfolded using an algorithm very similar to the one in Section 1.5. Note that cubic holes are trivially separable, thus one-layer polycubes with cubic holes can also be unfolded using this algorithm.



(a) Example of separable rectangles. (b) Example of **non**-separable rectangles.

Figure 1.16: Examples of separable and non-separable rectangles.

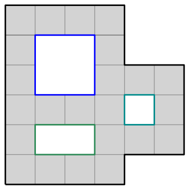


Figure 1.17: Example of a one-layer polycube with separable rectangular holes.

Let us extend the edges of all rectangles that are parallel to x-axis to lines. This creates several horizontal stripes. Analogously, we can create vertical stripes. Instead of coloring pairs of neighboring rows or columns of T and B as in Section 1.5, we color pairs of neighboring horizontal stripes. You can see an example of such coloring in the Figure 1.18.

(a) Coloring of the top base. (b) Coloring of the bottom base.

Figure 1.18: Coloring of the polycube in Figure 1.17.

The rest of the algorithm is the same as in Section 1.5. We consider connected components of both colors. The leftmost and rightmost squares of connected components are incident to E and will be connected on the left or right side depending on their color. The distance of any pair of stripes is again at least 2 for the same reasons as in the algorithm for cubic holes. We omit the case analysis this time. We then unfold the horizontal and vertical faces of internal boundaries separately. Since the distance between neighboring stripes is at least 2, there are no overlaps. Figures 1.19 and 1.20 show the steps of this algorithm.

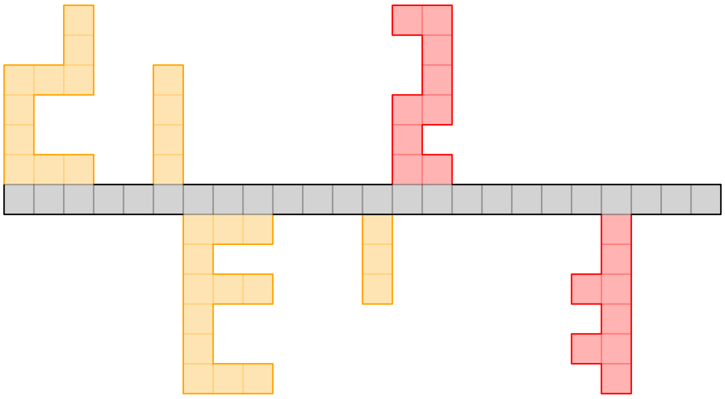


Figure 1.19: The first step of unfolding of the polycube in Figure 1.17.

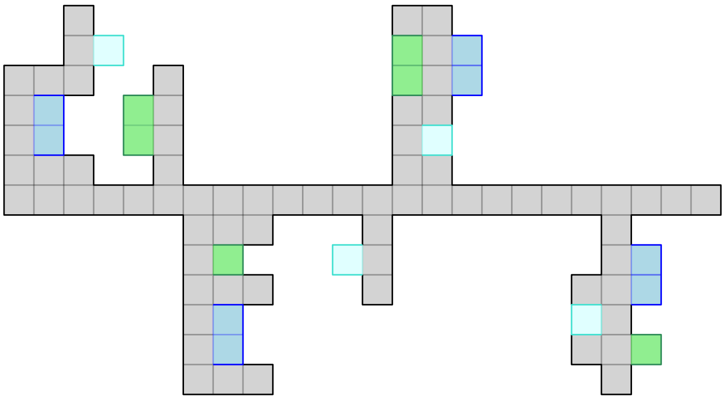


Figure 1.20: Unfolding of the polycube in Figure 1.17.

1.8 Orthotrees

Definition. Polycube $\mathcal{P}$ is called an *orthotree* if the cube graph $C(\mathcal{P})$ is a tree.

The figures in this section do not contain boundaries between squares. This way it is easier to see which squares were connected in the polycube before unfolding.

Although the question of unfolding general orthotrees is open, it is quite easy to unfold a one-layer orthotree. Since one-layer orthotrees have no holes, they can be unfolded using the algorithm presented in Section 1.3, you can see this in Figure 1.21. However, we can do something stronger. It is possible to unfold any one-layer orthotree into a stripe of height 3. In other words, the difference of y-coordinate of the highest and lowest points of the resulting polygon will be exactly 3.

