

# Drawing Euler Diagrams with Circles

Gem Stapleton<sup>1</sup>, Leishi Zhang<sup>2</sup>, John Howse<sup>1</sup>, and Peter Rodgers<sup>2</sup>

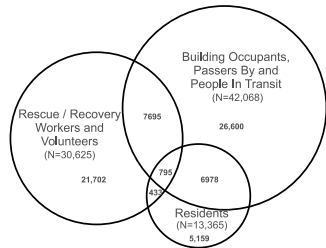
<sup>1</sup> Visual Modelling Group, University of Brighton, UK  
{g.e.stapleton,john.howse}@brighton.ac.uk  
<sup>2</sup> University of Kent, Canterbury, UK  
{l.zhang,p.j.rodgers}@kent.ac.uk

**Abstract.** Euler diagrams are a popular and intuitive visualization tool which are used in a wide variety of application areas, including biological and medical data analysis. As with other data visualization methods, such as graphs, bar charts, or pie charts, the automated generation of an Euler diagram from a suitable data set would be advantageous, removing the burden of manual data analysis and the subsequent task of drawing an appropriate diagram. To this end, various methods have emerged that automatically draw Euler diagrams from abstract descriptions of them. One such method draws some, but not all, abstract descriptions using only circles. We extend that method so that more abstract descriptions can be drawn with circles. Furthermore, we show how to transform any ‘undrawable’ abstract description into a drawable one. Thus, given any abstract description, our method produces a drawing using only circles. A software implementation of the method is available for download.

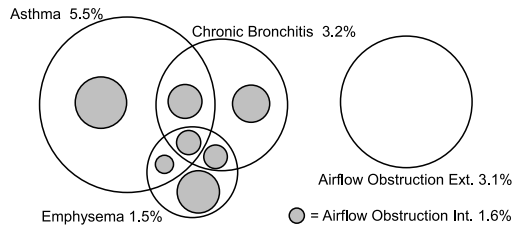
## 1 Introduction

It is commonly the case that data can be more easily interpreted using visualizations. One frequently sees, for instance, pie charts used in statistical data analysis and graphs used for representing network data. These visualizations are often automatically produced, allowing the user to readily make interpretations that are not immediately apparent from the raw data set. Sometimes, the raw data are classified into sets and one may be interested in the relationships between the sets, such as whether one set is a subset of another or whether one set contains more elements than another.

For example, the authors of [6] have data concerning health registry enrollees at the world trade centre. Each person in the health registry is classified as being in one or more of three sets: rescue/recovery workers and volunteers; building occupants, passers by, and people in transit; and residents. In order to visualize the distribution of people amongst these three sets, the authors of [6] chose to use an Euler diagram which can be seen in figure 1. A further example, obtained from [16], shows a visualization of five sets of data drawn from a medical domain. The authors of [16] chose to represent one of the sets (Airflow Obstruction Int) using multiple curves. Other areas where Euler diagrams are used for information visualization include crime control [7], computer file organization [4], classification systems [20], education [10], and genetics [12].



**Fig. 1.** Data visualization using an Euler diagram.



**Fig. 2.** Using multiple circles to represent a set.

As with other diagram types for data visualization, the ability to automatically create Euler diagrams from the data would be advantageous. To date, a range of methods for automatically drawing Euler diagrams have been developed, with most of them starting with an abstract description of the required diagram. The existing methods can be broadly classified into three classes.

**Dual Graph based methods:** With these methods, a so-called dual graph of the required Euler diagram is identified and embedded in the plane. Then the Euler diagram is formed from the dual graph. Methods in this class include the first Euler diagram drawing technique, attributable to Flower and Howse [8]. Others who have developed this class of drawing method include Verroust and Viaud [22], Chow [2], and Simonetto et al. [15]. Recently, Rodgers et al. have developed a general dual graph based method that is capable of drawing a diagram given any abstract description [13]. Some of these methods allow the use of many curves to represent the same set, to ensure drawability (as in figure 2).

**Inductive Methods:** Here, one curve of the required Euler diagram is drawn at a time, building up the diagram as one proceeds. This is a recently devised method, attributable to Stapleton et al. [18], and builds on similar work for Venn diagrams [5, 21]. Stapleton et al.’s method is also capable of drawing a diagram given any abstract description and it has advantages over the dual graph based methods in that it readily incorporates user preference for properties that the to-be-drawn diagram is to possess.

**Methods using Particular Shapes** A large number of methods attempt to draw Euler diagrams using particular geometric shapes, typically circles, because they are aesthetically pleasing. Chow considers drawing diagrams with exactly two circles [2], which is extended to three circles by Chow and Rodgers [3]. The Google Charts API includes facilities to draw Euler diagrams with up to three circles [1] and Wilkinson’s method allows any number of circles but it often fails to produce diagrams with the specified abstract description [23]; Wilkinson’s diagrams can contain too few zones and, thus, fail to convey the correct semantics. Similarly, Kestler et al. devised a method that draws Euler diagrams with regular polygons but it, too, does not guarantee that the diagrams have the required zones [11]. In previous work, we have devised a method for drawing a particular class of abstract descriptions with circles, which does ensure the

correct abstraction is achieved [19]. None of these methods is capable of drawing an Euler diagram given an arbitrary abstract description. In part, this is because many abstract descriptions are not drawable with a circles or regular polygons, given the constraints imposed by the authors on the properties that the diagrams are to possess (such as no duplicated curve labels). However, these methods often produce aesthetically pleasing diagrams.

In this paper, we take the method of [19] and extend it, so that every abstract description is (essentially) drawable at the cost of representing sets with more than one curve (as in figure 2). Our method takes the abstract description and draws a diagram with circles that contains all required zones, but may contain additional zones; any extra zones are shaded. Section 2 presents necessary background material on Euler diagrams, along with some new concepts that are particular to the work in this paper. Abstract descriptions are defined in section 3 and we provide various definitions of abstract-level concepts. Section 4 describes the class of inductively pierced abstract descriptions developed in [19], on which the results in this paper build. Our drawing method is described in section 5. Section 6 shows some output from the software implementation of the method, alongside diagrams drawn using previously existing methods.

