Visualise Undrawable Euler Diagrams

Paolo Simonetto, David Auber
LaBRI, Université Bordeaux I
paolo.simonetto@labri.fr, auber@labri.fr

May 8, 2008

Abstract

Given a group of overlapping sets, it is not always possible to represent it with Euler diagrams. Euler diagram characteristics might collide with the sets relationships to depict, making it impossible to outline a correct draw. In order to be able to show a greater class of instances, Euler diagrams have been extended allowing more general patterns, but so far all the most common definitions cannot represent all the possible connection between sets.

We aim to introduce methods and constructions to produce a clear representation, as close as possible to Euler diagrams, even for sets that are not formally drawable in that way.

We will investigate on the reasons that make a diagram undrawable, in order to evaluate how and when to apply the mentioned structures, and to give the foundations necessary to design algorithms for this purpose.

Keywords—**Euler diagrams, overlapping clustering**

1 Introduction

Euler diagrams [4] are the most natural and used way to depict sets and their reciprocal relationships. They consist in an association between regions of the plan and the abstract sets, where the topological concepts of inclusion, exclusion and overlap of these regions are used to represent the analogue sets relationships (fig. 1).

As these diagrams were introduced by Euler by examples, there is not a complete agreement about their formal definition. Some topological characteristics (such the shape of the sets or the way they intersect) might be identified either as essential traits or merely aesthetic ones, making authors propose different definitions [5], [3], [9].

Drawing Euler diagrams is a difficult task. The potential growth of the complexity of the diagram is exponential with respect to the increase in the number of sets represented, as n sets might form up to 2^n intersections [8]. For this reason, drawing Euler diagrams is challenging even for instances of a dozen of sets [9].

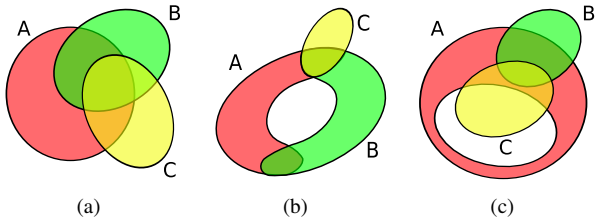


Figure 1: Euler diagrams as studied by different authors. (a) shows Euler diagrams as studied in [5]. They do not permit overlapping lines and multiple-line crossing points. (b) shows the more general Euler diagrams studied in [3]. Multiple line crossing point and overlapping boundary are allowed, as well as disconnected overlaps between the same sets. (c) shows Extended Euler diagrams (EED) as defined in [9]. Holes inside the sets are permitted.

Euler diagrams and clustering. The main practical aim of our research is to visualise overlapping clustering in a clear way. Large telecommunication networks, biological and social networks, financial data, are usually represented as graphs and visualised through embedding of graphs. Grouping elements in these graphs exactly corresponds to defining set combinations, and the visualisation of these sets can be achieved using Euler diagrams.

Even if clustering is classically intended as partitioning the elements, overlapping clustering is an interesting approach in many fields. Algorithms producing possibly overlapping sets have been defined, for instence, for analysing social networks [7] or protein-protein interaction networks [1].

In order to visualise each clustering detected, we need to ensure we are always able to represent overlapping sets. Standard Euler diagram definitions are not able to represent all the possible set configurations, as some of them have a topological structure that inevitably violates the basic diagram rules.

In section 2 we will present some work relative to Euler diagrams. In particular, we will describe more in depth the problem of drawing Euler diagrams, and we will introduce some issues about their readability and comprehension. In section 3, we will describe the relation between Euler diagrams and graphs, and we will introduce some useful definitions. In section 4 we will introduce the *Euler representation*, the kind of diagrams we will use to overcome standard Euler diagram limitations. Finally, in section 5 we will analyse how is practically possible to manage the characteristics of Euler representations, and we illustrate a possible algorithm paradigm for their generation.

2 Related work

Initial usage of these diagrams has been made by Euler for reasoning on categorical proposition and syllogisms [4]. John Venn also studied Euler diagrams as a tool for logical reasoning, proposing a particular subclass of them successively called Venn Diagrams [8].

