How to draw combinatorial maps?

Dainis Zeps, Paulis Ķikusts *

Abstract

In this article we consider the combinatorial map (rendered by permutations) approach to graphs on surfaces and how between both could be establish some terminological uniformity in favor of combinatorial maps in the way rotations were set as fundamental structural elements, and other necessary notions were derived from them. We call this the rotational prevalence with respect to how to build a graph drawing environment. We deal here with simple operations of how to draw combinatorial maps and partial maps. One of our aims would be to advocate a wider use of combinatorial maps in the graph drawing applications. Besides, we advocate to use corners of halfedges where upon permutations act in place of halfedges.

1 Introduction

We deal in this article with some simple considerations and observations of how to draw combinatorial maps, and how it comes in connection with traditional drawing of graphs, fig. 1, see fig. 4 in 3.1 too.

On the right side of fig. 1 the drawing is performed by a program designed by the second author. Graph drawing traditions in Riga started yet in the seventies, when among other places we used to visit every year the Odessa graph seminar guided by the celebrated graph theorist A.A.Zykov. Already then Emanuels Grinbergs, Jānis Dambītis and Šneors Berezins from Riga were talking their results at Odessa seminar. We presented the graph drawing experience of later years in articles [13, 5, 6].

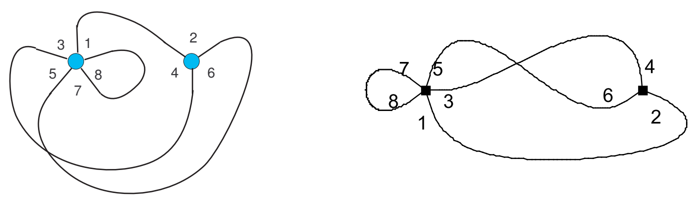

Figure 1: Two drawings of the same combinatorial map.

Combinatorial map theory is a new area of combinatorics the development of which may have a considerable impact on topological graph theory. Besides, combinatorial functions around these maps are both theoretical tools and effectively calculable means that could be developed in complex environments for applications. These directions, as it seems to us, are weakly developed, partially maybe due to the fact that combinatorial maps and apparatus around them as if repeat all what is already presented in topological

*Authors' address: Institute of Mathematics and Computer Science, University of Latvia, 29 Rainis blvd., Riga, Latvia, dainize@mii.lu.lv, paulis@mii.lu.lv

graph theory. But combinatorial maps as a branch of combinatorics may be investigated independently from graph theory using e.g. such simple tools as permutations, and use whatever results already achieved in e.g. permutational group theory. Astonishing results of the application of constellations (what is some generalization of combinatorial maps) in the Riemann geometry [14] show that this area deserves to be researched with much greater efforts than before.

To draw either a graph or a combinatorial map doesn't much differ if both require to come to the picture of points and lines confining areas in the plane. Combinatorial map's approach suggests its own natural way how to come to its image and we try to follow directly this line.

The invention of combinatorial maps is attributed to Tutte [23, 24], their theory was developed further by Stahl [20, 21, 22] and many other researchers, see e.g. [15, 3, 16]. The combinatorial map approach is based on the discovery of rotations in graphs on surfaces, see Heffter [9, 10], Edmonds [4] and Jacques [11]. These facts were researched by many, see e.g. [7, 25, 18, 17]. One way to introduce a combinatorial map is by a pair of permutations, see [3, 14]. We do researches in this area since 1993 [26, 12, 27, 28, 29, 30, 31, 32, 34, 35].

Specifically, combinatorial maps as a purely combinatorial structure could deserve a wider application in graph theory and some of its applications such as e.g. graph drawing, where combinatorial maps could serve as a more fundamental structure than graphs themselves. Rotations could be the combinatorial objects on which all other graph-theoretical invariants could be represented. We aim in this article to advocate a wider use of combinatorial maps in graph theory, and graph drawings particularly. By the way we advocate the use of *corners* as elements on which the permutations should act, see subsection 3.3.

We endeavored to built combinatorial map drawing tools where we used directly the combinatorial map approach to depict its eventual visual image, see fig. 4 in 3.2. At the same time we give account to ourselves that problems typical for the graph drawing area arise here too, because combinatorial map's image, that we are to draw, is a picture in the plane of the same nature as that of a graph.

By developing tools for the drawing of combinatorial map and by giving ways to the visualization of combinatorial maps we hope to widen the use of them in topological graph theory.

2 From graph on surface to set of rotations in the graph

For the purposes of this paper we define a combinatorial map as an arbitrary pair of permutations.

A combinatorial map as a purely combinatorial object naturally models a graph (in general, a hypergraph) embedded in an oriented surface. By the discovery of Heffter [9, 10] and the rediscovery of Edmonds [4] we know that the vertex rotation, i.e., the rotational order of edges around a vertex, taken for all vertices, fixes the graph on some orientable surface. But,if we fix the rotational order of edges in faces too, than these two rotations, the vertex rotation and the face rotation, are sufficient to code the whole graph, i.e., no sets of vertices and edges are to be specified, because these two rotations already determine the graph on some orientable surface.

