Drawing Euler Diagrams with Circles and Ellipses

Gem Stapleton
University of Brighton, UK
g.e.stapleton@brighton.ac.uk

Peter Rodgers
University of Kent, UK
p.j.rodgers@kent.ac.uk

***Abstract*—The use of Euler diagrams as a basis for visual languages is commonplace and they are often used for visualizing information. The ability to automatically draw these diagrams is, therefore, likely to be of widespread practical use. The Euler diagram drawing problem is recognized as challenging, but the potential pay-off from the derivation of a comprehensive solution, that produces usable and effective diagrams, is significant. Previous research on automated Euler diagram drawing has used various different approaches, each of which had their own problems, including: (a) failure to draw a diagram in all cases, (b) poor diagram layout, and (c) inability to ensure that certain wellformedness properties of the drawn diagrams hold. In this paper, we present a novel approach to Euler diagram drawing that draws diagrams with circles, ellipses and curves in general. This new approach will draw a diagram in all cases, avoiding bad layout where possible (by the use of 'nice' geometric shapes) and can enforce wellformedness properties as chosen by the user.**

I. Introduction

Euler diagrams are syntactic components of a number of visual languages [1], [2], [3]. However, until recently, it has not been possible to automatically generate all required diagrams. The first work in automatic Euler diagram generation [4] only produced diagrams in a limited number of cases, those where certain so called *wellformedness properties* could be shown to hold. Other work restricted the shapes that can be drawn to circles, again at the expense of not drawing all diagrams [8] More recent work takes any abstract description and produces a diagram [5]. However, this often produces poor layout and leaves the user with little control over the wellformedness properties of the diagram.

The lack of effective general embedding methods for Euler diagrams severely restricts the use of visual languages that are based on them. Using hand drawn instances of such diagrams is cumbersome in all but the smallest examples, and it is not possible to automatically layout, transform or translate into such languages. Hence a general embedding tool that produces good layout would be of great benefit.

This paper improves on the state of the art by showing how circles can be drawn in diagrams including those which are not wellformed. We also show how to integrate ellipse based layout where a position for circles cannot be found. It is a general embedding method as it allows for arbitrary curves to be drawn when neither circles nor ellipses can be used. Our approach takes an inductive strategy, adding the required curves to the diagram one at a time. To add a curve we form the *hybrid graph* for the current diagram and examine this graph to choose the next curve to add, based on what shapes the candidate curves can take.

Section II introduces important concepts used throughout this paper. Section III-A presents techniques for adding a circle to a diagram. Section III-B devises methods for adding an ellipse to a diagram. In both the circles and ellipses cases, we demonstrate that the wellformedness properties of the original diagram are largely preserved under the addition of the new curve. In section III-C we demonstrate how to add curves in general, using a method similar to that of [7]. These techniques are combined into a new method to draw an Euler diagram in section IV. The approach prioritizes the drawing of circles and ellipses over arbitrary shaped curves, since these are aesthetically pleasing.

II. Definitions

An **Euler diagram** consists of labelled closed curves, where labels may be used for more than one curve. Since Euler diagrams convey information by the overlaps present between the curves, it is helpful to identify the regions to which the curves give rise. The smallest of these regions are called minimal regions: a **minimal region** is a connected component of the plane bounded by curve segments.

We extend the notion of minimal regions to **basic regions**, which are sets of minimal regions contained by some set of curves, but outside the rest of the curves. We can then go on to define a **zone**, which is a set of basic regions defined by curves with the same label sets [11].

A range of diagram properties have been defined, which are sometimes called wellformedness conditions:

1) All of the curves are simple (no curve self-intersects).
2) No pair of curves runs concurrently.
3) There are no triple points of intersection between the curves (there are no points that are mapped to more than twice by the curves).
4) Whenever two curves intersect, they cross.
5) Each basic region is connected (consists of exactly one minimal region).
6) Each curve label is used on at most one curve.