## 2 Euler Diagrams

An Euler diagram is a set of closed curves drawn in  $\mathbb{R}^2$ . Each curve has a label chosen from some fixed set of labels,  $\mathcal{L}$ . Our definition of an Euler diagram is consistent with, or a generalization of, those found in the literature, such as in [2, 8, 17, 22]. An **Euler diagram** is a pair,  $d = (Curve, l)$ , where

1.  $Curve$  is a finite set of closed curves in  $\mathbb{R}^2$ , and
2.  $l: Curve \rightarrow \mathcal{L}$  is a function that returns the label of each curve.

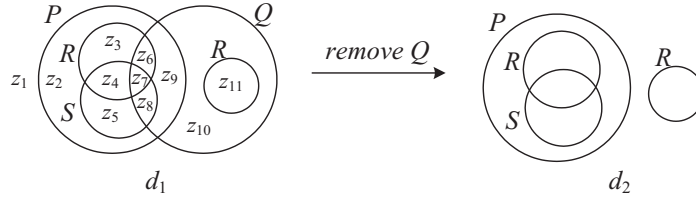
A **minimal region** of  $d$  is a connected component of

$$\mathbb{R}^2 - \bigcup_{c \in Curve} image(c)$$

where  $image(c)$  is the set of points in  $\mathbb{R}^2$  to which  $c$  maps. We define the set of curves in a diagram with some specified label,  $\lambda$ , to be a **contour** with label  $\lambda$ . The diagram  $d_1$  in figure 3 has four contours, but five curves. A point,  $p$ , is inside a contour precisely when the number of the contour's curves that  $p$  is inside is odd. Another important concept is that of a **zone**, which is a set of minimal regions that can be described as being inside certain contours (possibly none) and outside the rest of the contours. The diagram  $d_1$  in figure 3 has 11 zones, each of which is a minimal region.

There are a collection of properties that it is desirable for Euler diagrams to possess, since they are often thought to correlate with the ease with which the diagrams can be interpreted. The most commonly considered properties are:

1. **Unique Labels:** no curve label is used more than once.



**Fig. 3.** Euler diagram concepts.

2. **Simplicity:** all curves are simple (have no self-intersections).
3. **No Concurrency:** the curves intersect at a discrete set of points (i.e. no curves run along each other in a concurrent fashion).
4. **Only Crossings:** whenever two curves intersect, they cross.
5. **No 3-points:** there are no 3-points of intersection between the curves (i.e. any point in the plane is passed through at most 3 times by the curves).
6. **Connected Zones:** each zone consists of exactly one minimal region.

A diagram,  $d$ , possessing all of these properties is **completely wellformed**. Neither diagram in figure 3 is completely wellformed, since both use the curve label  $R$  twice and, thus, in each diagram the set  $R$  is represented by more than one curve. Now,  $d$  is **completely wellformed up to labelling** if it possesses all properties except, perhaps, the unique labels property. If all of the curves in  $d$  are circles then  $d$  is **drawn with circles**. Our drawing method only produces diagrams drawn with circles that are completely wellformed up to labelling.

Further concepts that we need concern the topological adjacency of zones and ‘clusters’ of topologically adjacent zones. We define these concepts only for diagrams that are completely wellformed up to labelling, since this is sufficient for our purposes. In particular, in such diagrams we know that two zones which are topologically adjacent are separated by a single curve. For example, in figure 3, the zones  $z_2$  and  $z_3$  are topologically adjacent in  $d_1$ , separated by the leftmost curve labelled  $R$ ; when this curve is removed,  $z_2$  and  $z_3$  form a minimal region. The zones  $z_6$  and  $z_{11}$  are not topologically adjacent and neither are  $z_2$  and  $z_4$ .

Let  $z_1$  and  $z_2$  be zones in  $d = (Curve, l)$ . If there exists a curve,  $c$ , in  $Curve$  such that  $z_1$  and  $z_2$  form a minimal region in the diagram  $(Curve - \{c\}, l - \{(c, l(c))\})$  then  $z_1$  and  $z_2$  are **topologically adjacent** in  $d$  **separated** by  $c$ . Regarding our drawing problem, we could choose to draw a circle that splits two adjacent zones and which intersects their separating curve. We call topologically adjacent zones  $z_1$  and  $z_2$  a **cluster** given  $c$ . We also define a cluster comprising four zones. Let  $c_1$  and  $c_2$  be distinct curves in  $d$ , that intersect at some point  $p$ . The four zones in the immediate neighbourhood of  $p$  (since we are assuming wellformedness up to labelling, precisely four such zones exist) form a **cluster** given  $c_1$ ,  $c_2$  and  $p$ , denoted  $C(c_1, c_2, p)$ . In figure 3, the zones  $z_3$ ,  $z_4$ ,  $z_6$  and  $z_7$  form a cluster given  $Q$  and  $S$  (blurring the distinction between the curves and their labels). Given a cluster of four zones, we can draw a circle around the point  $p$  that splits all and only these zones.

### 3 Abstract Descriptions

As with typical Euler diagram drawing methods, we start with an abstract description of the required diagram. This description tells us which zones are to be present. An **abstract description**,  $D$ , is a pair,  $(L, Z)$ , where

1.  $L$  is a finite subset of  $\mathcal{L}$  (i.e. all of the labels in  $D$  are chosen from the set  $\mathcal{L}$ ) and we define  $L(D) = L$ ,
2.  $Z \subseteq \mathbb{P}L$  such that  $\emptyset \in Z$  and for each  $\lambda \in L$  there is a zone,  $z$ , in  $Z$  where  $\lambda \in z$  and we define  $Z(D) = Z$ .

The abstract description,  $D$ , of  $d_2$  in figure 3 has labels  $\{P, R, S\}$  and zones  $\{\emptyset, \{P\}, \{R\}, \{P, R\}, \{P, S\}, \{P, R, S\}\}$ ; we say that  $d_2$  is a *drawing* of  $D$ . We will sometimes abuse notation, omitting the label set and writing the zone set as, for instance,  $\{P, R, PR, PS, PRS\}$ .