The algorithm starts, as always, by unfolding the external boundary E into a single horizontal stripe. We will only focus on the top base T, the bottom base B can be unfolded symmetrically. Almost all squares of the top base are incident to the external boundary. Let us connect each of them to one of the squares of the already unfolded external boundary. The square of E to which the square of T will be connected is chosen as the first available in this order: left, bottom, top, right. It remains to unfold the squares not incident to the external boundary, these are the top faces the of cubes with degree 4 in $C(\mathcal{P})$. Each will be connected to its right neighbor. It is easy to see that the resulting unfolding

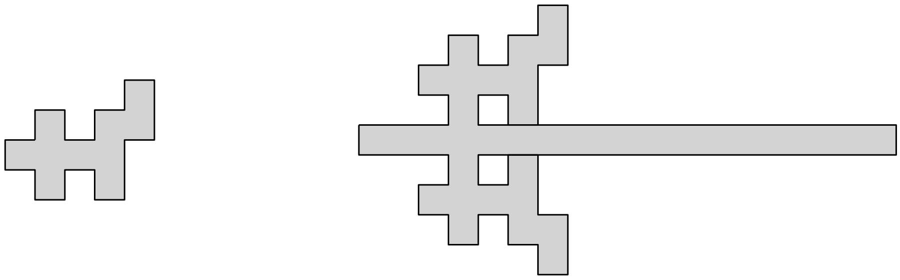


Figure 1.21: Example of a one-layer orthotree and unfolding produced by the algorithm from Section 1.3.

will have height exactly 3. See Figure 1.22 for an example.

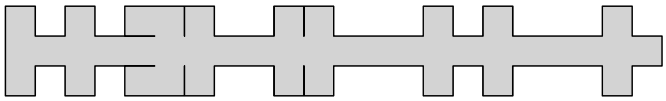


Figure 1.22: Unfolding of the polycube in Figure 1.21 into a stripe of height 3. We started at the top-left corner of the polycube.

Let us show that there is no overlap. The only squares that could overlap with something are the ones which are not incident to E. Consider one such square at the coordinates $[x, y]$ before unfolding. This square is connected to its right neighbor, which has the coordinates $[x+1, y]$. There must be no square at the coordinates $[x+1, y-1]$, since the polycube is supposed to be an orthotree. Hence, the square at the coordinates $[x+1, y]$ is connected to E by its bottom edge. The only possible overlap may occur with the square at coordinates $[x, y-1]$ if it were connected to E by its right edge. However, there is no square at the coordinates $[x-1, y-1]$, and thus the square at the coordinates $[x, y-1]$ is connected to E by its left edge and there is no overlap. The situation is depicted in Figure 1.23.

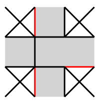


Figure 1.23: If there is a square not incident to E on coordinates $[x, y]$ (middle of the cross), there are no squares on coordinates $[x-1, y-1]$ and $[x+1, y-1]$. The edges connecting squares of T to the boundary E in the unfolding are highlighted in red. The squares without cubes are crossed out.

Note that not all one-layer polycubes are unfoldable into a stripe of height 3. An example of such a polycube is a one-layer polycube whose base is a square 7×7, let us denote it by S_7.

Theorem 7. *It is impossible to unfold the polycube S_7 into an orthogonal polygon of height at most 3.*

Suppose that there exists an unfolding of S_7 into a stripe of height 3. Let m be the middle square of the top base. Let H denote the horizontal faces of E and let V denote the vertical ones. Let us say that a square *touches* H if it shares an edge with a square of H or is contained in H itself; analogously for V.

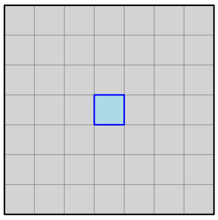


Figure 1.24: The top face of S_7 with the middle square highlighted.

Now consider the subsets of the boundary of S_7 with the following property. The subset contains m, is connected after cutting and contains at least one square touching V and at least one square touching H. Let us call the smallest such subset C. C must contain at most one square that touches V or at most one square that touches H. If C contained at least two squares touching H and V, it could not be the smallest one with the property mentioned. Suppose that C contains at least two squares touching H and V. Since C is connected, we can consider a spanning tree $S \subseteq F(S_7)$ on the squares of C. We can remove one of the leaves different from m of the spanning tree S (the tree obviously has at least two leaves) from C and it will still satisfy all the required properties, a contradiction.

