A General Method for Drawing Area-Proportional Euler Diagrams

Gem Stapleton[a], Peter Rodgers[b], John Howse[a]

[a] *Visual Modelling Group, University of Brighton, UK*
{g.e.stapleton,john.howse}@brighton.ac.uk
[b] *University of Kent, UK*
p.j.rodgers@kent.ac.uk

Abstract

Area-proportional Euler diagrams have many applications, for example they are often used for visualizing data in medical and biological domains. There have been a number of recent research efforts to automatically draw Euler diagrams when the areas of the regions are not considered, leading to a range of different drawing techniques. By contrast, substantially less progress has been made on the problem of automatically drawing area-proportional Euler diagrams, although some partial results have been derived. In this paper, we considerably advance the state-of-the-art in area-proportional Euler diagram drawing by presenting the first method that is capable of generating such a diagram given any area-proportional specification. Moreover, our drawing method is sufficiently flexible that it allows one to specify which of the typically enforced wellformedness conditions should be possessed by the to-be-drawn Euler diagram.

Key words: Area-Proportional, Euler diagrams, Information Visualization, Non-hierarchical data visualization, Venn diagrams

1. Introduction

There are many situations where data is more easily interpreted using visualizations. For instance, in statistical data analysis bar charts or pie charts are frequently used, and graphs can be used for network visualization. These diagrams 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. Euler diagrams are an effective means for visualizing this type of data.

For example, the authors of [9] have data concerning health registry enrollees at the world trade center. 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 [9] chose to use an Euler diagram which can be seen in figure 1. Here, the areas of the regions in the diagram are taken to be in proportion to the cardinalities of the represented sets. Since the areas of the regions convey cardinality information, the diagram is said to be area-proportional. Area-proportional Euler diagrams are also used for information visualization in areas including crime control [10], computer file organization [6], classification systems [26], education [16], genetics [18], and medicine [22].

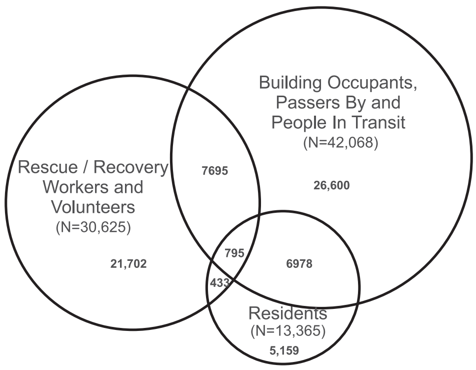

Figure 1: Major enrollment groups in the World Trade Center Health Registry.

As with other diagram types, the ability to automatically create area-proportional Euler diagrams from the data would be advantageous. Indeed, one could argue that their automated construction is necessary if we are to be able to widely apply the use of these diagrams in information visualization. To date, few methods for automatically drawing area-proportional Euler diagrams have been developed and those which exist are all limited to some extent; these methods will be discussed below. As a consequence of these severe limitations, area-proportional Euler diagrams typically need to be manually drawn, using only estimates of area.

In general, Euler diagrams [8] are collections of closed curves and are typically used to visualize relationships between sets. They are a generalization of Venn diagrams [27]; in a Venn diagram, all possible intersections between the sets must be represented, unlike an Euler diagram where all the intersections do not have to be represented. Euler diagrams exploit topological properties to convey information. For instance, a curve completely enclosed by another curve asserts a containment (subset) relationship. In figure 2, the diagram asserts that all executive members are also members, since the curve representing the former set is completely inside the curve representing the latter set. This diagram also asserts that the set's members and staff are disjoint since the respective two circles have no common points inside them. The regions formed by the curves represent the intersections of the corresponding sets. So, in figure 2, the region inside the curve Members but outside both the Exec_Members and Staff curves represents the set

$$Members \cap \overline{Exec\_Members} \cap \overline{Staff}.$$

In the area-proportional case, the relative areas of the regions are taken to be in proportion to the represented data, as illustrated in figure 1.

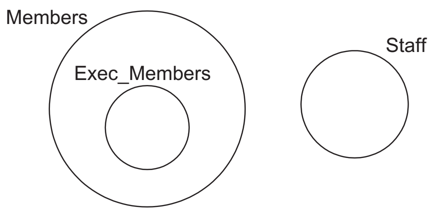

Figure 2: Exploiting topological properties.