It is not possible to identify whether two zones will necessarily be topologically adjacent when presented only with an abstract description. However, we can observe that, in a diagram that does not possess any concurrency, two zones that are topologically adjacent have abstractions that differ by a single curve label. For example, the topologically adjacent zones  $z_2$  and  $z_3$  in figure 3 have abstractions  $\{P\}$  and  $\{P, R\}$  which differ by  $R$ , the label of their separating curve. We use this observation to define an abstract notion of a cluster. Let  $z$  be an abstract zone (i.e. a finite set of labels) and let  $A \subseteq \mathcal{L}$  be a set of labels disjoint from  $z$ . The set  $\{z \cup A_i : A_i \subseteq A\}$  is a **A-cluster** for  $z$ , denoted  $\mathcal{C}(z, A)$ . The cluster  $\mathcal{C}(\{P, Q\}, \{Q, S\}, d_1)$  is the cluster  $\{PR, PQR, PRS, PQRS\}$  and corresponds to the cluster  $\{z_3, z_4, z_6, z_7\}$  in  $d_1$ , in figure 3. In general, a set of zones in a diagram that form a cluster will have abstractions that form a cluster. However, a set of zones may have abstractions that form a cluster but need not themselves be a cluster in the drawn diagram. For example,  $z_6$  and  $z_{11}$ , figure 3, do not form a cluster but their abstractions,  $\{R, Q\}$  and  $\{P, R, Q\}$ , do form a cluster.

Further abstract level concepts are useful to us. Our drawing method first draws curves that are not contained by any other curves and ‘works inwards’ drawing contained curves later in the process. We can identify at the abstract level whether a contour,  $C_1$ , is to be contained by another,  $C_2$ , and, as such, in any drawing  $C_2$ ’s curves will each be contained by at least one of  $C_1$ ’s curves. We are also interested in which abstract zones are contained by which curve labels.

Let  $D = (L, Z)$  be an abstract description and let  $\lambda_1$  and  $\lambda_2$  be distinct curve labels in  $L$ . If  $\lambda_1 \in z$  and  $z \in Z$  then we say  $\lambda_1$  **contains**  $z$  in  $D$  with the set of such zones denoted  $Z_c(\lambda_1)$ . If  $Z_c(\lambda_1) \subset Z_c(\lambda_2)$  then  $\lambda_2$  **contains**  $\lambda_1$  in  $D$ . The set of curves that contain  $\lambda_1$  in  $D$  is denoted  $L^c(\lambda_1)$ . In the abstract description (given above) for  $d_2$  of figure 3, the curve label  $P$  contains the curve label  $S$  but not the curve label  $R$ . This reflects the fact that, in  $d_2$ , the contour labelled  $P$  does not contain the contour labelled  $R$ .

We need an operation to remove curve labels from abstraction descriptions. Given an abstract description,  $D = (L, Z)$ , and  $\lambda \in L$ , we define  $D - \lambda$  to be  $D - \lambda = (L - \{\lambda\}, \{z - \{\lambda\} : z \in Z\})$ . The abstract description for  $d_1$  in figure 3

becomes the abstract description for  $d_2$  on the removal of  $Q$ . A **decomposition** of  $D$  is a sequence,  $dec(D) = (D_0, D_1, \dots, D_n)$  where each  $D_{i-1}$  ( $0 < i \leq n$ ) is obtained from  $D_i$  by the removal of some label,  $\lambda_i$ , from  $D_i$  (so,  $D_{i-1} = D_i - \lambda_i$ ) and  $D_n = D$ . The description  $D_0$  is called a **subdescription** of  $D_n$ . If  $D_0$  contains no labels then  $dec(D)$  is a **total decomposition**.

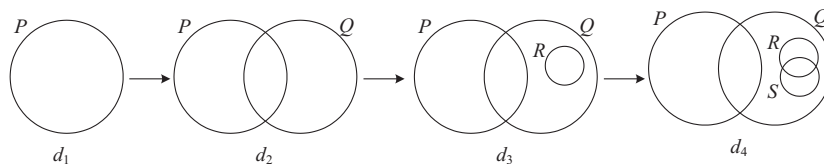
## 4 Inductively Pierced Descriptions

A class of abstract descriptions that can be drawn with circles in a completely wellformed manner can be built by successively adding *piercing curves*. Figure 4 shows a sequence of diagrams where, at each stage, the curve added is a piercing curve. This section summarizes results in [19] and adds a new concept of an inductively pierced diagram. The following definition is generalized from [19].

**Definition 1.** Let  $D = (L, Z)$  be an abstract description. Let  $\lambda_1, \lambda_2, \dots, \lambda_{n+1} \in L$  be distinct curve labels. Then  $\lambda_{n+1}$  is an ***n*-piercing** of  $\lambda_1, \dots, \lambda_n$  in  $D$  if there exists a zone,  $z$ , such that

1.  $\lambda_i \notin z$  for each  $i \leq n + 1$
2.  $Z_c(\lambda_{n+1}) = \mathcal{C}(z \cup \{\lambda_{n+1}\}, \{\lambda_1, \dots, \lambda_n\})$ , and
3.  $\mathcal{C}(z, \{\lambda_1, \dots, \lambda_n\}) \subseteq Z$ .

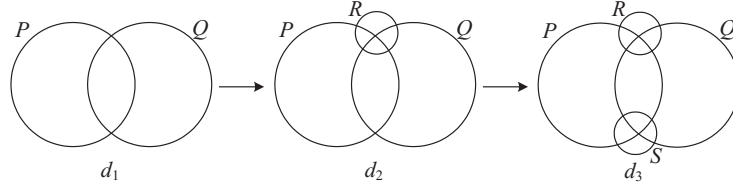
The zone  $z$  is said to **identify**  $\lambda_{n+1}$  as a piercing.