Moreover, coding these rotations with permutations and performing operations purely with permutations, we may maintain all operations upon a graph in the same permutational way, or as close as possible, that usually is done in the vertices-edges-faces operational framework.

There is a very natural way to come from vertices and edges to the permutational setting. Let the graph $G = (V, E)$ be fixed by a vertex rotation. If we depict each unoriented edge as a pair of oriented edges in a way that the outgoing edge comes first before the incoming edge in the rotation of edges around vertices in the clockwise direction, we observe that the oriented edges around faces are oriented all in one direction, and this direction is anticlockwise. See fig. 2 left. Now we are to make one more abstract operation – we replace a corner of the face that was formed by two oriented edges by a pair of halfedges, namely, the pair of the incoming head and the outgoing tale of the oriented edges, and call this new object a *corner*. By the way, doing this, we trivially, replace the double cover of borders of faces with corners of oriented edges by the simple cover of corners of halfedges. See fig. 2.

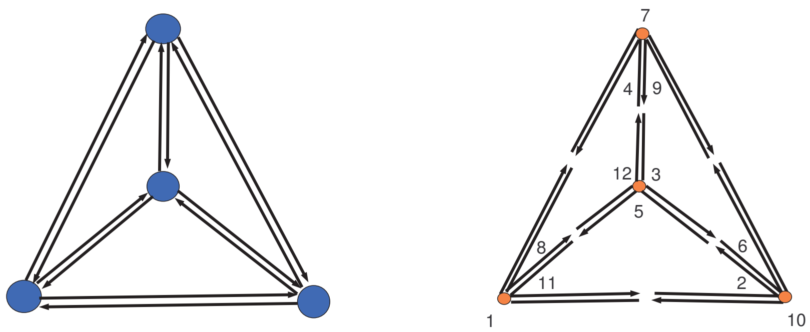

Figure 2: An embedding of the graph K_4 in the plane fixed by a rotation of alternatively outgoing and incoming edges around the vertices – left, and by a vertex rotation and a face rotation of corners of halfedges – right. By numbering corners we come to permutations which stand for rotations. Cyclical order of corners around the vertices are represented by the permutation $(18\bar{1})(26\bar{0})(35\bar{2})(479)$ and around the faces by the permutation $(17\bar{0})(25\bar{1})(369)(48\bar{2})$. Two edge rotations are respectively $(12)(34)(56)(78)(9\bar{0})(\bar{1}\bar{2})$ and $(14)(23)(58)(67)(9\bar{2})(\bar{0}\bar{1})$. Here we use the Grinbergs' notation for 10, 11, ... with bars, $\bar{0}$, $\bar{1}$,...

There are as many corners of halfedges as oriented edges in the graph, namely, double as many as edges in the graph. Of course, these corners (as corners of halfedges) cover all oriented edges only once. Now, by fixing both the vertex and the face rotations we actually fix two corner rotations, respectively around the vertices and faces. The size of both rotations is the same, this is the number of corners. Actually two more rotations of the same size appear, namely, two edge rotations, which we get if we establish the new adjacency of corners across what former was the edge. Edge rotations in the graph have only 2-cycles, thus corresponding permutations are involutions. In the case of hypergraphs the edge rotations may have orbits of an arbitrary size, namely, corresponding to the size of an hyperedge.

Now, in the case of graphs with possibly hyperedges we also may get along with four rotations of an equal size that fix this graph on a surface without any necessary additional information. But, the four rotations are redundant. We actually need only two rotations to specify a graph or a multigraph on a surface. Both edge rotations have the same cycle structure, and they may be calculated from other two rotations. Besides, taking only three rotations, but all in one direction, say, anticlockwise, the product of them is equal to the identity permutation, [14]. (For that reason, any combinatorial map is a 3-constellation.) This means that the third permutation always may be calculated from the two given. Now, in the case of graphs without hyperedges, edge rotations are involutions without fixed points with respect to the set of corners they act upon. If we fix an edge rotation once and for ever, because it is an involution without fixed points, we may calculate all permutations from only one given permutation. This wouldn't work for an edge rotation, which were to fix not a particular graph but some larger class[28].

2.1 From operations with graphs on surfaces to a permutational calculus

With having the corners (C) introduced above we directly and naturally come to a permutational calculus. There are as many corners (of halfedges) as oriented edges in the graph, so the number of the corners $\|C\|(= m)$ is equal to $2\|E_G\|$, therefore for the graphs without hyperedges this number is always even. Thus, we may label the corners (or equate in practice) with natural numbers from the interval $[1..m]$. We have four permutations corresponding to the four rotations of the same size m. By the way, the identity permutation of degree m as the vertex rotation should correspond to the graph consisting from $m/2$ isolated edges.