This paper extends the existing state-of-the-art, presenting the first a method capable of drawing any area-proportional specification. Section 2 provides an overview of Euler diagram drawing methods and gives some examples of their use in information visualization. We give a sketch of our area-proportional Euler diagram drawing method in section 3. We provide definitions of Euler diagrams and related concepts in section 4. Section 5 defines area-proportional abstract descriptions, together with abstract-level concepts that correspond to diagram-level concepts given in section 4. Section 6 identifies how to decompose area-proportional abstract descriptions into a sequence of such descriptions in order to allow diagrams to be drawn inductively. Our drawing method uses graph-theoretic concepts, with section 7 defining the graphs that we require. In section 8, we describe how to use cycles in these graphs to add curves to a diagram. The theory is drawn together in section 9, where we present our inductive drawing method and discuss choices of area-proportional abstract description decomposition with respect to their impact on the final, drawn diagram. Finally, in section 10, we establish how properties of the cycles we use to add curves impacts the well-formedness properties possessed by the drawn diagram.

2. Related Work and Motivation

We cover two main themes: existing drawing methods for Euler diagrams (area-proportional or otherwise), and application areas where area-proportional Euler diagrams are helpful. Existing drawing methods start with an abstract description of the to-be-drawn diagram (possibly with an area specification) and proceed to seek a diagram; some of these methods guarantee the production of a diagram. Prior to this paper, there has only been limited success in the area-proportional case, with all methods often failing to draw any appropriate area-proportional Euler diagram.

2.1. Existing Drawing Methods

The vast majority of the research on automated Euler diagram drawing has focused on the non-area-proportional case. An abstract description of the to-be-generated diagram consists of a set of labels, which represent the sets to be visualized, and a set of 'zone descriptions', which represent the intersections of the sets that are to be visualized. For example, the diagram in figure 3 was automatically drawn using the methods of Rodgers et al. [20] and has zone descriptions $\emptyset$, $\{A\}$, $\{B\}$, $\{A,B\}$, $\{B,C\}$; for the sake of clarity, we will frequently omit $\emptyset$ from the list, and write the description as A, B, AB, BC. In essence, the abstract description describes the regions (called zones) formed by the curves. For example, AB describes the zone inside both of the curves labelled A and B, but not inside any other curves. We will simply call zone descriptions, such as AB, zones.

To specify an area-proportional Euler diagram, abstract descriptions are augmented with an area specification, assigning a required area to each zone. For example, abstract description A, B, AC, BC can be augmented with area specification $area(A) = 15$, $area(B) = 5$, $area(AC) = 7$ and $area(BC) = 10$.

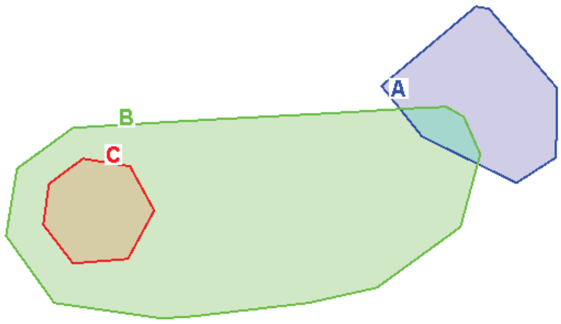

Figure 3: An automatically drawn Euler diagram.

2.1.1. Drawing Methods for Area-Proportional Euler Diagrams

There have been recent efforts towards solving the area-proportional Euler diagram drawing problem, that is: given an abstract description with an area specification, draw an area-proportional Euler diagram with that abstract description and the specified zone areas. Early work, by Chow and Ruskey [4], on the automatic drawing of area-proportional diagrams concentrated on rectilinear diagrams with three curves and diagrams drawn with two circles, further studied in [2]. Both methods are severely restricted, due to being able to represent only three or two sets respectively. Moreover, rectilinear layouts are not always effective from a usability perspective, having curves with 90 degree bends that can be difficult to follow, particularly at curve intersections. Later, their 'two circles' method was extended to the (still restrictive) three circle case by Chow and Rodgers [3]; in this case, however, only approximate areas can be drawn. Figure 1 shows three intersecting sets, taken from [9] where the authors state that it was drawn with the software developed by Chow and Rodgers.