**Fig. 4.** An inductively pierced diagram.

In figure 4, the curve  $S$  is a 1-piercing of  $R$  in  $d_4$ . If an abstract description can be built by successively adding 0-piercing, 1-piercing, or 2-piercing curves then, usually, it can be drawn with circles in a completely wellformed manner. However, there are occasions when this is not possible. For example, in figure 5, we may want to add a curve,  $T$ , to  $d_3$  that is a 2-piercing of  $P$  and  $Q$ . However, it is not possible to do so using a circle whilst maintaining wellformedness. Thus, the definition of an inductively pierced description, which allows only 0, 1, or 2-piercings, restricts the ways in which 2-piercings can arise.

**Definition 2.** Let  $\mathcal{C}_1 = \mathcal{C}(z, \{\lambda_1, \lambda_2\})$  and  $\mathcal{C}_2 = \mathcal{C}(z \cup \{\lambda_3\}, \{\lambda_1, \lambda_2\})$  be clusters. Let  $D = (L, Z)$  be an abstract description. If  $\mathcal{C}_1 \cup \mathcal{C}_2 \subseteq Z$  then  $\lambda_3$  is **outside-associated** with  $\mathcal{C}_2$  in  $D$  and is **inside-associated** with  $\mathcal{C}_1$  in  $D$ .



**Fig. 5.** Adding three 2-piercing curves.

**Definition 3.** Let  $D = (L, Z)$  be an abstract description. Then  $D$  is **inductively pierced** if either

1.  $D = (\emptyset, \{\emptyset\})$ , or
2.  $D$  has a 0-piercing,  $\lambda$ , such that  $D - \lambda$  is inductively pierced, or
3.  $D$  has a 1-piercing,  $\lambda$ , such that  $D - \lambda$  is inductively pierced, or
4.  $D$  has a 2-piercing,  $\lambda_3$ , of  $\lambda_1$  and  $\lambda_2$  identified by  $z$ , and either
  - (a) no other curve label,  $\lambda_4$ , in  $D$  is outside-associated with  $\mathcal{C}(z, \{\lambda_1, \lambda_2\})$  or
  - (b) exactly one other curve label,  $\lambda_4$ , in  $D$  is outside-associated with  $\mathcal{C}(z, \{\lambda_1, \lambda_2\})$  and we have either
    - i.  $L^c(\lambda_3) = L^c(\lambda_4) = L^c(\lambda_1)$  or
    - ii.  $L^c(\lambda_3) = L^c(\lambda_4) = L^c(\lambda_2)$ .
 and  $D - \lambda_3$  is inductively pierced.

All of the diagrams in figures 4 and 5 have inductively pierced descriptions whereas the diagram  $d_1$  in figure 3 does not.

**Definition 4.** A diagram,  $d$ , is **inductively pierced** if either  $d$  contains no curves or the following hold:

1.  $d$  is drawn entirely with circles,
2.  $d$  is completely wellformed,
3. given any pair of abstract zones,  $z_1$  and  $z_2$ , in  $d$ 's abstraction,  $D$ , if the symmetric difference of  $z_1$  and  $z_2$  contains exactly one label,  $\lambda$ , then in  $d$  the zones with abstractions  $z_1$  and  $z_2$  are topologically adjacent, separated by the curve labelled  $\lambda$ , and
4. there is a circle,  $c$ , whose label is an  $i$ -piercing ( $i \leq 2$ ) in the abstraction,  $D$ , of  $d$ , and the diagram obtained from  $d$  by removing  $c$  is inductively pierced.

The diagrams in figures 4 and 5 are inductively pierced. However, the diagram  $d_2$  in figure 3 has an inductively pierced abstract description but  $d_2$  itself is not inductively pierced; it can be redrawn in an inductively pierced manner.

**Theorem 1.** Let  $D$  be an inductively pierced abstract description. Then there exists an inductively pierced drawing,  $d$ , of  $D$ . Moreover such a  $d$  can be drawn in polynomial time, [19].

Presented in [19] is a detailed algorithm to draw  $d$  given  $D$ , as in theorem 1. A total, decomposition,  $dec(D) = (D_0, \dots, D_n)$  is an **inductively pierced decomposition** if every  $D_i$  is an inductively pierced abstract description and is obtained from  $D_{i+1}$  by the removal of a piercing curve.

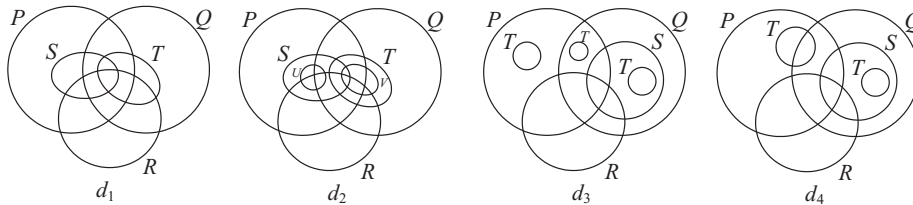
## 5 Drawing with Circles

We will now demonstrate how to turn an arbitrary abstract description into another abstract description that can be drawn in an inductively pierced manner, except that it may have duplicated curve labels. A diagram is **inductively pierced up to curve relabelling** if there exists a relabelling of its curves so that the curve labels are unique and the resulting diagram is inductively pierced. The diagram  $d_2$  in figure 3 is inductively pierced up to curve relabelling. In addition,  $d_1$  is also inductively pierced up to curve relabelling but, unlike  $d_2$ , its abstract description is not inductively pierced.

It is helpful to summarize the initial stages our drawing process. We take an abstract description,  $D$ , and find a total decomposition,  $dec(D) = (D_0, \dots, D_n)$  of  $D$ . At least one of the  $D_i$ s is an inductively pierced subdescription of  $D_n$  (for instance,  $D_0$  is inductively pierced). We can draw such a  $D_i$ , yielding  $d_i$ , using the methods of [19] which draws  $D_i$  by adding an appropriate circle to the drawing of  $D_{i-1}$ . Once we reach the first  $D_j$  which is not inductively pierced, we start to draw contours consisting of more than one circle. We will address how to choose sensibly a decomposition and how to add the remaining contours to  $d_{j-1}$  in order to obtain  $d$ . We point the reader to subsection 5.4, which includes a comprehensive illustration of our drawing method.