2.2 Hypergraphs and pairs of permutations

By the way, in case of graphs we should have only even permutations, but in the case of hypergraphs arbitrary permutations may arise. To fix a hypergraph on a surface we need three rotations, or, by fixing a hyperedge rotation, both a vertex and a face rotation. In general, two permutations always have some fixed hypergraph in correspondence [29]. In [29] we call combinatorial maps corresponding to hypergraphs *partial maps* as opposing to *graphical maps* corresponding to graphs without hyperedges. The name of partial map is motivated by the fact that a partial map may be considered as a map with some faces being cut out, thus being as if partial with respect to the graphical map with all the faces present.

See in fig. 3 an example of a hypergraph, graphs K_4 embedding on the torus with a face being cut out. This face stands for the hyperedge of degree four. That we are to do with the hypergraph we get to know from the fact that the edge rotation is not graphical.

3 Graphical combinatorial maps introduced

Now we are ready to start a combinatorial map calculus as a permutational calculus where permutations act on the set C of corners.

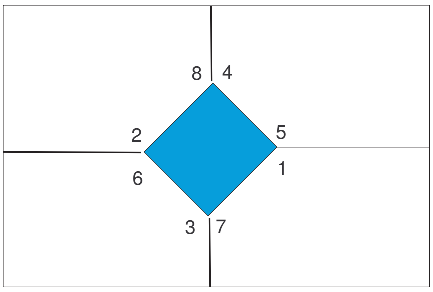

Figure 3: An example of a partial map with the vertex rotation $P = (15)(26)(37)(48)$, the face rotation $Q = (17452896)$ and the edge rotation $R = (1423)(56)(78)$. One of edges, (1423), is not graphical but is an hyperedge. See the covers of two books [14] and [33].

We multiply permutations from left to right. Graphical *combinatorial map* is a pair of permutations, (P, Q), with $P \cdot Q^{-1} (= \rho)$ being involution (i.e., multiplied by itself giving identity permutation) without fixed elements. P and Q are correspondingly *vertex rotation* and *face rotation*. We distinguish two edge rotations, *inner edge rotation* $\pi = Q^{-1}P$ and *(outer) edge rotation* $\rho = P \cdot Q^{-1}$. We assume permutations acting on the set C, usually natural numbers from 1 to m, $m = \|C\|$. We try to confine ourselves with maps with fixed π (for all the class of the maps) calling them *normalized maps*, and mostly we use one particular choice of π equal to (12) ... $(2k-1\ 2k)$, $k \geq 1$. Under these assumptions, the graphical combinatorial map is characterized by one permutation or one rotation, either vertex rotation P, or face rotation Q.

3.1 Drawing graphs on surfaces

A graph with loops and multiedges on an orientable surface corresponds to an arbitrary graphical combinatorial map in a very natural way. One intuitively well based way to persuade oneself about this is to draw directly this graph in the plane in the following way. Let us first put as many points in the plane as orbits in the vertex rotation with *edge-ends* clockwise around them with corners in between from particular orbit following in the same cyclical order, see fig.1, a. Further, let us unite two edge-ends with corners (following clockwise) a and $b (= a^{\pi})$.(An edge establishes between the corners with edge adjacency between them, see 3.3.) That means that corners that form orbit of inner edge rotation π are to be united with an edge in the drawing, thus justifying the choice to speak about *an edge of the combinatorial map*. It is easy to see that changing the orientation of clockwise to anticlockwise we had to use the edge rotation ρ in place of the inner edge rotation π. Thus, two opposite directions of rotations give two opposite, inner and outer, edge rotations.

See e.g. fig. 4 where a combinatorial map $P = (18753)(264)$ is drawn. In a) we place points in the plane corresponding to the orbits in the vertex rotation. In b) to d) to get the drawing of the combinatorial map, we are to unite the pairs of edge-ends $(2k-1, 2k)$ with curves for the edges from $k = 1$ to 4. So, in b) we unite the edge-ends with close by corners 1 and 2. Notice the order of the edge-end and the close by corner in the clockwise direction. In d) we have the drawing fulfilled. Similarly we unite the edge-ends with the close by corners 3 and 4, and in d) we unite two left edge-end pairs. Notice that loops