Definitions of these properties can be found in [9], except that for basic region connectedness (property 5) which is more typically stated for zones (a *connected zones* condition); for our purposes weakening the condition to basic regions is helpful. A further diagram property concerns the connectedness of the diagram. If all the curves form a connected component then we say the diagram is **atomic**, otherwise it is **nested** [10].

978-1-4577-1247-0/11/$26.00 ©2011 IEEE

Euler diagrams are associated with various graphs. In this paper, we are interested in three of these associated graphs, all of which are defined in [7]. First, we can take an Euler diagram, and construct its **Euler graph** which has a vertex at each point where two curves meet and the edges are the curve segments that connect the vertices.

A common concept in graph theory is that of a dual graph of a plane graph. From the Euler graph we can also form its dual, but we need a slightly more general notion, the **modified Euler dual**, which contains additional vertices and edges in the region 'outside' the curves. In particular, there is one extra vertex for each Euler graph edge next to the outside region and these vertices are connected by a loop that encloses the whole diagram.

Finally, we require the **hybrid graph**, which is formed from the Euler graph and the modified Euler dual by joining them together, placing a vertex wherever two edges cross and adding a triangulation. The triangulation edges join modified dual vertices to Euler graph vertices. A hybrid graph is shown on the right of figure 1.

Whilst we are primarily interested in the hybrid graph, we want to be able to talk about the graphs from which it arose. To this end, we say that the vertices (resp. edges) of a hybrid graph that arose from the Euler graph are **Euler vertices**, those which arose from the modified Euler dual are **dual vertices** and the other vertices are **new vertices** (resp. edges).

Of fundamental significance to the drawing problem is the notion of an abstract description of a diagram. These descriptions capture the overlaps that are to be present between the curves and, thus, capture the semantics of the diagram. However, as these descriptions do not carry with them any information about the actual layout of the diagram; the task of Euler diagram generation is that of laying out the curves in the diagram. Our diagram descriptions thus comprise a set of **abstract curves**, which are labelled. Note that these elements are not curves in any concrete sense. In addition, a description includes a set of **abstract basic regions**; each abstract basic region is a set of abstract curves.

III. Adding Curves To Diagrams

Given an Euler diagram, d, with diagram description D, we now present a method to add a curve to d, given a description of how to add an abstract curve to D. In order to add a new abstract curve, to D, we need to describe how the curve impacts on the abstract basic regions. The impact on each abstract basic region, is captured by saying that either it is to be inside the curve, outside the curve or 'split' by the curve. The impact on the abstract basic regions can then be characterized by two sets: in and out, where the abstract basic regions inside or split are in in and those outside or split are in out.

Of course, we do not want to add arbitrary curves, we want to add curves that have a particular impact on the basic regions. Given the sets in and out we can identify how to add an appropriate curve to d using the hybrid graph, provided d is atomic. Thus, given an arbitrary atomic diagram, d, we first construct its Euler graph and then the dual of the Euler graph. If d is nested then d is transformed into an atomic diagram, d', by moving the nested parts until they just touch. Then we can use the hybrid graph of the obtained atomic diagram to add a curve. After the curve addition, this can be reversed to return the diagram to its original nested state, modulo the new curve addition; see [11] for graph transformations used in the context of Euler diagrams.

A. Adding Circles

We can add a circle to the diagram centred on any vertex in the hybrid graph. Whilst the techniques we describe do not capture all of the ways in which a circle can be placed in the diagram, the techniques are more general than [6], which only considered completely wellformed Euler diagrams.

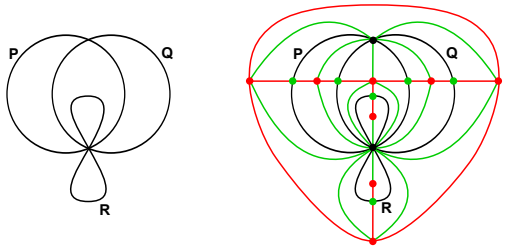

Fig. 1. An Euler diagram and its hybrid graph.