### 5.1 Choosing a Decomposition

There are choices about the order in which the curve labels are removed when producing a decomposition of an abstract description and we prioritize removing curve labels that do not contain other curve labels.



**Fig. 6.** Choosing a decomposition.

**Definition 5.** Let  $D = (L, Z)$  be an abstract description that contains curve label  $\lambda$ . We say that  $\lambda$  is **minimal** if  $\lambda$  does not contain any curve labels in  $D$ .

In figure 6,  $d_1$ 's abstract description has minimal curve labels  $R$ ,  $S$  and  $T$ , whereas for  $d_2$  the minimal labels are  $R$ ,  $U$  and  $V$ . Trivially, every abstract description,  $D$  (with  $L(D) \neq \emptyset$ ), contains at least one minimal curve label and, moreover, every piercing curve is minimal. When producing a decomposition,



our method removes a minimal curve label at each step. This ensures that, when we draw the diagram (the process for which is described later), if curve label  $\lambda_1$  is contained by curve label  $\lambda_2$  then the contour,  $c_1$ , for  $\lambda_1$  will be drawn inside the contour,  $c_2$ , for  $\lambda_2$ . This nicely reflects the semantics of the diagram: if  $\lambda_1$  represents a proper subset of  $\lambda_2$  then  $c_1$  will be contained by  $c_2$ .

**Definition 6.** Let  $D = (L, Z)$  be an abstract description. To produce a **chosen total decomposition** of  $D$  carry out the following steps:

1. Set  $i = n$ , where  $|L(D)| = n$  and define  $D = D_i$  and  $dec_i(D) = (D)$ .
2. Identify a minimal curve label,  $\lambda$ , in  $D$ .
3. Remove  $\lambda$  from  $D_i$  to give  $D_{i-1}$ .
4. Form  $dec_{i-1}(D)$  by copying  $dec_i(D)$  and placing  $D_{i-1}$  at the beginning.
5. If  $i > 1$  decrease  $i$  by 1 and return to step 2. Otherwise  $dec_i$  is a chosen total decomposition.

In figure 6, we could remove the curve labels in the following order to produce a chosen total decomposition of the abstract description for  $d_2: U \rightarrow V \rightarrow S \rightarrow T \rightarrow R \rightarrow P \rightarrow Q$ ; here we obtain an inductively pierced abstract description on the removal of  $S$ . An alternative order is  $V \rightarrow T \rightarrow U \rightarrow S \rightarrow R \rightarrow Q \rightarrow P$ .

## 5.2 Transforming Decompositions

We would like to be able to visualize abstract description,  $D$ , using only circles (which are aesthetically pleasing) at the expense of duplicating curve labels. If  $D$  is an arbitrary abstract description this is, unfortunately, not necessarily possible. However, it is always possible to add zones to  $D$  and realize an abstract description that is drawable in this manner. Here, we show how to add sufficient zones to  $D$  to ensure drawability, given a chosen total decomposition,  $dec(D) = (D_0, \dots, D_n)$ .

We observe that, when removing  $\lambda_i$  from  $D_{i+1}$  to obtain  $D_i$ , the zone set  $Z(D_i)$  can be expressed as  $Z(D_i) = in_i \cup out_i$ , where

1.  $in_i = \{z \in Z(D_i) : z \cup \{\lambda_i\} \in Z(D_{i+1})\}$ , and
2.  $out_i = \{z \in Z(D_i) : z \in Z(D_{i+1})\}$ .

We say that the zone sets  $in_i$  and  $out_i$  are defined by  $D_i$  and  $D_{i+1}$ . If  $\lambda_i$  is a piercing curve label then  $in_i \subseteq out_i$ , since  $\lambda_i$  ‘splits’ all of the zones through which it passes (if a piece of a zone is inside  $\lambda_i$  then a piece is also outside  $\lambda_i$ ). consider a zone,  $z$ , that is in  $in_i$  but not in  $out_i$ . Then  $z$  is not split by  $\lambda_i$  and  $z \notin Z(D_{i+1})$ ; transforming  $D_{i+1}$  by adding  $z$  to  $Z(D_{i+1})$  will result in  $z$  being split by  $\lambda_i$  and being added to  $out_i$ . We transform  $dec(D)$  into a new sequence of abstract descriptions that ensure all zones passed through are split on the addition of  $\lambda_i$ . This transformation process is defined below.

The addition of these zones removes any need for concurrency in the drawings. For instance, suppose we wish to add a contour labelled  $U$  to  $d_4$  in figure 6, so that the zone  $\{P\}$  is contained by  $U$  and all other zones are outside  $U$ . Then

the new curve would need to run along the boundary of the zone  $\{P\}$  and, therefore, be (partially) concurrent with the curves  $P$ ,  $R$ , and  $T$ . Altering this curve addition so that the zone  $\{P\}$  is instead split by  $U$  allows us to draw  $U$  as a circle inside the zone  $\{P\}$ , and the ‘extra’ zone will be shaded.

**Definition 7.** Given a chosen, total decomposition,  $dec(D) = (D_0, \dots, D_n)$ , transform  $dec(D)$  into a **splitting super-decomposition**,  $dec(D') = (D'_0, \dots, D'_n)$ , associated with  $D$  as follows:

1.  $D_0$  remains unchanged, that is  $D_0 = D'_0$ .
2.  $D_{i+1} = (L_{i+1}, Z_{i+1})$  is replaced by  $D'_{i+1} = (L_{i+1}, Z'_{i+1})$  where

$$Z'_{i+1} = Z_{i+1} \cup \bigcup_{j \leq i} in_j$$

where  $in_j$  is as defined above, given  $D_j$  and  $D_{j+1}$ .

Given a splitting super-decomposition associated with  $D$ , we know that if  $D_i$  is inductively pierced then  $D'_i = D_i$ .