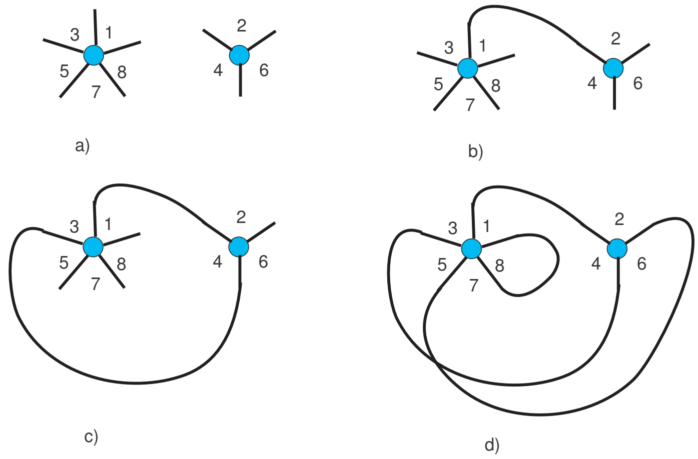

Figure 4: a) Putting two points in the plane with edge-ends and corresponding corners from orbits of the given permutation (18753)(264); b) uniting the edge-ends with close by corners 1 and 2. Notice the order of the edge-end and the close by corner in the clockwise direction. c) Uniting the edge-ends with the close by corners 3 and 4; d) uniting two left edge-end pairs. Notice that the last "edge" of the map is a loop.

correspond to the edge-ends that go out from the same point in the plane.

Further examples of combinatorial map drawings performed with an implemented map drawer see in the section 4.1.

3.2 Definition of the edge of combinatorial map

Let us define the edge of combinatorial map formally.

Let Ω_p and ω_p be correspondingly the set of orbits and some orbit of permutation p.

Definition 1. *Let for a given combinatorial map* P, Q, π, ρ *the sextet* $e = \langle \omega_P^1, \omega_P^2; a, b, c, d \rangle$ *be such that* $a, b \in \omega_P^1$, $c, d \in \omega_P^2$, $(ac) \in \Omega_\pi$, $(bd) \in \Omega_\rho$ *and* $(ac)^P = (bd)$*. We call* e *an edge of this combinatorial map.*

Let us use for a single element a small letter, say, a, and for a (cyclical) (possibly empty) sequence of elements a capital letter, say A. In this way aA may denote some orbit of a permutation in cyclical form with corner a being distinguished and rest part (possibly empty) being A.

Using this convention we may write an edge of a map in the form

$$\langle baA, dcC; a(= c^\pi), b(= d^\rho), c(= a^\pi), d(= b^\rho) \rangle.$$

Let us fix this as a Proposition.

Proposition 2. *Orbits* (ac) *and* (bd) *correspondingly of inner edge rotation* π *and edge rotation* ρ *have an edge*

$$\langle baA, dcC; a(= c^\pi), b(= d^\rho), c(= a^\pi), d(= b^\rho) \rangle$$

of a combinatorial map in correspondence with corresponding equalities being held.

We must observe that the edge we speak about turns into a loop in case orbits baA and dcC coincide. But the corners a, b, c, d all should be distinct nevertheless except in cases of isolated or hanging edges or loops. So isolated edges or loops have two distinct corners, $\langle a, b; a, a, b, b\rangle$ being an isolated edge, and $\langle ab, ab; a, b, b, a\rangle$ being an isolated loop. Hanging edges or loops have three distinct corners, e.g., $\langle baA, c; a, b, c, c\rangle$ being an hanging edge, and $\langle baDc, cbaD; a, b, b, c\rangle$ being an hanging loop. One more even simpler combinatorial object, that ceases to be geometrical combinatorial map, is an isolated loop with cut out one face $\langle a, a; a, a, a, a\rangle$.

We may sum up these relations by stating a following Corollary.

Corollary 3. *For an edge $\langle baA, dcC; a, b, c, d\rangle$ of a given combinatorial map P, Q, π, ρ coincidence $c = d$ gives an hanging edge with C being empty string, and coincidence $b = c$ gives an hanging loop with equalities $A = Dd$ and $C = aD$.*

As an example, for a normalized map $P = (18753)(264)$ $e = \langle(18753), (264); 1, 3, 2, 4\rangle$ is an edge, comp. fig. 4. In case we don't want to specify the rotations where from the edges go out we use a shorter form, the quartet notation only with four corners, e.g., we write for the edge $e = \langle 1, 3, 2, 4\rangle$.

It is easy to show that the full set of edges in quartet notation specifies the combinatorial map uniquely. Indeed, let us write m edges as $4 \times m$ matrix $E_{i,j}$ with i-th row set as a, b, c, d for i-th edge. Then the two mappings $E_{i,2} \to E_{i,1}$ and $E_{i,4} \to E_{i,3}$ taken together produce an automorphism of the set of corners that actually is the vertex rotation P. Similarly the pair of mappings $E_{i,2} \to E_{i,3}$ and $E_{i,4} \to E_{i,1}$ taken together produce another automorphism of the set of corners, the face rotation Q. We sum up this as a proposition.

Proposition 4. *m edges of a combinatorial map in the quartet notation uniquely determine this combinatorial map.*

We may use a similar device for partial maps. Let us allow an edge to have an empty place, denoting it by pair of zeros, i.e., edges $e = \langle a, 0, 0, d\rangle$ and $e = \langle 0, b, c, 0\rangle$ being called partial edges. Now we may raise a question: which set of m partial edges specify a partial map? The answer is quite simple.