In addition, the Google Chart API [1] includes facilities for drawing area-proportional Euler diagrams with at most three circles, using methods which approximate the areas. Figure 4 shows the output for an Euler diagram with abstract description A, B, AC, BC with areas $area(A) = 10$, $area(B) = 10$, $area(AC) = 10$ and $area(BC) = 10$; this area specification is impossible to draw using circles. Notice that (a) A and B overlap slightly, giving an extra zone (ABC), (b) there is also an extra zone, C, which consists of two minimal regions and (c) the area of A is larger than the area of AC, for instance. Recent work by Rodgers et al. also focuses on visualizing three

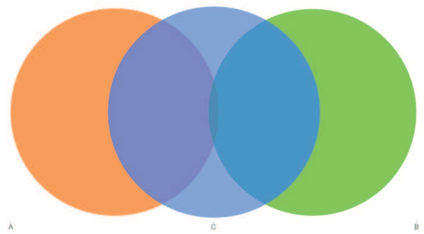

Figure 4: Google Charts Euler diagram.

sets, providing constructive (exact) drawing methods for symmetric Venn-3 diagrams using three convex curves [19].

Alternative work by Kestler et al. [17] developed drawing methods that produce diagrams with approximate areas, whose curves are restricted to being regular polygons. An implementation of their method was provided, called the VennMaster tool; figure 5 was automatically generated by VennMaster. Kestler et al.'s method does not guarantee that the drawn diagram has the correct abstract description or area specification. Typically the diagrams produced by VennMaster contain some zones that are represented by more than one minimal region[1], zones that are not required, zones which are required are not present, and the zone areas are inaccurate compared to the input data. These latter three (erroneous) features give rise to misleading judgements about the data being visualized. The problems here are partly because representing area specifications exactly with regular polygons is not always possible. Indeed, not all abstract descriptions (without area specifications) can be represented using regular polygons.

A drawing method by Chow draws *monotone* Euler diagrams, the definition of which implies that the intersection between all curves present [5]. Again, this means that most abstract descriptions and, therefore, most area specifications, cannot be drawn. However, when this method can draw the required diagram it does guarantee to produce the correct areas.

2.1.2. Drawing Methods for Non-Area-Proportional Euler Diagrams

Early automatic Euler diagram layout methods did not consider areas at all and simply concentrated on deriving diagrams with the correct zones. The first Euler diagram drawing algorithm, which was developed by Flower and

[1]Such zones are said to be *disconnected*.

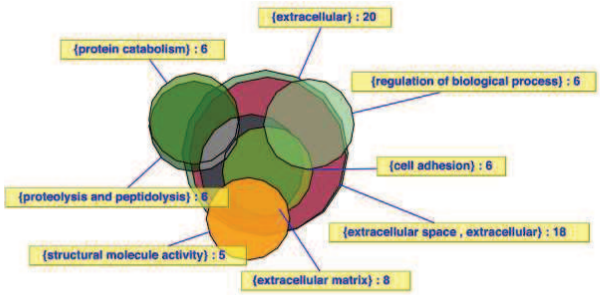

Figure 5: Euler diagram of results from a genetic microarray experiment.

Howse [13], guarantees the production of Euler diagrams restricted to those that were considered 'well-formed' which ensured that the diagrams do not have certain undesirable properties, including: no concurrent curves, no non-simple curves, and no points where more than two curves meet; various well-formedness properties will be described in more detail below (section 10). A software implementation of this first method was provided, but it was limited to diagrams containing at most four curves which could be drawn under the imposed well-formedness properties.

Extending this work, Chow [2] provided a drawing method that relaxed these well-formedness properties to a certain extent, although no software was provided. Both methods use a *dual graph approach* and Chow established that the Euler diagram drawing problem is NP-Complete in this case. Verroust and Viaud [28] chose an alternative relaxation of the well-formedness properties, which allowed curve labels to be used more than once. Under this relaxation, any abstract description representing at most eight sets can be drawn. As with Chow's method, no implementation of Verroust and Viaud's drawing method has been provided.

Recently, Rodgers et al. [20], have solved the general embedding problem by providing a method that, given any abstract description, draws an appropriate Euler diagram. Here, an implementation has been provided, available from `www.eulerdiagrams.com`, with figure 3 showing an example of the software's output. The diagrams drawn by the method are guaranteed to have connected zones and no non-simple curves, but may have concurrency, points where more than two curves meet, and multiple curve label use. Following a similar approach, Simonetto et al. [21] have devised a drawing method, with implementation, that produces Euler diagrams containing concurrency.

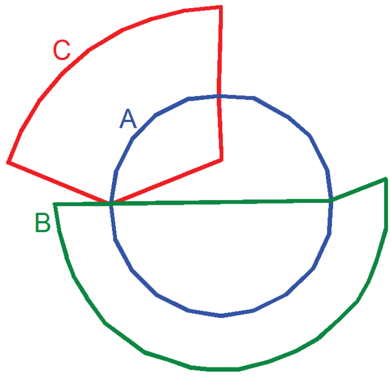

Figure 6: Inductive Euler diagram drawing.