**Theorem 2.** A splitting super-decomposition,  $dec(D') = (D'_0, \dots, D'_n)$ , associated with  $D$  is a total decomposition of  $D'_n$ .

Our problem is now to find a drawing of  $D'_n$  rather than  $D_n$ . We note that  $D'_n$  has a superset of  $D_n$ 's zones and we will use shading, as is typical in the literature, to indicate that the extra zones are not required (semantically, the extra zones represent the empty set).

### 5.3 Contour Identification and the Drawing Process

Given a splitting super-decomposition,  $dec(D') = (D'_0, \dots, D'_n)$ , we are in a position to start drawing our diagram. First, we identify  $D'_i$  in  $dec(D')$  such that  $D'_i$  is inductively pierced but  $D'_{i+1}$  is not inductively pierced. We draw  $D'_i$ , using the methods of [19], yielding an inductively pierced drawing of  $D'_i$ . The manner in which we add the remaining curves using partitions (described below) should give the idea as to how  $D'_i$  is drawn; in the inductively pierced case, there is one ‘valid partition’ that includes all zones in  $in'_j$  which gives rise to one circle.

Suppose, without loss of generality, that we have obtained a drawing,  $d'_j$ , of  $D'_j$ , where  $j \geq i$ , that is inductively pierced up to curve relabelling (so it is drawn with circles). It is then sufficient to describe how to add a contour, labelled  $\lambda_j$ , to  $d'_j$  in order to obtain such a drawing,  $d'_{j+1}$ , of  $D'_{j+1}$ . This will justify that  $D'_n$  has a drawing that is inductively pierced up to curve relabelling.

Consider the sets  $in'_j$  and  $out'_j$  which describe, at the abstract level, how to add  $\lambda_j$  to  $d'_j$ : the zones in  $in'_j$  are to be split by curves labelled  $\lambda_j$  whereas those in  $out'_j$  are to be completely outside curves labelled  $\lambda_j$ . Trivially, we can draw one circle inside each zone of  $d'_j$  whose abstraction is in  $in'_j$  to obtain  $d'_{j+1}$ ; label each such circle  $\lambda_j$ . See figure 6, where the contour  $T$  has been drawn in this manner in  $d_3$  given the set  $in = \{P, PQ, QS\}$ .

**Theorem 3.** *Let  $\text{dec}(D) = (D_0, \dots, D_n)$  be a decomposition with splitting super-decomposition  $\text{dec}(D') = (D'_0, \dots, D'_n)$ . Then  $\text{dec}(D')$  has a drawing,  $d$ , that is inductively pierced up to curve relabelling.*

Of course, the justification of the above theorem (drawing one circle in each split zone) may very well give rise to contours consisting of more curves than is absolutely necessary, as in  $d_3$  of figure 6. We seek methods of choosing how to draw each contour using fewer curves. Consider the drawing,  $d'_j$ , of  $D'_j$ . We know that each zone in  $in'_j$  is to be split by the to-be-added contour. We partition  $in'_j$  into sets of zones, according to whether they are topologically adjacent or form a cluster in  $d'_j$ . The sets in the partition will each give rise to a circle labelled  $\lambda_j$  in  $d'_{j+1}$ . In  $d_3$  of figure 6, the zones  $A$  and  $AB$  form a cluster, so  $in = \{P, PQ, QS\}$  can be partitioned into two sets:  $\{\{P, PQ\}, \{QS\}\}$ . Using this partition, we draw  $d_4$  in figure 6 rather than  $d_3$ .

**Definition 8.** *A partition of  $in'_j$  is **valid** given  $d'_j$  if each set,  $S$ , ensures the following:*

1.  $S$  is a cluster that contains 1, 2 or 4 zones,
2. if  $|S| = 2$  then the zones in  $d'_j$  whose abstractions are in  $S$  are topologically adjacent given a curve whose label is in the symmetric difference of the zones in  $S$ , and
3. if  $|S| = 4$  then there exists a pair of curves,  $c_1$  and  $c_2$ , that intersect at some point  $p$  in  $d'_j$  such that the zones in  $d'_j$  whose abstractions are in  $S$  form a cluster given  $c_1$ ,  $c_2$  and  $p$ .

Each set,  $S$ , in a valid partition gives rise to a circle in  $d'_{j+1}$ :

1. if  $|S| = 1$  then draw a circle inside the zone whose abstraction is in  $S$ ,
2. if  $|S| = 2$  then draw a circle that intersects  $c$  (as described in 2 above), and no other curves, and that splits all and only the zones whose abstractions are in  $S$ , and
3. if  $|S| = 4$  then draw a circle around  $p$  (as described in 3 above) that intersects  $c_1$  and  $c_2$ , and no other curves, and that splits all and only the zones whose abstractions are in  $S$ .

There are often many valid partitions of  $in'_j$  and we may want to use heuristics to guide us towards a good choice. One heuristic is to minimize the number of sets in the partition, since each set will give rise to a circle in the drawn diagram.

#### 5.4 Illustrating the Drawing Method

We now demonstrate the drawing method via a worked example, starting with  $D = \{\emptyset, P, PQ, R, PR, QR, PQR, PS, PQS, PRS, PQRS, QS\}$ . Since there are four curve labels, as the first step in producing a chosen total decomposition, we define  $D = D_4$ . Next, we identify  $S$  as a minimal curve label and remove  $S$  to give  $D_3 = \{\emptyset, P, PQ, R, PR, QR, PQR, Q\}$ . Similarly, we identify  $R$ , then  $Q$ ,

then  $P$  as minimal, giving  $dec(D) = (D_0, D_1, D_2, D_3, D_4)$  as a chosen decomposition of  $D$ , where  $D_2 = \{\emptyset, P, PQ, Q\}$ ,  $D_1 = \{\emptyset, P\}$ , and  $D_0 = \{\emptyset\}$ . The table summarizes  $in_i$  and  $out_i$  at each step, and gives  $Z'_i$  (the zone sets of the abstract descriptions in the splitting super-decomposition):

