Subdivision Drawings of Hypergraphs*

Michael Kaufmann[1], Marc van Kreveld[2], and Bettina Speckmann[3]

[1] Institut für Informatik, Universität Tübingen
mk@informatik.uni-tuebingen.de
[2] Department of Computer Science, Utrecht University
marc@cs.uu.nl
[3] Department of Mathematics and Computer Science, TU Eindhoven
speckman@win.tue.nl

Abstract. We introduce the concept of subdivision drawings of hypergraphs. In a subdivision drawing each vertex corresponds uniquely to a face of a planar subdivision and, for each hyperedge, the union of the faces corresponding to the vertices incident to that hyperedge is connected. Vertex-based Venn diagrams and concrete Euler diagrams are both subdivision drawings. In this paper we study two new types of subdivision drawings which are more general than concrete Euler diagrams and more restricted than vertex-based Venn diagrams. They allow us to draw more hypergraphs than the former while having better aesthetic properties than the latter.

1 Introduction

A graph G is a pair $G = (V, E)$, where V is a set of elements or vertices and $E \subseteq V \times V$ is a set of pairs of vertices, called edges. A *hypergraph* $H = (V, E)$ is a generalization of a graph, where again V is a set of elements or vertices, but E is a set of non-empty subsets of V, called *hyperedges* [1]. The set $E \subseteq \mathcal{P}(V)$ of hyperedges is a subset of the powerset of V.

Hypergraphs are not as common as graphs, but they do arise in many application areas. In relational databases there is a natural correspondence between database schemata and hypergraphs, with attributes corresponding to vertices and relations to hyperedges [7]. Hypergraphs are used in VLSI design for circuit visualization [6,14] and also appear in computational biology [10,12] and social networks [3].

Drawings of hypergraphs are less well-understood than drawings of graphs. There is no single "standard" method of drawing hypergraphs, comparable to the point-and-arc drawings for graphs. When drawing hypergraphs, vertices are usually depicted as points or regions in the plane, but hyperedges can have very varied forms, including Steiner trees, closed curves in the plane, faces of subdivisions, and points. As a result, there is no unique definition of planarity for hypergraphs—different drawing methods imply different, non-equivalent planarity definitions. In the following, we describe some of the drawing methods for hypergraphs in more detail.

* This research was initiated during the Bertinoro Workshop on Graph Drawing, 2008. Bettina Speckmann is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

I.G. Tollis and M. Patrignani (Eds.): GD 2008, LNCS 5417, pp. 396–407, 2009.

Hypergraph Drawings. The two most common methods to draw hypergraphs are the subset-based and the edge-based method. A subset-based drawing highlights the fact that a hypergraph can be interpreted as a set system: Vertices are drawn as points in the plane and hyperedges are drawn as simple closed curves that contain exactly those vertices that they are incident to (see Fig. 1(a)). It is easy to see that any hypergraph can be drawn this way. Subset-based drawings neither know any concept of or experience any problems with planarity. Bertault and Eades [2] show how to create subset-based hypergraph drawings.

An edge-based hypergraph drawing resembles standard drawings of graphs more closely [11]. Vertices are again drawn as points, but hyperedges are drawn as Steiner trees (see Fig. 1(b)). Edge-based drawings imply the most common definition for hypergraph planarity: A hypergraph is *planar* if and only if it has an edge-based drawing without hyperedge crossings.

Another way to draw hypergraphs is the so-called *Zykov representation*. Vertices are again drawn as points, but hyperedges are visualized by faces of a subdivision. The vertices around a face of the subdivision are the ones connected by the corresponding hyperedge (see Fig. 1(c)). To distinguish faces that represent hyperedges from faces that do not, a background color is needed to fill all faces of the subdivision that do not correspond to hyperedges. A hypergraph is *Zykov-planar* if it has a Zykov representation. Zykov-planarity is equivalent to hypergraph planarity as induced by edge-based drawings.

A hypergraph H can also be drawn as a bipartite graph where one set of vertices corresponds to the vertices of H and the other set corresponds to the hyperedges (see Fig. 1(d)). The edges of the bipartite graph represent vertex-hyperedge incidences. Also this representation immediately implies a definition of planarity, which is equivalent to Zykov-planarity and hence to hypergraph planarity as induced by edge-based drawings [17].