Nowadays, Euler diagrams are widely and more frequently used in the set theory field. Answering to problems related to their existence and drawability has become crucially important.

The problem of identifying and drawing a Euler diagram is called the *Euler Diagram Generation Problem* (EDGP). The usual way to approach this problem goes trough the detection of the topological structure of the intersections between the sets, the creation of a skeleton graph and the identification of a planar embedding on the plane. The several approaches to EDGP differ in the input given and the properties of returned Euler diagrams.

Euler diagram definitions. Flower and Howse [5] developed a method to obtain a clear and simple subclass of Euler diagrams (fig. 1.a). In this class, the lines of the diagram do not overlap and intersect just pairwise. Although these limitations create nicer diagrams, they are not merely aesthetic, as they reduce the range of the representable instances.

EDGP has also been studied as planarization of *hypergraphs* [6]. Hypergraphs are graphs in which edges are identified as generic subsets of nodes, rather than couples of them. Drawing hypergraphs in their vertex-based planar representation has been proved to be equivalent to the generation problem of a class of Euler-like diagrams (EED, fig. 1.c) by Verroust and Viaud [9]. They introduced this class of diagrams, that can be informally thought of Euler diagrams that might contain holes, and proved that they are always drawable when representing eight or less intersecting sets.

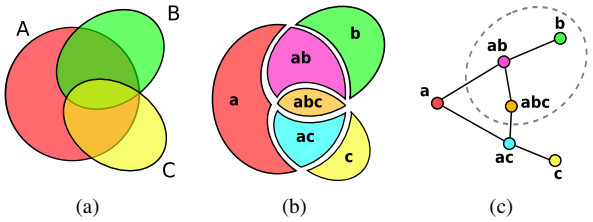


Figure 2: From the Euler diagram to the intersection graph. (a) the original diagram. (b) individuation of the diagram zones. (c) the resulting intersection graph. The dashed line is not part of the graph, but shows how to reverse the procedure. For drawing the boundary of the class B we need to enclose the nodes b, ab, abc and intersect the edges $(a, ab), (ac, abc)$.

The possibly more exhaustive analysis on general Euler diagrams (fig. 1.b) and their representation has been done by Stirling C. Chow in his PhD thesis [3]. Chow analysed the drawability of Euler diagrams in several different cases, although his work was essentially focused on the correspondent problems for area-proportional Euler diagrams.

Representable instance classes. Each of the quoted approaches is able to depict a different class of instances. Euler diagrams as defined by Flower and Howse (fig. 1.a) cannot represent, for instance, the diagram in fig. 1.b. The class of instances respresentable by those simple Euler diagrams is actually a proper subset of the instances representable by Euler diagrams as defined by Chow (fig. 1.b). In his work [3], Chow also showed how Euler diagrams are a proper subset of Euler-like diagrams like EED (fig. 1.c). Unfortunately, even EED can represent just a proper subset of all the possible instances of EDGP.

All the previous approaches are not suitable to be used to represent general groups of overlapping sets, unless accepting to have no-output for non representable instances.

Diagram readability. As we will necessarily have to force some rules of well defined Euler diagrams, it is essential to understand which characteristics are more important for their comprehension. Benoy and Rodgers [2] started answering these issues evaluating the readability of Euler diagrams according to three aesthetic parameters: set boundary irregularity, zone area inequality, boundary closeness. They found evidence that all of them strongly contribute in diagram comprehension.

3 Euler diagrams and graphs

Even if it is possible to define Euler diagrams in a mathematical and formal way, working directly with them is quite complicated. For this reason, Euler diagrams are usually studied and analysed as graphs.

We will represent diagrams as graphs in a way which is quite common in literature. This way is illustrated in fig. 2 and consists in the construction of what we will call intersection graphs.

We start having a collection of sets to represent. These sets are defined independently of each other on a set of elements, so they will generally overlap. We will indicate this collection with $C = \{C_a, C_b, \ldots\}$[1]. To avoid confusion with the more common word "set", we will call each C_x *class* and C itself *classification*.