$D_i$	$in_i$	$out_i$	$Z'_i$
$D_0$	$\{\emptyset\}$	$\{\emptyset\}$	$Z(D_0)$
$D_1$	$\{\emptyset, P\}$	$\{\emptyset, P\}$	$Z(D_1)$
$D_2$	$\{\emptyset, P, Q, PQ\}$	$\{\emptyset, P, PQ, Q\}$	$Z(D_2)$
$D_3$	$\{P, PQ, PR, PQR, Q\}$	$\{\emptyset, P, PQ, R, PR, QR, PQR\}$	$Z(D_3)$
$D_4$	—	—	$Z(D_4) \cup \{Q\}$

Thus, the splitting super-decomposition is  $dec(D') = (D'_0, D'_1, D'_2, D'_3, D'_4)$  where  $D_i = D'_i$  for  $i \leq 3$  and  $D'_4$  has zone set  $Z(D_4) \cup \{Q\}$ . We note that  $D'_3$  is an abstract description of Venn-3, the Venn diagram with three curves, and is drawn by our method as  $d'_3$  in figure 7. To  $d'_3$  we wish to add a contour labelled  $S$ ; note that  $in'_4 = \{P, PQ, PR, PQR, Q\}$  and  $out'_4 = \{\emptyset, P, PQ, R, PR, QR, PQR, Q\}$ . Given  $d'_3$ ,  $\{\{P, PQ, PQR, PQ\}, \{Q\}\}$  is a valid partition of  $in'_4$ . Using this partition, we obtain  $d'_4$  where the zone with abstraction  $\{Q\}$  is shaded, since  $\{Q\}$  is in  $D'_4$  but not in  $D_4$ .

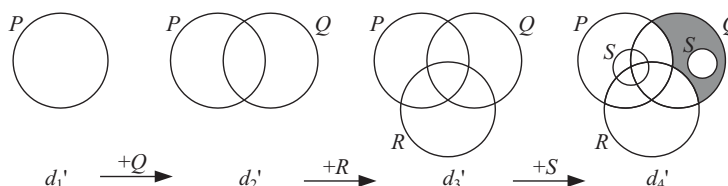


Fig. 7. Illustrating the drawing method.

## 6 Implementation and Comparison with other Methods

We have implemented our drawing method and the software is available for download; see [www.eulardiagrams.com](http://www.eulardiagrams.com). Examples drawn using our software are shown in figure 8. The lefthand diagram was drawn from abstraction  $\{\emptyset, ac, ab, b\}$ ; when entering the abstract description into the tool, the  $\emptyset$  zone is not entered and the commas are omitted. The other two diagrams were drawn from abstractions  $\{\emptyset, a, b, ab, ac, bd, ef\}$  and  $\{\emptyset, b, ab, c, ac, bc, abc, cd, bd, d, ae\}$  respectively, where the contour  $d$  comprises two curves in the latter case. In all cases, the shaded zones were not present in the abstract description. Layout improvements are certainly possible, particularly with respect to the location of the curve labels relative to the curves and the areas of the zones. We plan to investigate the use of force directed layout algorithms to improve the layout.

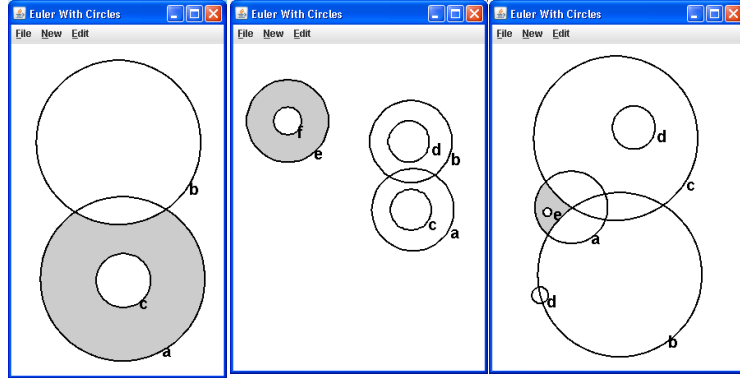


Fig. 8. Output from our software.

We now include some examples of output from other implemented drawing methods, permitting their aesthetic qualities to be contrasted with the diagrams drawn using our software. Figure 9 shows an illustration of the output using the software of Flower and Howse [8], which presents techniques to draw completely wellformed diagrams, but the associated software only supports drawing up to 4 curves. The techniques of Flower and Howse [8] were extended in [9] to enhance the layout; the result of the layout improvements applied to the lefthand diagram in figure 10 can be seen on the right.

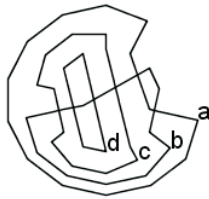


Fig. 9. Generation using [8].

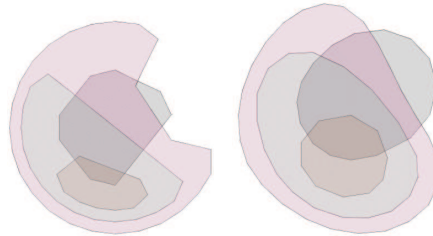
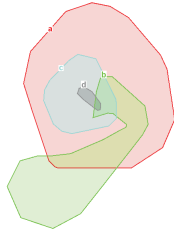
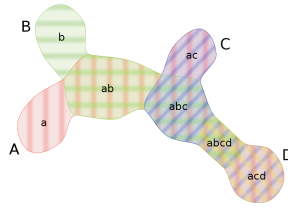


Fig. 10. Using the layout improvement [9].

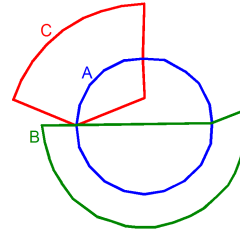
Further extensions to the methods of [8] allow the drawing of abstract descriptions that need not have a completely wellformed embedding. This was done in [13], where techniques to allow any abstract description to be drawn were developed; output from the software of [13] is in figure 11. An alternative method is developed by Simonetto and Auber [14], which is implemented in [15]. Output can be seen in figure 12, where the labels have been manually added post drawing; we thank Paolo Simonetto for this image. Most recently, an inductive generation method has been developed [18], which draws Euler diagrams by adding one curve at a time; see figure 13 for an example of the software output.