Proposition 5. *In order for m partial edges to specify a partial map, empty places in the mapping formed from the pair $E_{i,2} \to E_{i,3}$ and $E_{i,4} \to E_{i,1}$ should correspond to orbits of Q, i.e., the mapping should be a bijection.*

We don't need to require that "empty" orbits are not allowed to have common edges, because partial edge is allowed only to have one empty place.

But a set of partial edges in general in quartet notation doesn't specify a unique partial map. E.g., a pair of partial edges $\langle 0, 1, 2, 0\rangle$ and $\langle 0, 2, 1, 0\rangle$ specify two partial maps, a 2-edge with cut out face, and a 2-loop with cut out two faces.

In applications such as winged edge [1] and quad edge [8] and similar, see [2] chapter 2, actually combinatorial maps in this edge aspect are used.

3.3 Combinatorial adjacency of corners

By drawing a graph we use vertices, edges and faces. Replacing these three types of objects with rotations we retain what is minimally necessary to get the picture of the graph on some surface. One more way to see this clearly is to notice that combinatorial maps may be treated as adjacency relations of corners of halfedges of three types, vertex, face and edge adjacencies, see fig.5, 6. Taking corners of two halfedges we may assemble them into rotation in three ways: a) around common apex of corners, forming vertex rotation, b) by adjacing halfedges in a way forming a face rotation, and c) by flipping one corner to its mirror image and adjacing halfedges in a way forming an edge rotation. In this way we may replace vertices, edges and faces with the adjacency of corners of three types, and by doing this to turn graphs with rotations into combinatorial maps. Let us name these three adjacencies correspondingly p-adjacency, q-adjacency and r-adjacency.

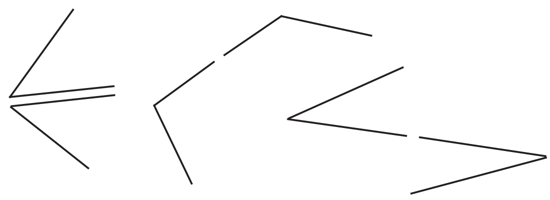

Figure 5: Three types of the combinatorial adjacency of corners of halfedges.

Let us say that a map is fixed by p-adjacency and q-adjacency if rotations P and Q are given. Thus, normalized maps are fixed by p-adjacency or q-adjacency, r-adjacency being fixed for the whole class of the maps. The knot (see [28]) appears to us as a clockwise anticlockwise alternation of r-adjacency, or *left-right r-adjacency*.

In [14] similar result is achieved via canonical triangulation of faces of the graph and getting three involutions, see Construction 1.5.20 on page 49, [14]. We may try to do the same by making our corner asymmetric by labeling it with three vertices of different color, the apex with green label, calling it a vertex's vertex, the one left to apex with blue label, calling it an edge vertex, and the one right to apex with red label, calling it a face vertex. Now we make three color rotations, green rotation for vertices, blue rotation for edges and red rotation for faces. To make our attempt legitimate we attribute to each color three "small" involutions from the edge definition, i.e., $(ab)(cd)$ for green vertex, $(ac)(bd)$ for blue vertex and $(ad)(bc)$ for red vertex. Now we have come to complete symmetry with respect to the three rotations used in combinatorial maps.

By the way, all considerations in this chapter persuades us that the choice to tag corners with numbers in the graph's picture to get combinatorial map is as legitimate and direct operation as replacing the drawing of the graph with combinatorial map. Taking this into account we use this article as advocation to use corners for coding maps in place of halfedges. Indeed, taking corners or halfedges for this coding are completely identical, but corner approach is much easier to use and comprehend by a human. Once we get persuaded that corners are as many as oriented edges, and having in mind the higher considered corner adjacencies we have already established an equivalence between the vertices-edges-faces coding and the rotations of corners coding, and, even more, a faculty to freely use both intermixed.

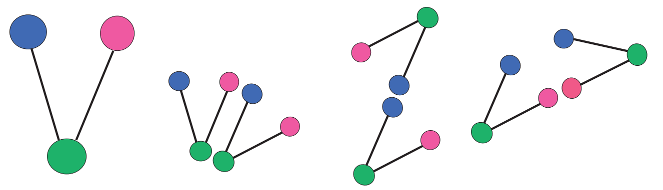

Figure 6: Combinatorial adjacency of corners symmetrized with asymmetric corner. Asymmetry in corner is made by marking vertex or red vertex, edge or blue vertex and face or red vertex. Uniting in rotations green, blue and red vertices we get correspondingly vertex, edge and face adjacency. Notice that in face adjacency one corner is flipped by mirror reflection.