Many hypergraphs are not (Zykov-)planar and hence have neither an edge-based drawing, nor a Zykov representation, nor a drawing as a planar bipartite incidence graph. Motivated by this fact, Pollak and Johnson [9] study alternative definitions of hypergraph planarity which are implied by yet two more methods to draw hypergraphs—*hyperedge-based Venn diagrams* and *vertex-based Venn diagrams*. (The choice of these

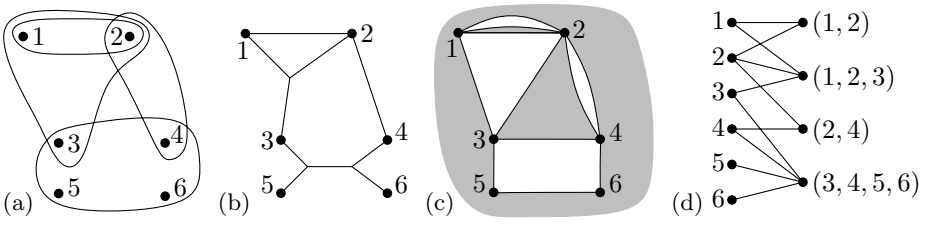

Fig. 1. Four drawings of the hypergraph $H = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 2, 3), (2, 4), (3, 4, 5, 6)\}$: (a) subset-based drawing; (b) edge-based drawing; (c) Zykov representation; (c) incidence representation (bipartite graph)

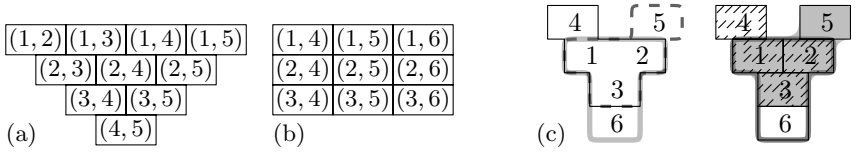

Fig. 2. A hyperedge-based Venn diagram for (a) K_5 and (b) $K_{3,3}$; (c) two different drawings of a vertex-based Venn diagram for H

names is somewhat unfortunate, since the drawings have little in common with standard Venn diagrams.)

In a hyperedge-based Venn diagram, hyperedges correspond uniquely to the faces of a planar subdivision in such a way that for any vertex v, the union of the faces corresponding to hyperedges that contain v, is connected. In a vertex-based Venn diagram, vertices correspond uniquely to faces of a planar subdivision in such a way that for any hyperedge, the union of the faces corresponding to its incident vertices is connected. Note that the subdivision itself does not show the hyperedges directly, curves that enclose unions of faces must still be drawn to visualize them. A hypergraph is *hyperedge-planar* or *vertex-planar* if a planar subdivision with the required properties exists. It is easy to see that both K_5 and $K_{3,3}$ are hyperedge-planar (see Fig. 2(a, b)). Concerning vertex-planarity, consider the following hypergraph H: H has six vertices and three hyperedges $(1, 2, 3, 4)$, $(1, 2, 3, 5)$, and $(1, 2, 3, 6)$. H is vertex-planar but not (Zykov-)planar. Figure 2(c) shows two drawings of H which showcase different methods to draw the hyperedges in a vertex-based Venn diagram.

Subdivision Drawings. Vertex-based Venn diagrams are a particular type of hypergraph drawings which we call *subdivision drawings*. In a subdivision drawing each vertex corresponds uniquely to a face of a planar subdivision. Furthermore, for any hyperedge, the union of the faces corresponding to the vertices incident to that hyperedge, the *hyperedge region*, is connected. Hypergraph drawings which are based on Euler diagrams are subdivision drawings as well. Below we discuss them in some more detail.

To draw a hypergraph H as an Euler diagram we again need to interpret H as a set system. Euler diagrams represent a collection of sets by simple, closed curves in the plane, such that the interior of each curve represents the elements of the corresponding set. Any face induced by the collection of curves is called a *zone*, which lies in the interior of some curves and in the exterior of the rest. In an Euler diagram, a zone z is only present if an element exists that is in exactly those sets whose curves have z as their common interior, and z is in the common exterior of all other curves [13]. No two zones may represent the same intersections of sets. Often, certain well-formedness conditions are considered part of the definition of Euler diagrams. These specify that no point may be the intersection of three of more curves, and all intersections of two curves are proper intersections (no two curves partially overlap). Flower and Howse call diagrams satisfying these conditions *concrete Euler diagrams* [8].

Extended Euler diagrams [16] allow curves to intersect in more general ways: They may partially coincide, and multiple intersections are allowed. Also, zones exist for all possible subsets that can be obtained by intersections. For example, the set system

$\mathcal{S} : \{ \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\} \}$ generates all four units, all six pairs, and all four triples of elements as intersections of one or more sets, so an extended Euler diagram will have 14 zones for these intersections. Strictly speaking, an extended Euler diagram is not a subdivision drawing, since there is no unique mapping between the vertices and the faces of the subdivision anymore. (There is, however, an injective mapping.)