Zone decomposition. Starting from a Euler diagram, it is possible to divide it in *zones* (fig. 2.b). Zones are the regions of the plan described by the way classes overlap: each of them contains all the, and only the, elements that are contained exactly in the same set S of classes. For instance, if $S_{ab} = \{C_a, C_b\}$, than the relative zone will contain all the, and only the, elements that are contained in the classes C_a and C_b, but not in others.

We will label each of the zones with the letters associated to the classes in S, so Z_{ab}[2] represents the mentioned zone. More formally, we will identify Z_{ab} with the set:

$$Z_{ab} = \left(\bigcap_{C_x \in S_{ab}} C_x \right) \bigcap \left(\bigcap_{C_x \notin S_{ab}} \overline{C_x} \right) = C_a \cap C_b \cap \overline{C_c}$$

similarly to what has been defined by other authors [3].

Intersection graphs. From the zone decomposition we can easily construct a graph, called *intersection graph* (fig. 2.c), that shows the interconnections between the classes. The graph has one node for each zone of the diagram, and one edge for each shared boundary between two zones.

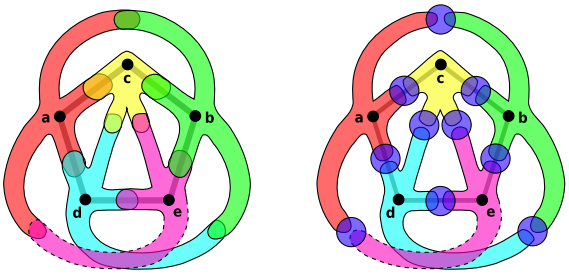
It is possible to prove that intersection graphs and Euler diagrams have the same expression power, and that there exists a bijection between equivalent Euler diagrams and equivalent intersection graphs [3]. This is proved showing constructive methods to move from one structure to the other.

For the reverse operation, that is obtaining a Euler diagram from an intersection graph, it is sufficient to realise where the classes boundaries have to be drawn. For each class, we need to consider the cut of the class nodes and the corresponding cutting edges. The set boundary can be drawn keeping in mind that it has to group the class nodes and intersect each cutting edge (fig. 2.c).

As they are equivalent, we will use diagrams or intersection graphs indifferently according to which one is clearer in the specific case.

[1] We will identify classes in pictures using just the pedix in capital letters.

[2] We will identify zones in pictures using just the pedix in lower case letters.

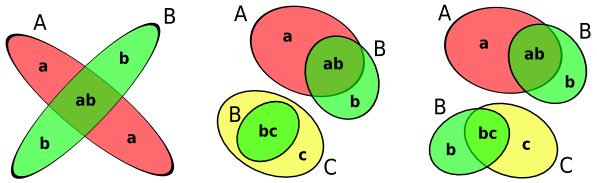


(a) (b)

Figure 3: (a) the complete graph K_5 generates an example of a diagram that is not Eulerian, as any attempt to draw it generates disconnected zones. In fact, we will have to disconnect the zones d and e (see fig. 4.a) to draw the dashed link. The same graph is drawable, if we allow duplicated zones. (b) an example of a graph that is not drawable whitout disconnecting classes, even if allowing disconnected zones. The circular sets are all meant to be distinct. This time any attempt to draw the dashed link brings undesired overlaps, so the class E will have to remain disconnected.

4 Euler representation

To be able to represent classifications that do not have a Euler diagram, we need to use a structure less restrictive. We will call this structure *Euler representation*,



(a) (b) (c)

Figure 4: (a) a diagram with disconnected zones, as zone a and zone b are represented by separated regions divided by the zone ab. (b) a diagram with disconnected classes, as class B has the zones ab, b separated from zone bc. (c) a diagram with disconnected zones and classes, as zone b is duplicated and the class B is disconnected.