4 Drawing maps

Assume someone used to combinatorial maps comes to some graph drawing specialist with request to draw a map, and the latter asks the former to prepare a graphical code for his/her combinatorial map. Doesn't it occur to him/her that the combinatorial map is a simpler structure than the code of the graph to be drawn? This situation is paradigmatic, and requests at least to be examined and maybe cured somehow, or at any rate to ask how to teach graph drawing community to learn more about rotational systems in order to make rotational prevalence we point to in this article as some working aspect. What would be that that the first had to teach the second in case one needs to draw a graph? One way is to work according the schema given in this article, in simple words, code the graph as two rotations of corners and calculate edges of the corresponding map. It would be a formal implementation of the rotational prevalence. Would it give much if we came to the necessity to implement rotations as points, lines and bordered areas in the plane, that would correspond to the elements of the pictures of the graphs we are used to? At lest, if we follow the line how the rotational prevalence would work on the level of theoretical considerations, we might keep as close as possible to it, and turn our attention on how to develop this ground level of rotational calculus with purpose to use it for graph drawing. Practically, alas, we end up with old, already used to ways to perform all the procedure of drawing of the graph, where the necessity to draw lines for edges determines all. We haven't yet invented anything to go around this. This article is motivated by the intention to think in this direction.

4.1 Combinatorial map drawer used in this article

We have endeavored to build a small drawer for the combinatorial maps using the simplest way by placing points in the plane supplied with the edge-ends with corners for the vertices and uniting them by edges from the edge rotation according 3.2. The edge ends first was united by rectangular lines, which afterwards were smoothed into curve-like edges. The drawer was supplied with a manual operation to move a vertex in the plane, that was supported with the corrector of all rest edge lines with respect to this relocated vertex.

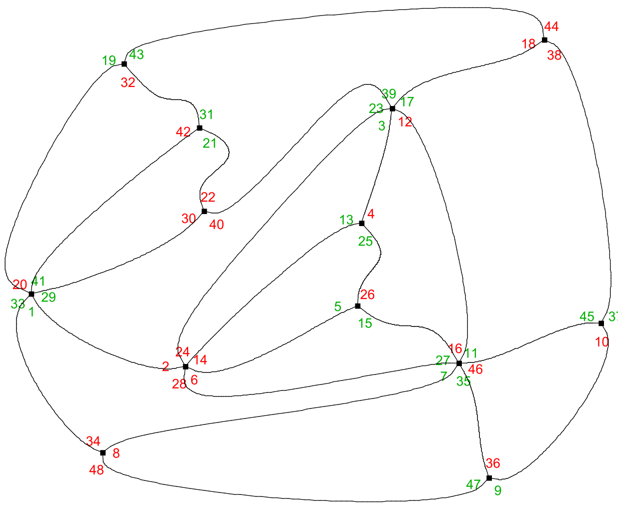

Figure 7: An example of a combinatorial map with normalized knot. The cover of the set of corners with green and red cycles is clearly seen.

The drawer used in this article was built by Paulis Ķikusts. See pictures of some maps in the figures 7, 8, 9, 10 and 11. In the examples of fig. 7, 8 a graph and its dual graph coded as a maps are drawn. Here, the maps are normalized with respect to their knots [28], and green and red cycles, that cover the corner set, are clearly seen.

Similarly as in the rotational case with three rotations we are to choose two notions from the triple - points, lines and areas - to be varied independently. The graph-theoretical tradition says that we leave areas to be determined by the first two. Thus, our rotational prevalence doesn't exceed this border of the point-line-area geometry. The breaking into this area would need new ideas. By the way, a simple way of uniforming these three geometrical quantities in a geometry of areas is suggested by the cubic combinatorial maps in the approach of C.H.C. Little, see [15, 3].

The partial maps may be useful in the graph-topological calculations, see [29, 30, 31, 32]. In the figures 9, 10, 11 we show some examples of the drawings of partial maps, that would correspond to the hypergraphs. In the figure 10 an example of the partial map (P, π), and in fig.11 the drawing of a partial map with two hyperedges of size 5 and 13 defined by pair of permutations $(1\bar{7}7)(2\bar{0}\bar{3})(3\bar{1}8)(4\bar{8}\bar{6})(6\bar{5}\bar{4})(59\bar{2})$ and $(138)(2\bar{2}\bar{4})(4\bar{1}9)(\bar{5}\bar{6}\bar{8})(5\bar{3}\bar{7})(76\bar{0})$ are given.

5 Conclusions

Combinatorial maps may be used parallel to other computational tools at the drawing of graphs. We want to persuade that it is worth to keep the so called rotational prevalence, i.e., first to fix the graph by rotations and then derive all the necessary data from these rotations. Combinatorial maps may turn to be very useful for developing such an approach.