There is no concrete Euler diagram for $\mathcal{S}$, but if we interpret $\mathcal{S}$ as a hypergraph, then a vertex-based Venn diagram of $\mathcal{S}$ with four faces exists. On the other hand, hyperedge regions of a concrete Euler diagram are simply connected, whereas hyperedge regions of a vertex-based Venn diagram need only be connected, that is, they can have holes. In this paper we study two new types of subdivision drawings for hypergraphs—*simple* and *compact* subdivision drawings—which are more general than concrete Euler diagrams and more restricted than vertex-based Venn diagrams. They allow us to draw more hypergraphs than the former while having better aesthetic properties than the latter. An interesting connection to traditional graph drawing is established by the observation that drawing simple subdivisions corresponds to the problem of overlapping cluster planarity [4] when single edges are missing.

Results. In Sect. 2 we define simple and compact subdivision drawings. We also show that there are hypergraphs which have a subdivision drawing, but not a simple subdivision drawing, and hypergraphs which have a simple, but not a compact subdivision drawing. Pollack and Johnson [9] proved that it is NP-complete to decide if a given hypergraph has a subdivision drawing. Nevertheless, there are classes of graphs that always have a subdivision drawing. In Sect. 3 we prove that hypergraphs which correspond to a particular hierarchy (when viewed as a set system) have a compact subdivision drawing where each face of the subdivision is convex. In the full paper we also show that hypergraphs which are reduced line graphs of complete graphs have a compact subdivision drawing.

2 Subdivision Drawings

Before we can define simple and compact subdivision drawings we first need to introduce some notation and state some assumptions. Let $H = (V, E)$ be a hypergraph. Two vertices u and v of H are *equivalent* with respect to E, if every hyperedge contains either both or none of u and v. To simplify the following discussion we assume that no two vertices of H are equivalent. (Equivalent vertices can easily be removed in a preprocessing step and can be added to the final drawing in an equally easy post-processing step.)

Recall that in a subdivision drawing each vertex corresponds uniquely to a face of a planar subdivision D. Furthermore, the hyperedge region of each hyperedge (the union of the faces corresponding to the vertices incident to that hyperedge) is connected. We assume that the subdivision D has only vertices of degree three. Not every bounded face of D has to correspond to a vertex of H. We call a face that does correspond to a vertex of H a *vertex face*.

A subdivision drawing is *simple* if every hyperedge region is simple, that is, bounded by one simple closed curve. A subdivision drawing is *compact* if it is simple and each

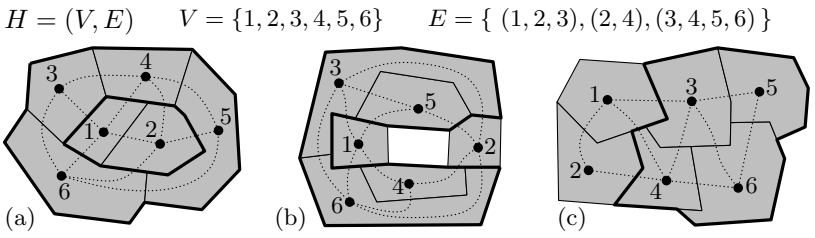

Fig. 3. Subdivision drawing of a hypergraph H, only the curves bounding the hyperedge $(3, 4, 5, 6)$ are indicated (a); simple subdivision drawing of H (b); compact subdivision drawing of H (c). Dotted edges indicate a planar support.

bounded face of D is a vertex face. That is, the complement of all vertex faces is connected. Note that collapsing a face that is not a vertex face will never destroy the connectivity of hyperedge regions, but it may create non-simple hyperedge regions. Consider the white face in Fig. 3(b): Removing it will make the region of either hyperedge $(1, 2, 3)$ or hyperedge $(3, 4, 5, 6)$ non-simple.

A concrete Euler diagram is a compact subdivision drawing which has only proper intersections between the simple closed curves which bound hyperedge regions: No three curves have a common point and no two curves intersect over a stretch of positive length. In Flower and Howse's terminology, not even the set system $\{ \{1, 2\}, \{2, 3\}, \{1, 3\} \}$ has a concrete Euler diagram.

A graph G is a *support* for a hypergraph H if the vertices of G correspond to the vertices of H such that for each hyperedge e the subgraph of G induced by e is connected. G is a *planar support* if it is planar. A planar support G is *simple* if G has a planar embedding where each cycle in a subgraph induced by a hyperedge e does not have any other vertex of G on the inside. Hence the planar support in Fig. 4(a) is a simple planar support if every hyperedge that induces the cycle c is also incident to vertex v. Intuitively, the planar support is a subgraph of the dual graph of any subdivision drawing of H (see Fig. 3).

Observation 1. *A hypergraph H*

(i) has a simple subdivision drawing if and only if it has a simple planar support.
(ii) has a compact subdivision drawing if and only if it has a simple planar support with an embedding where all bounded faces are triangulated.