To illustrate the idea, we consider adding circles in three different ways to the diagram on the right of figure 1. First, we will consider an in set consisting of one basic region, $in = \{P\}$ and an out consisting of all the basic regions in the diagram. Since in contains exactly one abstract basic region, we know that we can simply draw an appropriate circle in the corresponding basic region; such a circle can be constructed by centering it on a suitable dual vertex, as shown on the left of figure 2.

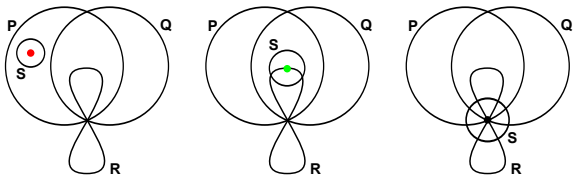

Fig. 2. Three ways of adding circles.

If, instead, $in = \{PQ, PQR\}$, and out as before, then we can add a circle around a new vertex as shown in the middle of figure 2. Here, in contains two abstract basic regions. To be able to add a circle in the appropriate manner, the two corresponding basic regions in the drawn diagram must be separated by an Euler edge. If this is the case then there will be a new vertex that the circle can be drawn around.

As a final example, taking in as all the basic regions in the diagram, we can add a circle around the Euler graph vertex placed on all three curves, as shown on the right of figure 2. Since in is large in this case (i.e. contains more than two abstract basic regions) we cannot add a circle around a new vertex or a dual vertex.

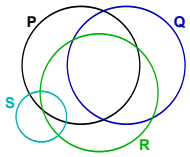

Fig. 3. A diagram to which we wish to add an ellipse.

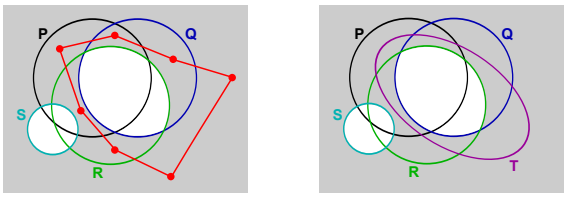

Fig. 4. A cycle in the modified Euler dual and an ellipse that results.

Finally, we note that adding a circle using this technique preserves the wellformedness properties of the original diagram, provided the label of the new circle is not already present.

B. Adding Ellipses

To add an ellipse, we utilize the modified Euler dual. In particular, we find a simple cycle in it that is a potential route for the ellipse. To illustrate the idea, consider the diagram in figure 3. Take $in = \{\emptyset, P, Q, PQ, R, PR, QR, PQR\}$ and $out = \{\emptyset, P, Q, PQ, R, PR, QR, S, PS, RS, PRS\}$. We can find a simple cycle in the modified dual that passes through precisely the basic regions to be split (i.e. those in $in \cap out$); such a cycle is shown on the left of figure 4. This cycle captures a route for a new curve that will result in a diagram with the required abstraction. The shading indicates the range of paths that such a curve can take: the cycle can be re-routed, provided it stays within the shaded area and the vertices do not leave the minimal regions in which they are placed. In addition the edges must cross the same Euler edges after the re-routing as they did before the re-routing. Using these observations, we can see that if it is possible to add an ellipse, it must pass through this shaded area, passing through each basic region in the area, and passing through each minimal region at most once. An appropriate ellipse is shown on the right of figure 4.

Once we have found an appropriate simple cycle, should one exist, we can observe that the wellformedness properties are maintained (as with circles), if the label of the new curve is not already in the diagram. Further, the cycle should not pass through the region outside all of the curves more than once. If it does then the basic region inside just the ellipse will be disconnected.

The problem remains of how to identify whether an ellipse can be drawn through the allowed region. A search based method could explore different values for the centre, rotation and axis length of potential ellipses. A fitness function based on the amount of overlap between the ellipse and the disallowed region would mean that zero fitness implies that there is an ellipse that will fit.