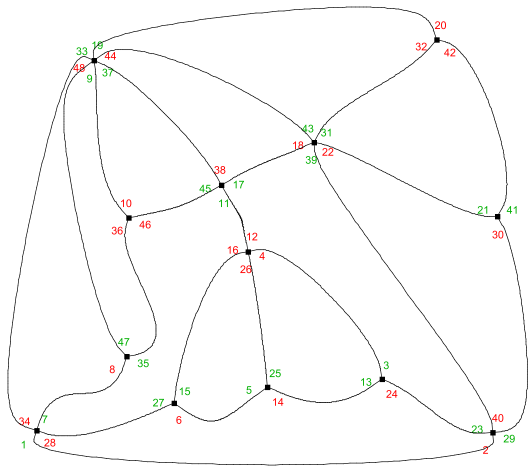

Figure 8: The dual map of the previous example.

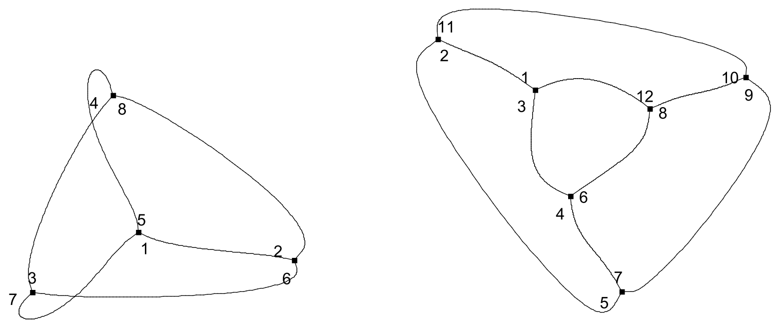

Figure 9: An example of two partial maps. Left: K_4 on torus with cut out face, comp. fig. 3. Right: the prism graph with cut out triangle faces.

6 Acknowledgements

We thank doctoral students of the second author Rihards Opmanis and Rudolfs Opmanis for their help with the building of the map drawer program.

References

[1] Baumgart Bruce G., *Winged Edge Polyhedron Representation*, Technical Report, Stanford University, Stanford, CA, USA, 1972.

[2] de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. *Computational Geometry. Algorithms and Aplications*, Springer, 3rd ed., 2008, XII, 386 pp.

[3] Bonington Paul, Little Charles, *The Foundations of Topological Graph Theory*, Springer, 1995, 178pp.

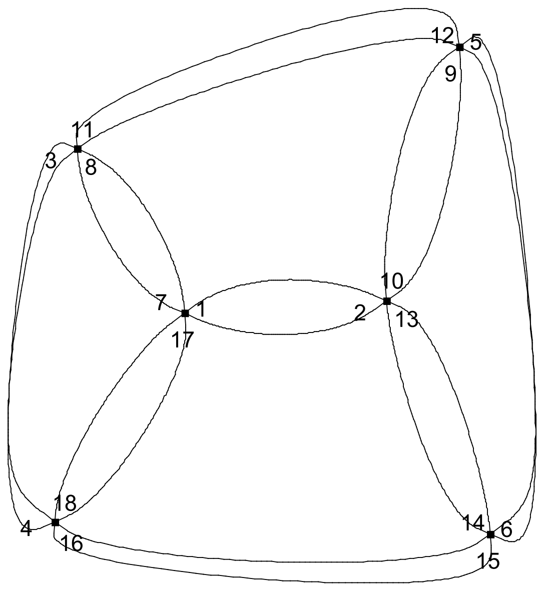

Figure 10: Partial map (P, π) as an example of the prism map (P) with cut out faces. Notice that the effect is achieved by taking the face rotation equal to π.

[4] Edmonds J. K. *A combinatorial representation for polyhedral surfaces*, Notices Amer. Math. Soc. (1960), 646.

[5] Freivalds K., Ķikusts P. *Optimum layout adjustment supporting ordering constraints in graph-like diagram drawing*, Proc. of the Latvian Academy of Sciences, Section B, Vol. 55, No 1, 2001, pp. 43-51.

[6] Freivalds K., Dogrusoz U., Ķikusts P. *Disconnected graph layout and the polyomino packing approach*, Proc. of Symposium Graph Drawing 2001, Lecture Notes in Computer Science, vol. 2265, 2002, pp. 378-391.

[7] Giblin P. J. *Graphs, Surfaces, and Homology.* John Willey & Sons, 1977.

[8] Guibas Leonidas J., Stolfi Jorge, *Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams*, ACM Transactions on Graphics, 4(2), 1985, pp. 75123

[9] Heffter L. Über das Problem der Nachbargebiete. Math. Ann., 1891, vol. 38, 477-508.

[10] Heffter L. *Uber metacyklische Gruppen und Nachbarconfigurationen*, Math. Ann. 50, 261- 268, 1898.

[11] Jacques A. *Constellations et graphes topologiques*, - In: "Combinatorial Theory and its Applications: (P.Erdos, A.Renyi, V Sos eds.), North-Holland, Amsterdam, 1970, vol. II, 657-673.