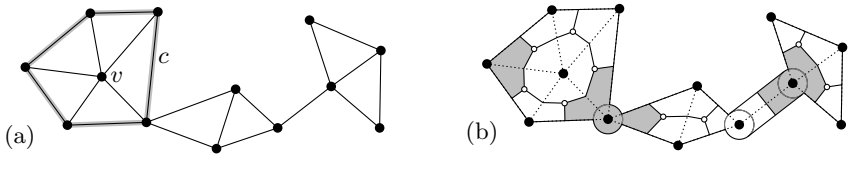

Fig. 4. A (simple) planar support (a); turning a support into a subdivision (b)

Subdivisions and their dual graphs have been extensively studied in, for example, graph theory and VLSI design, and there are several methods that can turn a planar support into a dual subdivision. For completeness, we sketch an easy approach that creates such a subdivision. Use any straight-line embedding of the support. (i) Trace a "race track" around every edge and cut it in the middle. Each vertex face is formed by the union of all half race tracks adjacent to the corresponding vertex. (ii) Place a vertex inside each triangle and connect it to the middle of each edge, forming three quadrilaterals. Place a small circle around every cut vertex, trace a race track around every bridge edge, and cut it in the middle (see Fig. 4(b)). Each vertex face is now the union of all quadrilaterals, circles, and half race tracks adjacent to the corresponding vertex in the support.

We now show that there are hypergraphs which have a subdivision drawing, but not a simple subdivision drawing, and hypergraphs which have a simple but not a compact subdivision drawing. First, consider the hypergraph H_1 on four vertices. The hyperedges of H_1 are the six pairs and the four triples of vertices. Since the hyperedges include all pairs of vertices all vertex faces of the subdivision must be adjacent. Hence, modulo re-labeling of vertices and the removal of white non-vertex faces, we must have a subdivision drawing as the one depicted in Fig. 5(a) with a non-simple hyperedge region for hyperedge $(2, 3, 4)$.

Second, consider the hypergraph H_2 that is schematically depicted in Fig. 5(b). The hyperedges of H_2 are all black edges plus the two hyperedges with nine vertices each, which are indicated by the gray contours. The black edges form a planar support which is uniquely defined, up to the choice of the outer face. Each hyperedge region is simply connected, so the black edges even form a simple planar support. However, we can not triangulate either of the quadrilaterals without making one hyperedge non-simple. Since at most one of the quadrilaterals can be the outer face, H_2 does not have a compact subdivision drawing.

3 Hierarchies and Subdivision Drawings

In this section we characterize certain hypergraphs which have a compact subdivision drawing. In fact, they even have a compact subdivision drawing where each face of the subdivision is convex. We again view hypergraphs as set systems and study a particular *hierarchy* defined on these set systems. Any hypergraph H with n vertices and k

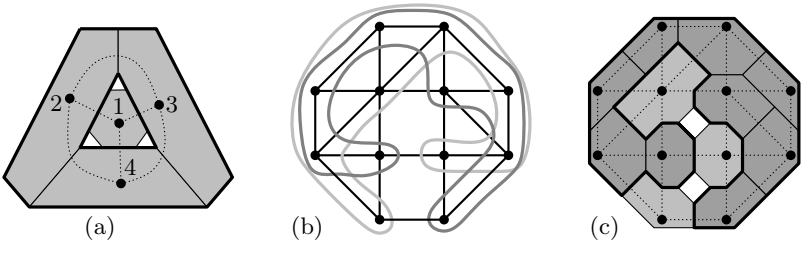

Fig. 5. A hypergraph which has no simple subdivision drawing (a); the hypergraph H_2 (b); a simple but not compact subdivision drawing of H_2 (c)

hyperedges can be interpreted as a set system $\mathcal{S} = \{S_1, \ldots, S_k\}$ on a base set M of n elements. (Here we again assume that no two vertices of H are equivalent.) Subdivision drawings naturally visualize set containment well: The region of a set (hyperedge) S_i is contained in the region of a set (hyperedge) S_j if and only if $S_i \subset S_j$.

A *hierarchy* $\mathcal{H}$ is a directed acyclic graph induced by a set system or hypergraph. A hierarchy has two types of vertices: *base vertices*, which represent a singleton set for each element in the base set M, and *set vertices*, which represent each set in the set system $\mathcal{S}$. Hence $\mathcal{H}$ has $n + k$ vertices. With slight abuse of notation we refer to the vertices of $\mathcal{H}$ by the names of the sets they represent. $\mathcal{H}$ has a directed edge (S_1, S_2) with $S_1, S_2 \in \mathcal{S} \cup \{\{v\} : v \in M\}$ if and only if $S_2 \subset S_1$ and for no set $S_3 \in \mathcal{S}$, we have $S_2 \subset S_3 \subset S_1$. That is, edges are directed from larger to smaller sets and represent direct containment—our hierarchies do not contain transitive edges. The base nodes are the *leaves* of the hierarchy (with only incoming edges) and the set nodes are the *internal nodes* of the hierarchy (each internal node has at least two outgoing edges). A hierarchy $\mathcal{H}$ corresponds uniquely to a hypergraph H and vice versa. $\mathcal{H}$ is a *planar hierarchy* if it is planar. We say that a hierarchy is *based* if it has a set vertex S_M that represents the complete base set M. S_M is necessarily the root of the hierarchy and has only outgoing edges. See Fig. 6 for an example.