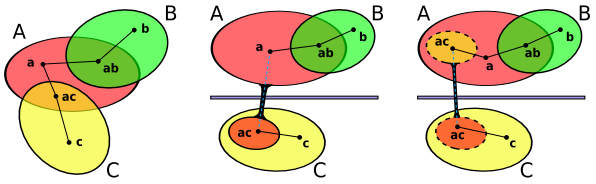


Figure 5: Visualisation of disconnected classes. (a) the original diagram, showing the relationships we aim to represent. Let us suppose the zones ac, c are not directly reachable from the others. (b) shows a possible way to depict a link between separated zones of the same class. This representation does not show straight away that the class A contains ac, especially if they are positioned far apart from each other. (c) shows the duplication of the zone ac and its nodes. A spotted boundary is used to indicate that the zone has been cloned and not simply represented with separated regions. This representation shows in a more immediate way that the classes A and C interact with each other, as well as it shows all the elements of the same class in the same connected area.

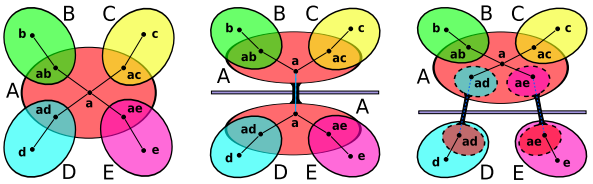


Figure 6: Another example of disconnected classes. (a) the original diagram, built without particular constraints. Let us now assume class D and E are not reachable from A, B, C. (b) the diagram obtained when representing the zone a as two separated regions. In this case, node duplication is generally not meaningful for a better comprehension of the diagram. (c) the same graph obtained duplicating the zones ad and ae and their nodes. Again, node duplication can be made clear by using a dashed line for the boundary. Altrough this solution allows us to see all the nodes of the same class in a connected region, it tends to be less readable than the previous one because of the greater number of extra links required.

and we will design its properties investigating the factors that make an EDGP instance undrawable.

Zone connectivity. According to Chow [3], a set of closed curves is a Euler diagram if every non-empty zone is represented as a connected region. The zone connectivity is the first problem for the existence of a Euler diagram. We can easily show EDGP instances[3] that are not drawable without splitting the zones in disconnected regions (fig. 3.a), proving that zone connectivity is actually a limiting condition.

Relaxing this condition we are practically allowed to duplicate a zone in a different area of the diagram (fig. 4.a), as long as we keep the classes connected. Unfortunately, this is not sufficient to draw every EDGP instance, as some of them are not representable even dropping this bound (fig. 3.b).

[3]These difficult instances are usually built starting from unplanar graphs and mapping sets in the graph elements in an suitable way.

For instance, we can associate sets N_i to the nodes, sets E_j to the edges, and impose that each set E overlaps only with the sets N associated to the nodes incident to their edges. This implies that the edges cannot overlap, otherwise we will describe an intersection between sets E that are not defined in our model.

For the Kuratowski's theorem, a graph containing a subdivision of the complete graph K_5 or the complete bipartite graph $K_{3,3}$ is not planar, or in other words, it cannot be represented without drawing crossing edges. These graphs, through the association explained, bring us to examples of undrawable Euler diagrams.

Class connectivity. In Euler diagrams classes are represented by a connected region, as implied by the usage of a single closed curve for each class. Again, we can see that this condition is restrictive showing EDGP instances that are not drawable without representing classes with separated regions (fig. 3.b).