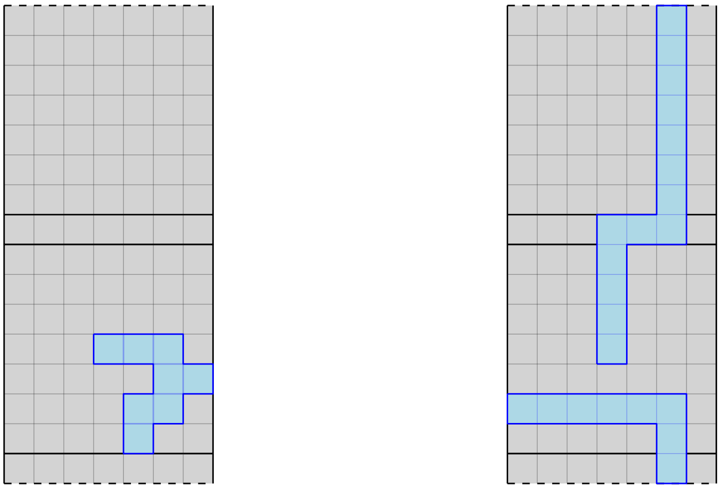


Figure 1.25: Examples of what can C look like on the band. The dashed lines are connected to each other.

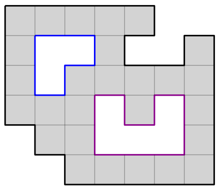
Let us without loss of generality assume that C contains only one square touching V. This means that C does not contain any square of V, if it did, it would have contained at least two squares touching V. C must therefore contain only squares from T, B and H. T, B and H together form a cyclic band of width 7. C must contain the middle square of T, exactly one square at the boundary of the band, and at least one square at the top or bottom of T. C has both height and width at least 4 in the band. The unfolding from the band to the plane does not change width at all. The height after the unfolding clearly is at least 4, we can consider the shortest path from m the the top or bottom edge of T. See examples of C in Figure 1.25.

1.9 Y-convex holes

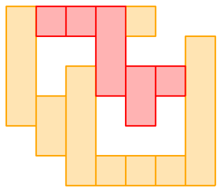
Definition. A set of points in the plane is called *y-convex* if its intersection with any vertical line forms a line segment.

In this section, we will use slightly different methods to unfold polycubes with y-convex holes that contain at most one hole (possibly consisting of multiple squares) in every column. See Figure 1.26. Our approach will be slightly similar to the algorithm used by Kiou, Poon, and Wei [6].

Let us start by unfolding the top base T, the bottom base B will be unfolded in the same way. We will color the top base orange and red in the following way. We consider connected components in every column. Every component is incident to E either at the top or at the bottom because the holes are y-convex and there is at most one hole in every column. We will color the components incident to E at the bottom orange and the remaining components red (all of them are incident to E at the top). See Figure 1.26 for an example of this coloring.



(a) Example of a one-layer polycube with y-convex holes and with at most one hole in every column.



(b) Colored components in the top base.

Figure 1.26: Polycube and coloring of the top base.

The algorithm starts by unfolding E, T and B; the internal boundaries will be connected later. We start by unfolding E to a horizontal stripe. Every orange connected component in every column will be connected to E by its bottom and every red component by its top. The result is depicted in Figure 1.27.

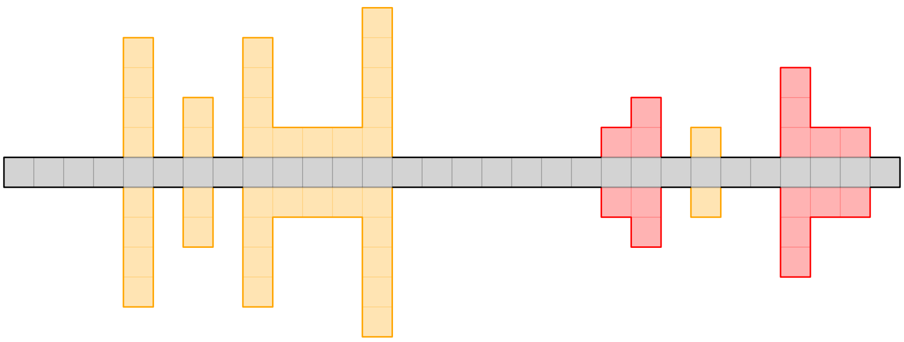


Figure 1.27: The first step of unfolding the polycube in Figure 1.26.