[12] Ķikusts P., Zeps D. *Graphs on surfaces*, Conf. LMS, Riga, 1994.

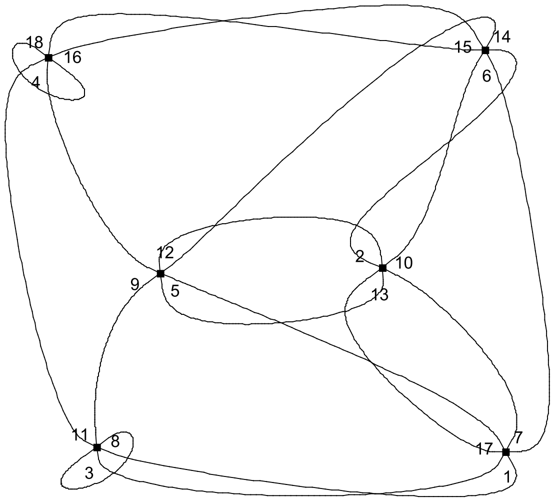

Figure 11: Partial map with two hyperedges of size 5 and 13 correspondingly defined by a pair of permutations $(1\bar{7}7)(2\bar{0}\bar{3})(3\bar{1}8)(4\bar{8}\bar{6})(6\bar{5}\bar{4})(59\bar{2})$ and $(138)(2\bar{2}\bar{4})(4\bar{1}9)(\bar{5}\bar{6}\bar{8})(5\bar{3}\bar{7})(76\bar{0})$.

[13] Ķikusts P., Ručevskis P. *Layout algorithms of graph-like diagrams for GRADE Windows graphic editors*, Proc. of Symposium Graph Drawing '95, Lecture Notes in Computer Science, vol. 1027, 1996, pp.361-364.

[14] Lando Sergei K., Zvonkin Alexander K. *Graphs on Surfaces and Their Applications*, Encyclopaedia of Mathematical Sciences, Vol 141, Low Dimensional Topology II, Springer, 2004, 455pp.

[15] Little C.H.C. *Cubic combinatorial maps*, J.Combin.Theory Ser. B 44 (1988), 44-63.

[16] Liu Yanpei A. *Polyhedral Theory on Graphs*, Acta Mathematica Sinica, New Series, 1994, Vol.10,No.2, pp.136-142.

[17] Mohar Bojan, Thomassen Carsten. *Graphs on Surfaces*, J. Hopkins Univ. Press, 2001.

[18] Ringel G. *Map Color Theorem*, Springer Verlag, 1974.

[19] Stahl S. *The Embedding of a Graph - A Survey.* J.Graph Th., Vol 2 (1978), 275-298.

[20] Stahl S. *Permutation-partition pairs: A combinatorial generalisation of graph embedding*, Trans Amer. Math. Soc. 1 (259) (1980), 129-145.

[21] Stahl S. *A combinatorial analog of the Jordan curve theorem*, J.Combin.Theory Ser.B 35 (1983), 28-38.

[22] Stahl S. *A duality for permutations*, Discrete Math. 71 (1988), 257-271.

[23] W.T. Tutte. *What is a map? New directions in the theory of graphs.* Academic Press, NY, 1973, pp. 309-325.

[24] Tutte W.T. *Combinatorial maps*, in *Graph theory*, chapter X, 1984.

[25] White A. T. *Graphs, Groups and Surfaces*, North-Holland, P.C., 1973.

[26] Zeps D. *Graphs with rotations as selfdual facet systems*, unpublished, 1993, 6pp.

[27] Zeps D. *Graphs with rotation in permutation technique*, KAM Series, N 94-274, 1994, Prague, Charles University, 8pp.

[28] Zeps D. *Graphs as Rotations*, KAM-Series, (96-327), Prague, 1996, 11pp, arXiv:0909.0104v1.

[29] Zeps. D. *Graphs with rotations:partial maps*, KAM Series, 97-365, 1996, 12pp.

[30] Zeps D. *The use of the combinatorial map theory in graph-topological computations*, KAM Series, 97-364, Prague, 1997, 8pp.

[31] Zeps D. *The theory of Combinatorial Maps and its Use in the Graph-topological computations*, PhD thesis, 1998.

[32] Zeps D. *Using combinatorial maps in graph-topological computations* , KAM Series, 99-438, Prague, 1997, 11pp.

[33] Zeps D. *Combinatorial Maps. Tutorial*, Online book, vixra.org/abs/1001.0030, 2004.

[34] Zeps D.*Combinatorial map as multiplication of combinatorial knots*, KAM Series, (2008-864), Prague, 2008, 7pp, hal-00408153.

[35] Zeps D. *Combinatorial Maps with Normalized Knot*, arXiv:1010.2615v1, 2010, 14pp.