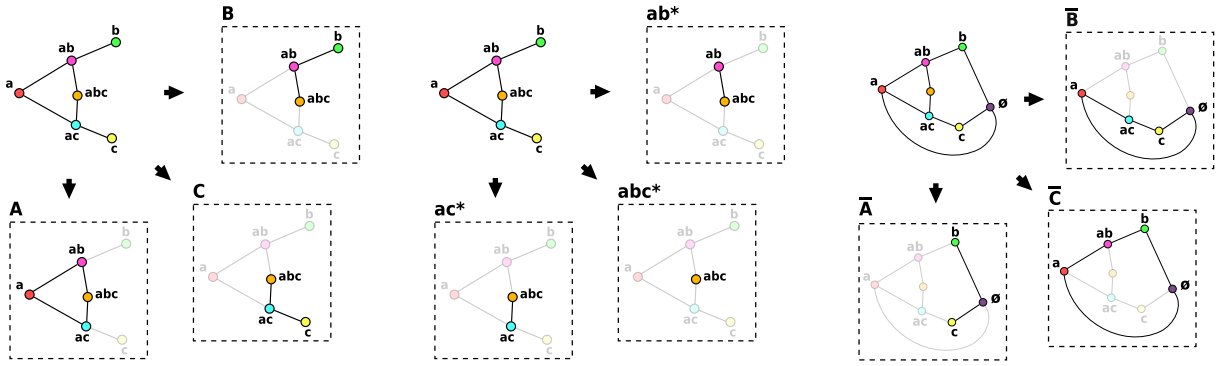
Relaxing this condition we are allowed to draw zones that are separated from each other (fig. 4.b). Clearly we are now able to draw each EDGP instance, as we are no more forced to link zones together.

Representation characteristics. From the previous analysis, we can deduce that Euler representations should allow classes to be represented by separated regions, if necessary. Disconnecting zones do not seem to be necessary, but sometimes they allow to obtain more readable diagrams. For the same reason we might also decide to duplicate a zone, creating a copy of the zone in another region of the graph and cloning all its elements.

Summarising, Euler representations are characterised by:

- classes not necessarily connected,
- zones not necessarily connected and eventually even cloned,

where the usage of these patterns is limited as much as possible.



(a) (b) (c)

Figure 7: Some tests on the intersection graph. (a) checking that all the class schemas are connected. (b) checking that subgraphs induced by the nodes containing any possible subset of classes are connected. Here the subgraphs induced by the nodes containing ab, ac, and abc are shown. Together with the ones containing a, b, and c (that correspond exactly to the class schemas in the previous picture), they are all the possible non empty subgraphs of this kind. (c) checking that the complementar class schemas are connected. At this point we need to consider also a node associated to the external area, that will always be part of the complementar class schemas.

Some examples of the application of these methods are shown in fig. 5 and fig. 6. In particular, fig. 5.a shows just a disconnected class, fig. 6.b also a disconnected zone, and fig. 5.c and fig. 6.c examples of zones cloning.

5 Properties of the intersection graph

Because of the bounds relaxation we did and the new structures we introduced, we have a high degree of freedom on representing diagrams. Choosing the more readable representation between all the possible ones requires at first to identify the most important properties of diagram comprehension. Assuming that Euler diagrams are more readable than general Euler representations, we need to try to:

1. avoid every undesired overlap. This is an indisputable point as we aim to draw just the non empty zones.
2. keep the classes as connected as possible. This will avoid having classes represented as separated regions (fig. 4.b) in the final diagram.
3. keep the single zones as connected as possible. This avoids zones being represented by more than one region (fig. 4.a), as it makes it difficult to understand the exact iteration of the zones with the rest of the diagram.
4. keep even zones that share the same subset of labels as connected as possible. This avoids disconnected overlaps between the same sets (fig. 1.b), as they make it difficult to trace how the intersection between classes is divided in the several zones.
5. avoid holes in the classes. Diagrams with holes (fig. 1.c) can generate confusion between holes and set inclusions.
6. make classes assume a smooth and regular shape.

As we will practically work with embeddings of the intersection graph, it is extremely useful to see how the previous diagram properties are translated in graph embedding properties:

1. make the intersection graph planar.
2. make the subgraphs induced by the nodes of the same class connected (fig. 7.a). We will call these induced subgraphs *class schemas*.
3. avoid node duplications in the intersection graph. In other words, limit the usage of node duplications in order to satisfy the previous points.
4. make the subgraphs induced by the nodes of all the same subset of classes connected, rather than just the nodes of the same class (fig. 7.b).
5. make the subgraph induced by the nodes outside each class schema connected (fig. 7.c). We will call these induced subgraphs *complementar class schemas*. It is also necessary to add a node associated to the null zone, corresponding to the external area. As this node is never part of the class schema, it is always in the complementar one.
6. place nodes in an area of the plan as compact and regular as possible.