In the second step, we unfold the internal boundaries $\mathcal{I}$ of all holes. Every internal boundary will be unfolded to a single stripe of width 1. It will be connected to T by the edges of its leftmost column. The unfolded stripe will go from the place where it is connected in the direction away from the unfolded E. See Figure 1.27.

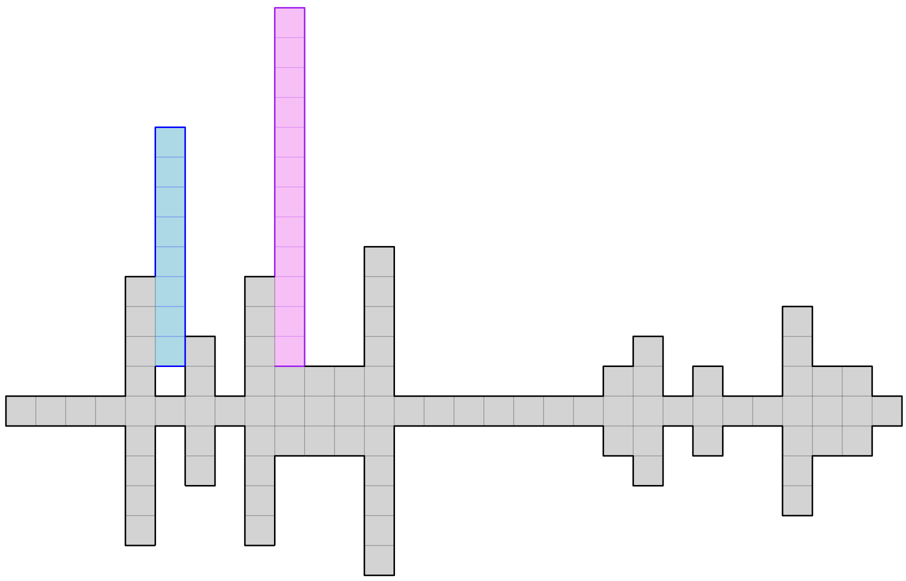
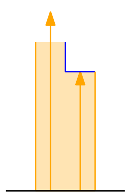


Figure 1.28: The unfolding of the polycube in Figure 1.26.

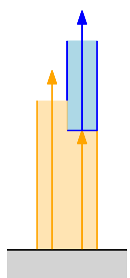
It remains to show that there is no overlap. The only parts that could overlap with something are the internal boundaries. Every unfolded internal boundary is connected to either an orange column to the left of it or to a red column to the right of it. Consider some hole h that has orange column to the left of it. The only part of the top base, which could overlap with the internal boundary I_h, is the leftmost orange column under h.

Let us look at the two neighboring orange columns — the one to the left of h (to which I_h will be connected) and the leftmost one under h. There are 2 cases:

- The bottoms of the two columns are at the same height. In this case, there will be no overlap because the right column must be shorter that the left one (it is blocked from above by the hole h). See Figure 1.29.
- The bottoms of the two columns are at different heights. There will again be no overlap since there will be a distance of at least one between the columns after unfolding. There will be enough space between those two columns to unfold I_h. See Figure 1.30.

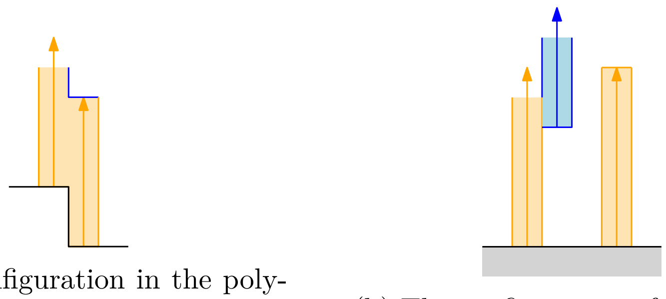


(a) The configuration in the polycube.