**Fig. 11.** Generation using [13].



**Fig. 12.** Generation using [15].



**Fig. 13.** Generation using [18].

A different method was developed by Chow [2], that relies on the intersection between all curves in the to-be-generated Euler diagram being present. We do not have access to Chow’s implementation, so we refer the reader to <http://apollo.cs.uvic.ca/euler/DrawEuler/index.html> for images of automatically drawn diagrams.

## 7 Conclusion

We have presented a technique that draws Euler diagrams that are completely wellformed up to labelling. The drawings use only circles as curves, which are aesthetically desirable; many manually drawn Euler diagrams employ circles which demonstrates their popularity. This is the first method that can draw any abstract description using circles. Of course, our drawings may include extra zones but we can mark them as such by shading them gray.

Along with layout improvements, as discussed in section 6, future work will involve giving more consideration as to how to choose valid partitions, since the choice of partition can impact the quality of the drawn diagram. Moreover, the zones we added to produce a splitting super-decomposition removed the need for concurrency in the diagram. We could add further zones that reduce the number of duplicate curve labels required. For instance, three zones,  $z_1$ ,  $z_2$  and  $z_3$ , in  $in_i$  may have a valid partition  $\{\{z_1, z_2\}, \{z_3\}\}$ , meaning we use two circles when adding  $\lambda_i$ . However, we might be able to add a fourth zone,  $z_4$ , to  $in_i$  where  $\{\{z_1, z_2, z_3, z_4\}\}$  is a valid partition (i.e.  $\{z_1, z_2, z_3, z_4\}$  forms a cluster) for which we are able to add a single 2-piercing curve. Finding a balance between the number of curves of which a contour consists and the number of ‘extra’ zones in order to obtain an effective diagram will be an interesting challenge.

**Acknowledgements** This research is supported by EPSRC grants EP/E011160/1 and EP/E010393/1 for the Visualization with Euler Diagrams project. We thank John Taylor for comments on an earlier draft.

## References

1. Google Charts API. <http://code.google.com/apis/chart/>, accessed August 2009.

2. S. Chow. *Generating and Drawing Area-Proportional Euler and Venn Diagrams*. PhD thesis, University of Victoria, 2007.
3. S. Chow and P. Rodgers. Constructing area-proportional Venn and Euler diagrams with three circles. In *Euler Diagrams 2005*.
4. R. DeChiara, U. Erra, and V. Scarano. VennFS: A Venn diagram file manager. In *Information Visualisation*, pages 120–126. IEEE, 2003.
5. A. Edwards. Venn diagrams for many sets. *New Scientist*, 7:51–56, 1989.
6. M. Farfel, L. DiGrande, R. Brackbill, A. Prann, J. Cone, R. Samuel R, D. Friedman, D. Walker, G. Pezeshki, P. Thomas, Sandro Galea, D. Williamson, T. Frieden, and L. Thorpe. An overview of 9/11 experiences and respiratory and mental health conditions among world trade center health registry enrollees. *Journal of Urban Health*, 85(6):880–909, 2008.
7. G. Farrell and W. Sousa. Repeat victimization and hot spots: The overlap and its implication for crime control and problem-oriented policing. *Crime Prevention Studies*, 12:221–240, 2001.
8. J. Flower and J. Howse. Generating Euler diagrams. In *2nd International Conference on the Theory and Application of Diagrams*, pages 61–75, 2002. Springer.
9. J. Flower, P. Rodgers, and P. Mutton. Layout metrics for Euler diagrams. In *Information Visualisation*, pages 272–280. IEEE, 2003.
10. E. Ip. Visualizing multiple regression. *Journal of Statistics Education*, 9(1), 2001.
11. H. Kestler, A. Muller, J. Kraus, M. Buchholz, T. Gress, H. Liu, D. Kane, B. Zeeberg, and J. Weinstein. Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9(67), 2008.
12. H. Kestler, A. Muller, H. Liu, D. Kane, B. Zeeberg, and J. Weinstein. Euler diagrams for visualizing annotated gene expression data. In *Euler Diagrams 2005*.
13. P. Rodgers, L. Zhang, and A. Fish. General Euler diagram generation. In *5th International Conference on the Theory and Application of Diagrams*, pages 13–27. Springer, 2008.
14. P. Simonetto and D. Auber. An heuristic for the construction of intersection graphs. In *Information Visualisation*, IEEE, 2009.
15. P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. *Computer Graphics Forum*, 28(3), 2009.
16. J. Soriano, K. Davis B. Coleman, G. Visick, D. Mannino, and N. Pride. The proportional Venn diagram of obstructive lung disease. *Chest*, 124:474–481, 2003.
17. G. Stapleton, P. Rodgers, J. Howse, and J. Taylor. Properties of Euler diagrams. In *Layout of Software Engineering Diagrams*, pages 2–16. EASST, 2007.
18. G. Stapleton, P. Rodgers, J. Howse, and L. Zhang. Inductively generating Euler diagrams. *accepted for IEEE Transactions on Visualization and Computer Graphics*, 2009.
19. G. Stapleton, L. Zhang, J. Howse, and P. Rodgers. Drawing Euler diagrams with circles: The theory of piercings. *accepted for IEEE Transactions on Visualisation and Computer Graphics*, 2010.
20. J. Thièvre, M. Viaud, and A. Verroust-Blondet. Using Euler diagrams in traditional library environments. In *Euler Diagrams 2004*, ENTCS 134, pages 189–202. 2005.
21. J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 1880.
22. A. Verrout and M.-L. Viaud. Ensuring the drawability of Euler diagrams for up to eight sets. In *3rd International Conference on the Theory and Application of Diagrams*, pages 128–141, 2004. Springer.
23. L. Wilkinson. VennEuler package for R, October 2009.