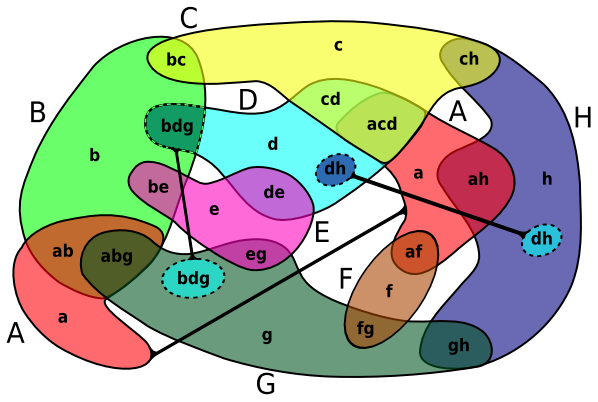


Figure 8: a Euler representation example. The diagram structure is not planar (it has a K_5 minor), so it cannot be represented with Euler diagrams. The Euler representation proposed uses zone duplication for bdg and dh, and has a disconnected class A.

Algorithms design. An algorithm that points to detecting a good Euler representation has to identify an intersection graph satisfying the previous points as much as possible, in order of importance. The most immediate way consists of identifying all the zones of the given classification, associating one intersection graph's node to each of them, and selecting carefully the edges to insert.

Node duplication, that corresponds to allow a zone to be disconnected, can be used when it is no longer possible to select useful edges in the graph. Disconnected class nodes will correspond, instead, to disconnected classes. Choosing to leave them disconnected, or to use node duplications to connect them, it is all matter of decision. As we saw, it depends on the specific case and on the specific relation one aims to represent.

6 Conclusions

We started by introducing several ways of defining Euler diagrams, showing or referencing proofs of their inability to represent every classification. We then analysed why Euler diagrams cannot be always drawn, pointing out two separat reasons that might impeding this process.

This analysis allowed us to detect some methods to show otherwise unrepresentable relationships. Using disconnected regions for classes, and graphically linking them together, is the simplest approach. We saw that this always works, but that the results are not necessarily the best possible. Another option we pointed out consists of representing some zones as disconnected regions. This might help to reduce the number of fictional links we need to introduce. A last possibility is to clone a whole zone in another part of the graph, cloning even the nodes of the zone. This helps in particular when we want to keep all the nodes of a class in the same connected region, even when the overlapping classes are not directly connected to each other.

Finally, we analysed the way each condition is expressed in the intersection graph. Structure graphs of this kind are the first step of most approaches to Euler diagrams generation. Knowing how the previous patterns are mapped in these graphs is essential to decide how, when and where to use them. An algorithm paradigm has also been pointed out, while concrete implementations of this approach need to conveniently define the necessary metrics according to the particular application.

References

[1] Gary D. Bader and Christopher W.V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. January 13 2003.

[2] Florence Benoy and Peter Rodgers. Evaluating the comprehension of euler diagrams. In *IV*, pages 771–780. IEEE Computer Society, 2007.

[3] Stirling Christopher Chow. *Generating and drawing area-proportional Euler and Venn diagrams*. PhD thesis, 2007.

[4] Leonhard Euler. Lettres une princesse d'allemagne, letters no. 102-108, 1761.

[5] Jean Flower and John Howse. Generating euler diagrams. *Lecture Notes in Computer Science*, 2317, 2002.

[6] D.S. Johnson and H.O. Pollak. Hypergraph planarity and the complexity of drawning venn diagrams. *Journal of graph theory*, 11(3):309–325, 1987.

[7] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[8] John Venn. On the diagrammatic and mechanical representation of propositions and reasonings, 1880.

[9] Anne Verroust and Marie-Luce Viaud. Ensuring the drawability of extended euler diagrams for up to 8 sets. In *Diagrammatic Representation and Inference, Third International Conference, Diagrams 2004, Cambridge, UK*, Lecture Notes in Computer Science. Springer.