In the following we prove that a hypergraph H has a compact drawing where each face of the subdivision is convex if the corresponding hierarchy $\mathcal{H}$ is based and planar. In particular we describe an algorithm that transforms $\mathcal{H}$ into an outerplanar support for H by "sliding" certain edges of $\mathcal{H}$ down to the leaf level. The complete process comprises the following four steps:

1. Fix an embedding and construct a depth-first search spanning tree of $\mathcal{H}$.
2. Slide all non-tree edges of $\mathcal{H}$ down to the leaf level.
3. Remove all internal nodes of $\mathcal{H}$ to create an outerplanar support for H.
4. Construct a compact subdivision drawing from the outerplanar support.

We now describe each step in more detail.

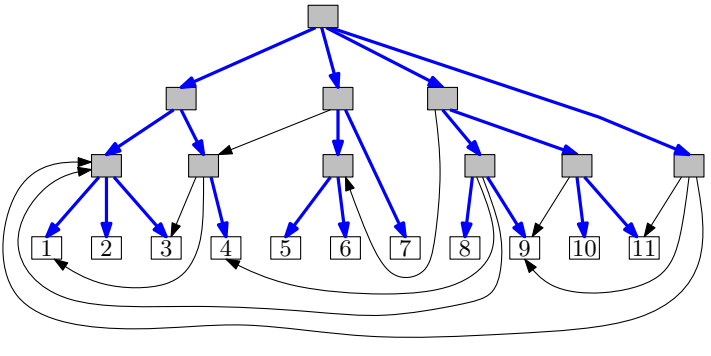

Fig. 6. A based planar hierarchy $\mathcal{H}$ defined by $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ and the sets $\{1, 2, 3\}$, $\{1, 3, 4\}$, $\{5, 6\}$, $\{1, 2, 3, 4, 8, 9\}$, $\{9, 10, 11\}$, $\{1, 2, 3, 9, 11\}$, $\{1, 2, 3, 4\}$, $\{1, 3, 4, 5, 6, 7\}$, $\{1, 2, 3, 4, 5, 6, 8, 9, 10, 11\}$, $S_M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$

Step 1: *Embedding $\mathcal{H}$ and constructing a depth-first search spanning tree.*

We embed $\mathcal{H}$ such that the root S_M lies on the outer face. Recall that all edges are directed from the root to the leaves. The embedding defines a left to right order (counterclockwise) of the children of each node. We traverse the hierarchy in a depth-first search (DFS) manner, creating an ordered DFS spanning tree, where every edge that reaches some node for the first time defines the *true parent* of that node. The true parent structure is a tree which categorizes the edges of $\mathcal{H}$ into *tree edges* and *non-tree edges*. W.l.o.g. we relabel the vertices in the base set with $1, 2, \ldots, n$ in the order in which they are encountered in the DFS traversal.

Step 2: *Slide all non-tree edges down to the leaf level.*

We now transform the based planar hierarchy $\mathcal{H}$ into another based planar hierarchy $\mathcal{H}'$ where all non-tree edges point to leaves. We do this in such a way that the outerplanar support for $\mathcal{H}'$ which we create in Step 3 is also an outerplanar support for $\mathcal{H}$.

Let $\vec{a}$ be any non-tree edge that points to an internal node S_i. We distinguish five different cases, depending on the next edge in the clockwise order around S_i, see Fig. 7. If the next edge in clockwise order is (i) an outgoing tree edge, then we can slide $\vec{a}$ to point to the child of that tree edge. If it is (ii) an incoming non-tree edge $\vec{c}$, then we can proceed with $\vec{c}$ instead. If it is (iii) an outgoing non-tree edge $\vec{c}$, then we can slide $\vec{a}$ to point to the destination of $\vec{c}$. Finally, the next clockwise edge can be an incoming tree edge from the parent. In this case we consider the counterclockwise neighbor of $\vec{a}$ at S_i. Due to the construction of the DFS tree, this can be only an outgoing tree edge (iv) or an incoming non-tree edge $\vec{c}$ (v). We can treat (iv) symmetric to (i) and (v) symmetric to (ii).