(b) The configuration after unfolding, there is no overlap.

Figure 1.29: The case when the bottom edges of both columns are at the same height.



(a) The configuration in the polycube.

(b) The configuration after unfolding, there is no overlap.

Figure 1.30: The case when the bottom edges of both columns are at different heights.

For an overlap of two inner boundaries to exist, there would have to be an orange and a red column very close to each other after the unfolding. In particular, there would have to be a red column at distance at most 1 to the right from an orange column. It is quite easy to see that this is impossible.

The same case analysis shows that the internal boundaries connected to the red columns can not overlap with anything either.

2. Multi-layer polycubes

We briefly mention algorithms for unfolding some classes of multi-layer polycubes. Let us first define an *overhang*.

Definition. An *overhang* of size k is the following formation of cubes. There are $k+1$ cubes in a row with with the same z coordinate. Under the first k of these cubes, there is an empty space. Under the last one, there is another cube. There is also an empty space next to the first cube of the row.

See Figure 2.1 for an example of an overhang. Overhangs will be useful for description of the classes we are able to unfold.

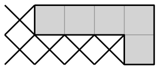


Figure 2.1: Example an overhang of size 3. We are looking at the polycube from a side (in the direction of the x-axis). Places that must be empty are crossed out.

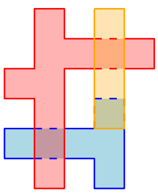
2.1 Treestacks

Let us repeat the definition from the introduction.

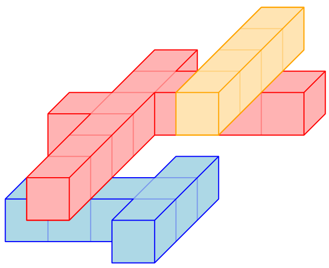
Definition. A *treestack* is a polycube whose every layer (set of cubes with the same z-coordinate) forms an orthotree.

Unfortunately, we need to place some restrictions on the treestacks we will be unfolding.

Theorem 8. *There exists an edge-unfolding of every treestack that contains an overhang of size at least 2 in every layer (except the bottom one) and in every layer every cube has at most 3 neighbors.*



(a) The treestack viewed from above.



(b) The treestack as a 3-dimensional object.

Figure 2.2: Example of a treestack with three layers and an overhang of size at least 2 in every layer.

To unfold treestacks with overhang of at least 2 in every layer and degrees at most 3, we will use the algorithm presented in Section 1.8 as a subroutine. In the first step of the algorithm, we unfold the external boundaries of all layers and connect them using the bottom parts of the overhangs; you can see the result in Figure 2.3. There will be at least two rows between every pair of unfolded external boundaries. The rest can be unfolded in the same way as in Section 1.8. Some squares of the top and bottom bases might be missing, but this is not an issue, it actually makes the unfolding easier. Thanks to the absence of cubes with four neighbors, there is no overlap. If there were some squares with 4 neighbors, they could not always be connected to their right neighbor as in Section 1.8 since the neighbor could be missing. See an example of the unfolding in Figure 2.4.

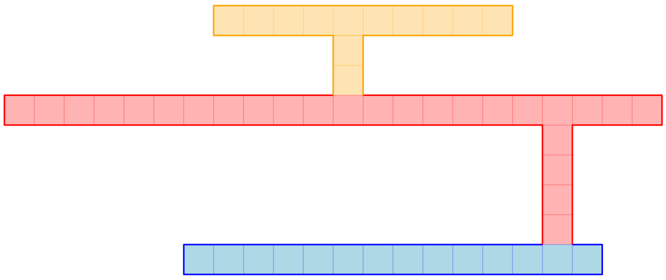


Figure 2.3: The first step of unfolding the treestack in Figure 2.2.