All of the non-area-proportional Euler diagram drawing methods described so far use a dual graph approach. A general framework has been devised by Stapleton et al. [23] for this class of drawing methods. The framework allows one to identify the well-formedness properties that the Euler diagram, d, will possess, prior to drawing it, from the dual graph. Moreover, it was shown that the number of times d will violate the properties can also be counted from the dual graph.

An alternative drawing method was recently devised by Stapleton et al. [25] which takes an inductive approach, extending methods for Venn diagrams [7, 27]. Here, one curve is drawn at a time, identifying a route for the curve to take given the abstract description of the required diagram. It has advantages over the class of techniques described above in that it readily incorporates user preference for imposed well-formedness conditions (such as the curves must be simple), where this can be achieved. A partial implementation has been provided for this method, therefore giving some automated drawing support. Using this inductive method, an automatically drawn diagram with abstract description A, B, AB, C, AC can be seen in figure 6. It is this method that we extend to the area-proportional case.

2.2. Application Areas

There are numerous places in which area-proportional Euler diagrams can be used. For example, figure 1, obtained from [9], shows an automatically drawn three set Euler diagram (which is also a Venn diagram) with the areas of the regions intended to be proportional to the values in them.

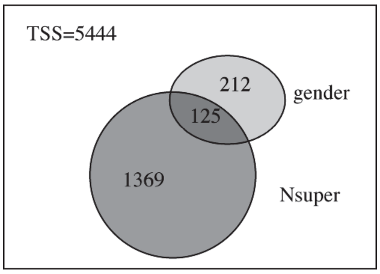

Figure 7: Statistics education.

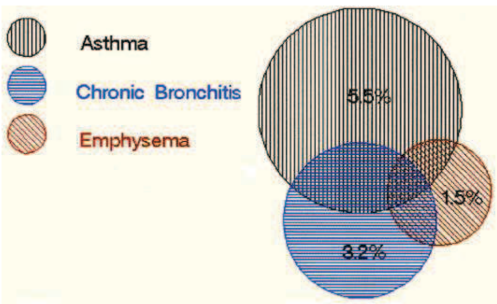

Figure 8: Medical example.

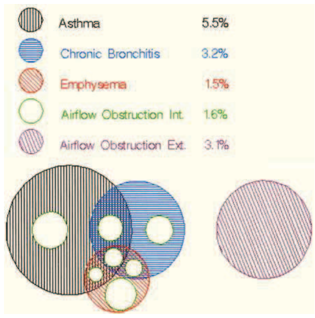

Figure 9: Extended medical example.

It was drawn using the software presented in [3]. This diagram illustrates some of the difficulties with current automated drawing methods: a fourth curve representing 'Students and School Staff (N=2646)' could not be drawn because the applied drawing method is limited to three circles.

Figure 5, from [17] and drawn using VennMaster, shows an automatically drawn diagram which is difficult to understand (and, as stated above, the drawing method used by VennMaster can yield diagrams with erroneous features). The diagram here represents data concerning genetic set relations.

An example from the education domain is in figure 7, obtained from [16]. The diagram illustrates predictive variables relating to annual salaries. This paper presented an empirical study that demonstrated Euler diagrams are an effective visualization of statistical information in some circumstances. The author of [16] acknowledges the current difficulty in automatically visualizing more than three sets using area-proportional Euler diagrams.

An example from the medical domain is shown in figure 8, obtained from [22]. It shows the intersections of physician-diagnosed asthma, chronic

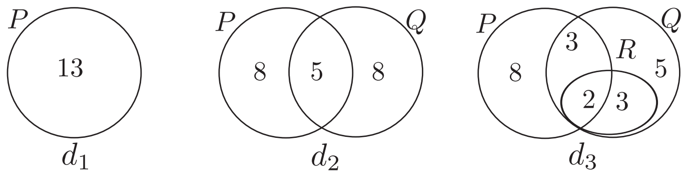

Figure 10: An inductive generation of an area-proportional Euler diagram.

bronchitis, and emphysema within patients with obstructive lung disease. An extension of this diagram, representing five sets and also obtained from [22], is shown in figure 9. The 'Airflow Obstruction Ext.' set is disjoint from the other four. Notice that the 'Airflow Obstruction Int.' set intersects with the first three sets but is represented using multiple curves.