The following lemma shows that this transformation preserves the adjacencies captured by the original hierarchy.

Lemma 1. *For any based planar hierarchy $\mathcal{H}$ there exists another based planar hierarchy $\mathcal{H}'$ where every non-tree edge points to a leaf, and a planar support for $\mathcal{H}'$ is also a planar support for $\mathcal{H}$.*

Proof. We first argue that the transformation described above terminates and results in a hierarchy $\mathcal{H}'$ where all non-tree edges point to leaves. The five cases of the transformation can be grouped as follows: In cases (i), (iii), and (iv) the non-tree edge $\vec{a}$ slides

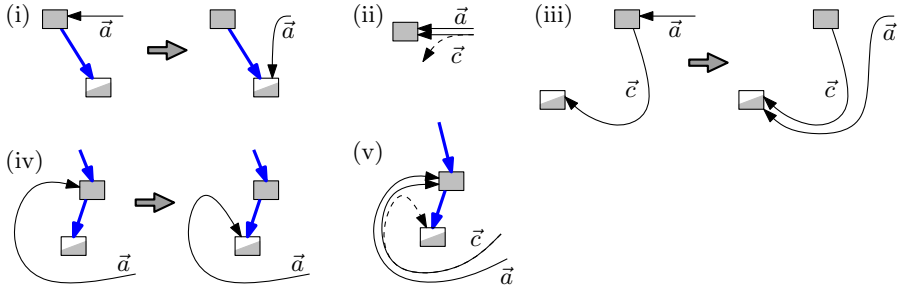

Fig. 7. Cases of the transformation. Grey nodes are internal, half-grey nodes can be internal or leaf nodes.

to a node S_j with $S_j \subset S_i$. In cases (ii) and (v) no sliding is done, but it is easy to see that these cases cannot occur more often than there are non-tree edges pointing to S_i. Hence the process terminates when all non-tree edges point to leaves.

Now we prove that a planar support for $\mathcal{H}'$ is also a planar support for $\mathcal{H}$ by considering the corresponding hypergraphs (set systems). Let H be the hypergraph corresponding to $\mathcal{H}$. We argue that any sliding operation results in a hypergraph H' whose planar support is also a planar support for H. In particular, let (S_i, S_j) be a non-tree edge that is slid and which becomes edge (S_i', S_k). The sets of H' are precisely the sets of H with the exception of S_i which is replaced by S_i'. We know that $S_k \subset S_j$ and $S_i' \subseteq S_i$. Consider a planar support G' for H'. The base elements in S_j, S_k, and S_i' are connected in G'. We have to show that the base elements of S_i are also connected in G', although S_i is not a set of H' and hence not a node of $\mathcal{H}'$. We have $S_i = S_i' \cup S_j$ and also $S_k = S_i' \cap S_j$ which implies $S_i' \cap S_j \neq \emptyset$. Hence we can conclude that the base elements of S_i are connected in G'. Since this argument holds for every sliding operation the lemma follows. □

Step 3: *Remove all internal nodes and create an outerplanar support.*

After Step 2 we have a hierarchy where every non-tree edge points directly to a leaf (see Fig. 8). We now embed this hierarchy in the plane such that all leaves (base elements) lie on a horizontal line ℓ. Non-tree edges are drawn as curves that connect an internal node to a leaf, either without crossing ℓ or by crossing it exactly once. Such a planar embedding always exists—its can be obtained directly from the planar embedding used in Step 1 of our algorithm (see Fig. 8).

We say that a base element *encounters* ℓ if either its leaf is intersected by ℓ or if ℓ intersects a non-tree edge that is directed to its leaf. Let $W = w_1, \ldots, w_s$ be the sequence of the base elements as they encounter ℓ from left to right. In Fig. 8, the sequence is $W = 1, 2, 3, 1, 4, 5, 6, 7, 6, 8, 4, 1, 9, 10, 11, 9, 1$. The base elements in any set (internal node) form a subinterval of W. Hence, any planar graph on the base elements that realizes all adjacencies of W is a planar support. Therefore we call W the *support sequence*.

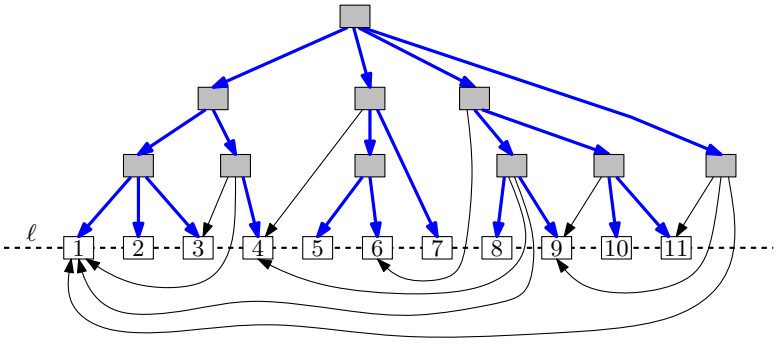

Fig. 8. Hierarchy after the transformation

Lemma 2. *A support sequence W does not have a subsequence $..a..b..a..b..$, where a and b represent any two distinct base elements.*

(In other words, W is a Davenport-Schinzel sequence of order 2 [15].)