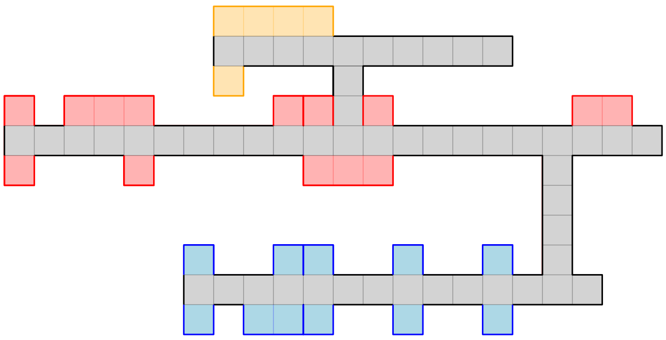


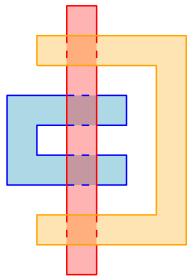
Figure 2.4: Unfolding of the treestack in Figure 2.2.

2.2 Pathstacks

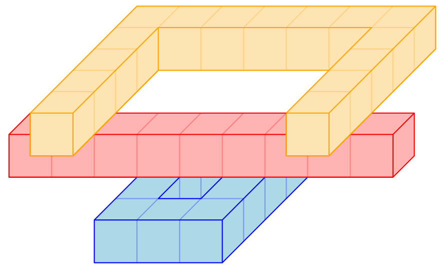
Definition. A *pathstack* is a polycube whose every layer forms an orthopath.

Theorem 9. *There exists an edge-unfolding of every pathstack that contains an overhang of size at least 1 in every layer.*

The algorithm for unfolding a pathstack with an overhang of size at least one in every layer is similar to the one in Section 2.1. We unfold the external boundaries and connect them by the overhangs as before; see Figure 2.6. The difference is that the distance between the unfolded external boundaries could be just one. The top and bottom faces of neighboring layers have to be unfolded into the stripe of height one. This issue can be solved quite easily thanks to the layers being paths.



(a) The pathstack viewed from above.



(b) The pathstack as a 3-dimensional object.

Figure 2.5: Example of a pathstack with three layers and overhang of size at least 1 in every layer.

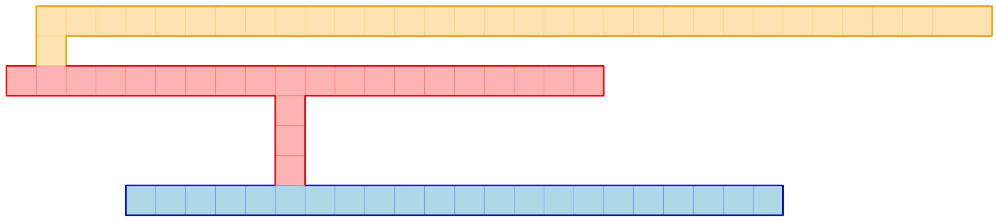


Figure 2.6: The first step of unfolding the pathstack in Figure 2.5.

Every square in the top or bottom base shares at least two edges with the external boundary; hence, there are at least two squares in the plane to which the square can be unfolded. Every square in the plane shares at most two edges with the unfolded external boundaries, and hence there are at most two squares from the top and bottom bases that could be unfolded here. The existence of the unfolding follows from Hall's marriage theorem. We can consider a bipartite graph whose first part will be the squares in the plane and the other part will be the squares of the top and the bottom base of the neighboring layers. There will be an edge between two vertices if the corresponding square from the surface of the polycube can be placed to the corresponding square in the plane. The degrees of all vertices in the first part are at most two and the degrees of all vertices in the second part are at least two. This graph satisfies the Hall's condition and therefore there exists a matching that covers the second part and the polycube can thus be unfolded.

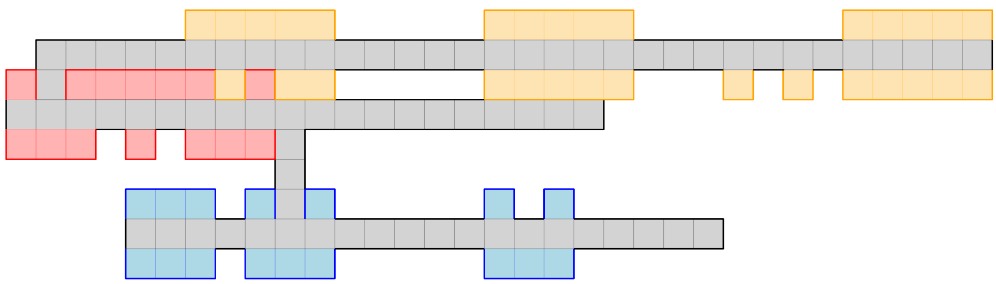


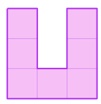
Figure 2.7: Unfolding of the pathstack in Figure 2.5.