Such a search method can be also be adapted to find a 'good' ellipse, if the search twice is run twice, each time with an adjusted fitness function. On one occasion searching for the smallest fitting ellipse, and on the other, searching for the largest fitting ellipse. Taking an average ellipse of the two is likely to produce a result that does not pass too close to regions it does not cross. Moreover, taking account of other good features of the ellipse, such as rotation aligned to 90 degree angles would also be feasible. We also note that this method can also be used to draw circles that cannot be added using the techniques of section III-A if making major and minor axes equal is prioritized.

C. Adding Curves in General

If the methods in the above sections for adding circles and ellipses fail to add a curve in the required manner then we can use the hybrid graph to find a route for the curve, which can always be done. This was previously considered in [7], so we just sketch the ideas here. We add the curve by finding a suitable closed sequence of edges in the hybrid graph, so that the correct basic regions contained by the new curve, outside the new curve, or split by the new curve. The properties of the selected path impact on the wellformedness properties of the drawn diagram; again, this was detailed in [7].

IV. Drawing Entire Descriptions

So far, we have demonstrated how to add curves to a diagram in a variety of ways. A more general problem is determining how to draw a diagram d given an arbitrary diagram description, D. For our approach, we need to add curves to d in a way that ensures the final result represents D.

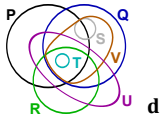

Fig. 5. A diagram to be drawn.

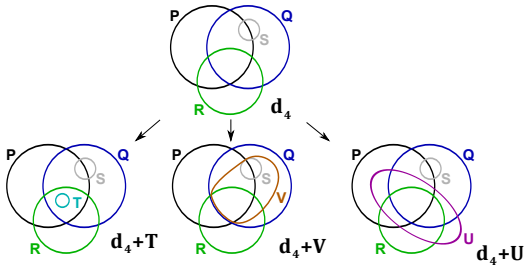

Fig. 6. Choosing a curve to add.

Deriving the in and out for adding a curve to a partially constructed diagram can always be achieved [7]. Hence, the process of creating an Euler diagram is that of successively adding curves to build up the diagram. For a given diagram, a curve can be added by:

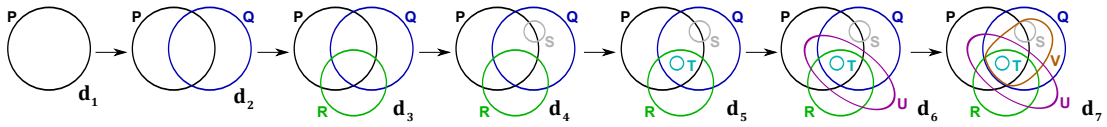

Fig. 7. The drawing sequence.

1) Determine whether one of the currently undrawn curves can be added as a circle using the techniques of section III-A. If so, add an appropriate circle, otherwise proceed to the next step.
2) Determine whether one of the currently undrawn curves can be added as an ellipse using the techniques of section III-B. If so, add an appropriate ellipse, otherwise proceed to the next step.
3) Finally, use the techniques outlined in section III-C to add one of the undrawn curves with an arbitrary shape.

This process iterates, starting with an empty diagram and adding an abstract curve at each iteration until all the curves are in the diagram. Clearly, there are various strategies that can be applied when more than one curve can be drawn at the same stage, but these are not detailed here due to space limitations.

There is good evidence that people like Euler diagrams drawn with circles, which is why we prioritise them; for example Wilkinson [12], identifies that out of the 72 Euler diagrams used in articles appearing in Science, Nature and online affiliated journals during 2009, 65 (90%) use circles.

A. Example

To illustrate the drawing process, suppose we start with description, D, of the diagram, d, in figure 5. Of course, here we have already drawn d, but this drawing is merely given to help illustrate the drawing approach, rather than the other option of writing down its description in full. We will work through one curve addition in detail, in the middle of the drawing process, making reference to figures 6 and 7.