Proof. By construction, the first occurrences of a and b in W correspond to leaves, the later ones to crossings of non-tree edges with ℓ. A subsequence $..a..b..a..b..$ implies that the non-tree edges to a and b cross below ℓ, but the initial hierarchy was planar and the transformations preserved planarity. □

Lemma 3. *There is an outerplanar graph G that realizes all adjacencies of W.*

Proof. Assume that the base elements are numbered $1, \ldots, n$. We compute G from W as follows. Create a node for every base element and connect them into a path using $n-1$ edges $(i, i+1)$, $1 \leq i \leq n-1$. Scan W from left to right, and for any repeated occurrence of i in $\ldots j, i, k \ldots$, create edges (i, j) and (i, k) such that the path edges $(i, i+1)$ always keep the unbounded face to the left. The planarity of this construction follows directly from the planarity of the transformed hierarchy $\mathcal{H}'$. G is outerplanar since all path edges $(i, i+1)$ bound the outer face and hence all nodes are incident to the outer face, see Fig. 9. □

Step 4: *Construct a compact subdivision drawing from the outerplanar support.*

The outerplanar graph G that results from Step 3 forms the basis of the planar support. We add an edge $(1, n)$ if it is not present yet and triangulate all bounded faces (in Fig. 9, only the faces $1, 4, 8, 9$ and $4, 5, 6, 8$).

An easy way to construct the regions is the following: Use any straight-line embedding of the planar support with triangulated bounded faces. For each triangle, choose any point in its interior, for instance the center of mass. For any edge, place an extra point in the middle. Partition each triangle into three quadrilaterals by drawing edges between the center and three edge midpoints. The region of each node is the union of the incident quadrilaterals (see Fig. 10(a)).

We can ensure that each face of the subdivision is convex, by using a slightly more involved method based on Voronoi diagrams. Dillencourt [5] showed how to realize an outerplanar graph as the Delaunay triangulation of points. The dual of this Delaunay triangulation, clipped to lie within a bounding box, is a compact subdivision drawing where each face is convex, see Fig. 10(b).

Theorem 1. *A hypergraph that corresponds to a based planar hierarchy has a compact subdivision drawing where each face of the subdivision is convex.*

To conclude this section we note that hypergraphs corresponding to a (non-based) planar hierarchy need not have a compact subdivision drawing, or even a simple one. Consider again the hypergraph H_1 on four vertices. The hyperedges of H_1 are the six pairs

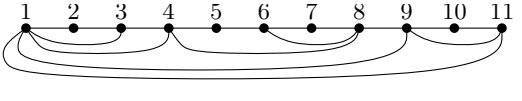

Fig. 9. Planar support for the hierarchy shown in Fig. 8

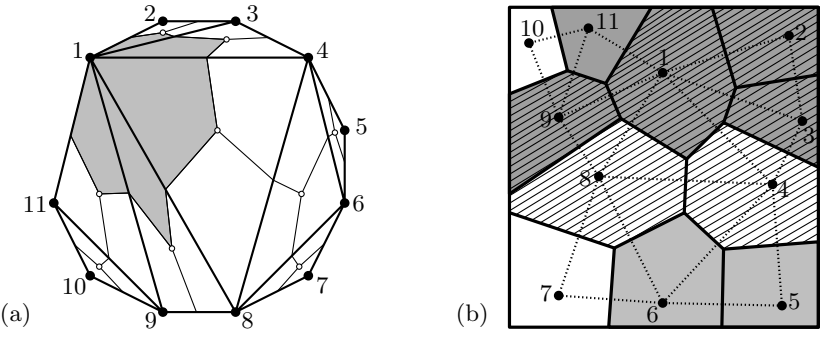

Fig. 10. Triangle-partition based method of constructing a subdivision for the planar support of Fig. 9, the face of node 1 is indicated (a); Voronoi diagram based method for the same support, showing 3 sets: $\{5, 6\}$, $\{1, 2, 3, 9, 11\}$, and $\{1, 2, 3, 4, 8, 9\}$ (b)

and the four triples of vertices. Its hierarchy is planar, as shown in Fig. 11(a), but—as argued in Sect. 2—H_1 has no simple subdivision drawing. It is easy to see, however, that each hypergraph corresponding to a planar hierarchy has a subdivision drawing, the planar support can be extracted from any planar embedding of the hierarchy by iteratively contracting edges incident to base vertices.