The description of the class of pathstacks we are able to unfold is quite strange. Perhaps the following formulation may be more useful.

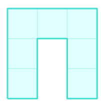
Theorem 10. *There exists an edge-unfolding of every pathstack if at least one of the ends of every path has an empty square under it.*

It is easy to show that the condition above is sufficient to force the existence of an overhang in every layer. We can simply start at the free end of the path and go along the path until there is a cube under us (this will always happen unless this path is the bottom one). At that point, the last straight segment of the path forms an overhang.

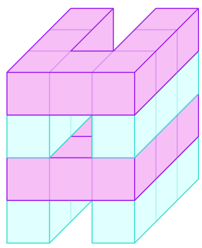
Note that there exist pathstacks without an overhang. One such example are identical paths stacked on each other. This case can be solved quite easily by simply making the external boundary taller. There are, however, some examples which are more difficult to deal with. One of them can be described as paths in the shape of the letter U where every other layer is rotated by 180 degrees. You can see this pathstack in Figure 2.8.



(a) Even layers have shape of the letter U.



(b) Odd layers have shape of the letter U upside down.



(c) The pathstack with 4 layers.

Figure 2.8: Example of a pathstack without overhangs.

Conclusion

We presented several linear-time algorithms for edge-unfolding of special cases of one-layer polycubes. The question of unfolding one-layer polycubes with arbitrary holes remains open. Interestingly, we are able to unfold one-layer polycubes with very small (cubic) holes and very large (wide) holes. These are, in some sense, opposite types of one-layer polycubes. generalizing our approach to unfold other classes of one-layer polycubes seems rather difficult since it relies on being able to cut the top and bottom faces into stripes such that all the connected components are incident to the external boundary.

We also showed how to unfold some classes of multi-layer polycubes. It might be possible to generalize this approach to unfold, for example, general pathstacks without any restrictions.

Bibliography

[1] Therese Biedl, Erik Demaine, Martin Demaine, Anna Lubiw, Mark Overmars, Joseph O'Rourke, Steven Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. *Proc. 10th Canad. Conf. Comput. Geom.*, 01 1998.

[2] E.W. Chambers, K.A. Sykes, and C.M. Traub. Unfolding rectangle-faced orthostacks. *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012*, pages 23–28, 01 2012.

[3] Yi-Jun Chang and Hsu-Chun Yen. Improved algorithms for grid-unfolding orthogonal polyhedra. *Internat. J. Comput. Geom. Appl.*, 27(1-2):33–56, 2017.

[4] Mirela Damian, Robin Flatland, Henk Meijer, and Joseph O'Rourke. Unfolding well-separated orthotrees. pages 23–25, 2005.

[5] Mirela Damian, Robin Flatland, and Joseph O'Rourke. Unfolding Manhattan towers. *Comput. Geom.*, 40(2):102–114, 2008.

[6] Meng-Huan Liou, Sheung-Hung Poon, and Yu-Jie Wei. On edge-unfolding one-layer lattice polyhedra with cubic holes. *Computing and combinatorics*, 8591:251–262, 2014.

[7] Joseph O'Rourke. Unfolding orthogonal polyhedra. *Surveys on discrete and computational geometry*, 453:307–317, 2008.

[8] Joseph O'Rourke. Unfolding polyhedra. 08 2008. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.8196.

[9] Joseph O'Rourke. Unfolding polyhedra. *Proc. 31st Canad. Conf. Comput. Geom.*, 2019.

[10] Lydie Richaume, Eric Andres, Gaëlle Largeteau-Skapin, and Rita Zrour. Unfolding level 1 Menger polycubes of arbitrary size with help of outer faces. In *Discrete geometry for computer imagery*, volume 11414 of *Lecture Notes in Comput. Sci.*, pages 457–468. Springer, Cham, 2019.