Suppose that we have reached a point where diagrams d_1, d_2, d_3 and d_4 have been drawn, as in figure 7. Then there remain three curves to be drawn, namely T, U, and V. Figure 6 shows possible drawings of the resultant diagrams. We can see that T can be added as a circle, so we choose to add this next, giving d_5 in figure 7. To d_5, neither U nor V can be added using the circle drawing methods of section III-A, so we see whether either can be added as an ellipse. We see that U can be added to d_5 as an ellipse using the methods of section III-B, which gives d_6. Finally, we add U (which cannot be added as a circle or ellipse) by finding a suitable closed sequence of edges in the hybrid graph.

V. Conclusion

This paper presents a novel method for drawing Euler diagrams using circles and ellipses, as well as arbitrary curves where necessary. The method is inductive, adding one curve at a time to the diagram. The approach we have taken to identify an order in which to draw the curves prioritizes drawing circles and ellipses, since these are aesthetically pleasing, over using arbitrary curves. Moreover, the techniques we presented for adding circles and ellipses largely ensure that the wellformedness properties of the original diagram are preserved. Even in the case of adding arbitrary curves, we can sensibly choose a closed sequence of edges in the hybrid graph, which form the route that the to-be-added curve will follow, so that we avoid breaking certain wellformedness properties. The approach is sufficiently general that all descriptions can be drawn.

Future work will involve extending the techniques so that circles and ellipses can be added in more general ways, allowing wellformedness conditions to be broken in preference to using a less pleasing shaped curve. Alternative shapes, such as rectangles or ovals can also be added to diagrams with this technique. The different routes that a curve may follow can profoundly impact on the ability of users to accurately and effectively interpret the drawn diagrams. Thus, empirical studies need to be undertaken so that any choice of curve routing is fully informed.

.

References

[1] J. Gil, J. Howse, and S. Kent, "Formalising spider diagrams," in *IEEE VL 1999*. , pp. 130–137.
[2] S. Kent, "Constraint diagrams: Visualizing invariants in object oriented modelling," in *OOPSLA97*. ACM Press, 1997, pp. 327–341.
[3] S.-J. Shin, *The Logical Status of Diagrams*. Cambridge University Press, 1994.
[4] J. Flower and J. Howse, "Generating Euler diagrams," in *Diagrams 2002*. Springer, pp. 61–75.
[5] P. Rodgers, L. Zhang, and A. Fish, "General Euler diagram generation," in *Diagrams 2008*. Springer, pp. 13–27.
[6] G. Stapleton, L. Zhang, J. Howse, and P. Rodgers, "Drawing Euler diagrams with circles: The theory of piercings," *IEEE Trans. on Visualisation and Computer Graphics*, vol. 17, no. 7, pp. 1020-0132, 2011.
[7] G. Stapleton, P. Rodgers, J. Howse, and L. Zhang, "Inductively generating Euler diagrams," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, no. 1, pp. 88–100, 2011.
[8] H. Kestler, A. Muller, T. Gress, and M. Buchholz, "Generalized Venn diagrams: A new method for visualizing complex genetic set relations," *Bioinformatics*, vol. 21, no. 8, pp. 1592–1595, 2005.
[9] G. Stapleton, P. Rodgers, J. Howse, and J. Taylor, "Properties of Euler diagrams," in *LED 2007*. EASST, pp. 2–16.
[10] J. Flower, J. Howse, and J. Taylor, "Nesting in Euler diagrams: syntax, semantics and construction," *Software and Systems Modelling*, vol. 3, pp. 55–67, 2004.
[11] J. Howse, P. Rodgers, and G. Stapleton, "Changing Euler diagram properties by Edge transformation of Euler dual graph," in *IEEE VL/HCC*, 2009, pp. 177–184.
[12] L. Wilkinson, Exact and Approximate Area-Proportional Circluar Venn and Euler Diagrams, IEEE Transactions on Visualization and Computer Graphics, 2011. doi: 10.1109/TVCG.2011.56.