4 Conclusion and Open Problems

We introduced the concept of subdivision drawings of hypergraphs which comprises both vertex-based Venn diagrams and Euler diagrams. We studied two new types of subdivision drawings, simple and compact subdivision drawings, and established some of their basic properties. We also characterized certain hypergraphs that have a compact subdivision drawing.

It is NP-complete to decide if a hypergraph has a subdivision drawing, but it is not clear if the same result holds for simple and compact subdivision drawings as well. Of

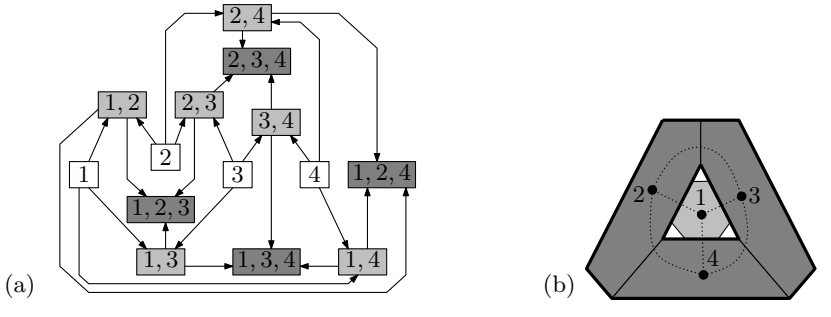

Fig. 11. Planar drawing of the hierarchy of H_1 (a); subdivision drawing of H_1 (b)

course it would be interesting to identify further classes of hypergraphs that have simple or compact subdivision drawings.

References

1. Berge, C.: Graphs and Hypergraphs. North-Holland, Amsterdam (1973)
2. Bertault, F., Eades, P.: Drawing hypergraphs in the subset standard. In: Marks, J. (ed.) GD 2000. LNCS, vol. 1984, pp. 164–169. Springer, Heidelberg (2001)
3. Brinkmeier, M., Werner, J., Recknagel, S.: Communities in graphs and hypergraphs. In: ACM CIKM 2007, pp. 869–872 (2007)
4. Didimo, W., Giordano, F., Liotta, G.: Overlapping cluster planarity. In: Asia-Pacific Symposium on Visualisation, pp. 73–80 (2007)
5. Dillencourt, M.B.: Realizability of Delaunay triangulations. Information Processing Letters 33(6), 283–287 (1990)
6. Eschbach, T., Günther, W., Becker, B.: Orthogonal hypergraph drawing for improved visibility. J. of Graph Algorithms and Applications 10(2), 141–157 (2006)
7. Fagin, R.: Degrees of acyclicity for hypergraphs and relational database schemes. J. of the ACM 30(3), 514–550 (1983)
8. Flower, J., Howse, J.: Generating Euler diagrams. In: Hegarty, M., Meyer, B., Narayanan, N.H. (eds.) Diagrams 2002. LNCS, vol. 2317, pp. 61–75. Springer, Heidelberg (2002)
9. Johnson, D., Pollak, H.: Hypergraph planarity and the complexity of drawing Venn diagrams. J. of Graph Theory 11(3), 309–325 (1987)
10. Lundgren, J.R.: Food webs, competition graphs, competition-common enemy graphs and niche graphs. Applications of Combinatorics and Graph Theory to the Biological and Social Sciences 17, 221–243 (1989)
11. Mäkinen, E.: How to draw a hypergraph. International J. of Computer Mathematics 34, 177–185 (1990)
12. Ramadan, E., Tarafdar, A., Pothen, A.: A hypergraph model for the yeast protein complex network. In: 18th Parallel and Distributed Processing Symposium, pp. 189–190 (2004)
13. Ruskey, F., Weston, M.: A survey of Venn diagrams. The Electronic J. of Combinatorics (2005), `http://www.combinatorics.org/Surveys/ds5/VennEJC.html`
14. Sander, G.: Layout of directed hypergraphs with orthogonal hyperedges. In: Kreowski, H.-J., Montanari, U., Orejas, F., Rozenberg, G., Taentzer, G. (eds.) Formal Methods in Software and Systems Modeling. LNCS, vol. 3393, pp. 381–386. Springer, Heidelberg (2005)
15. Sharir, M., Agarwal, P.K.: Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, Cambridge (1995)
16. Verroust, A., Viaud, M.L.: Ensuring the drawability of extended Euler diagrams for up to 8 sets. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS, vol. 2980, pp. 128–141. Springer, Heidelberg (2004)
17. Walsh, T.: Hypermaps versus bipartite maps. J. of Combinatorial Theory 18, 155–163 (1975)