3. An Overview of Our Area-Proportional Drawing Method

As previously indicated, our method for drawing a diagram, given an area-proportional abstract description, extends the inductive drawing method of [25]. To illustrate the technique, suppose we wish to draw the abstract description P, Q, PQ, QR, PQR with area specification:

$$\begin{aligned} area(P) &= 8 \\ area(Q) &= 5 \\ area(PQ) &= 3 \end{aligned} \qquad \begin{aligned} area(QR) &= 3 \\ area(PQR) &= 2. \end{aligned}$$

From this area specification, we can calculate the area inside curve P, for instance: the area is $8 + 3 + 2 = 13$. This sum is the total area of the zones (P, PQ, and PQR) that are inside curve P. We can then draw such a curve with the correct area, as shown in d_1 of figure 10. Next, we can add the curve Q, observing that the area inside this curve must be $5+3+3+2 = 13$ (from the four zones inside Q), with the overlap between curves P and Q having area $2 + 3 = 5$ (from the two zones inside both curves P and Q); the resulting diagram is d_2. Finally, we must add the curve R, which is to have a total area of $3 + 2 = 5$. The area inside both Q and R is specified to be 3, with the remaining 2 units to be inside P, Q and R. The final diagram containing all three curves is d_3.

What we have not described here is the manner in which we identify routes for the to-be-added curves to follow at each stage. Later, we provide

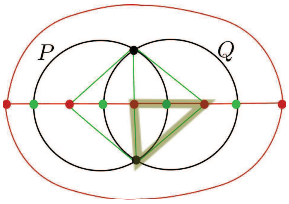

Figure 11: The hybrid graph.

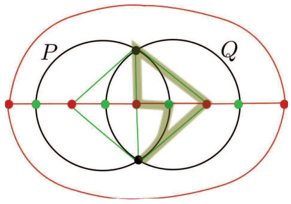

Figure 12: An alternative cycle.

a general method for routing the curves that can be fully automated. The technique uses graph theoretic concepts to identify routes for the curves. In particular, we construct a so-called hybrid graph (defined later), in which we identify cycles that form the basis of routes for the curves. We then modify the route so that we obtain the required areas.

To illustrate, the hybrid graph of d_2 given in figure 10 can be seen in figure 11. A cycle that identifies a region through which curve R may be routed is highlighted; it was this cycle that gave rise to d_3. Alternative cycles could have been chosen which would impact on the well-formedness properties possessed by the drawn diagram. For instance, a different choice of cycle is shown in figure 12 which would have given rise to the curve R being partially concurrent with the curve Q.

There are often many cycles that can be chosen as permissable curve routes and properties of the chosen cycles directly correspond to the well-formedness conditions that the drawn diagram will possess. Thus, placing restrictions on the permissable cycles corresponds to enforcing well-formedness properties. The remainder of the paper sets up the necessary framework for drawing area-proportional Euler diagrams, details the drawing method, and discusses ways in which we can add curves so that the resulting diagram respects specified well-formedness conditions.

The method we present builds on a substantial body of existing work, so here we outline the most significant novel aspects; note that the comments below make use of terminology that will be defined later in the paper. First, we provide generalizations of theoretical results on nested diagrams, given in [14], to the area-proportional case. The theory on decompositions in section 6 extends previous work to area-proportional abstract descriptions. The method we describe for adding contours in section 8 generalizes the method given in [25] in two significant ways: (a) the method describes how to add a contour consisting of multiple curves, whereas previously this was restricted to single-curve contours; (b) the curve routing method must ensure the correct areas are realized, as well as the correct zone set. Section 9 presents the inductive drawing method which contains additional steps to allow nested diagrams to be drawn, overcoming difficulties (which we discuss below) that arise due to area-proportionality. Moreover, in previous work, the order in which the contours were drawn was not given consideration. Section 9 discusses the impact of the contour ordering on the quality of the drawn diagram. This section further generalizes results about how the choice of cycles to add curves impacts on the well-formedness properties possessed by the drawn diagram; these generalizations are necessary since, unlike this paper, [25] did not allow contours to consist of multiple curves.

4. Euler Diagrams

An Euler diagram is a set of closed curves drawn in the plane[2]. We assume that each curve has a label chosen from some fixed set of labels, $\mathcal{L}$. The definitions given here are consistent with, or generalizations of, those found in the literature, such as in [2, 13, 24, 28].

Definition 4.1. *An **Euler diagram** is a pair, $d = (Curve, l)$, where*

1. *$Curve$ is a finite collection of closed curves each with codomain $\mathbb{R}^2$, and*
2. *$l\colon Curve \to \mathcal{L}$ is a function is that returns the label of each curve.*

Definition 4.2. *A **minimal region** of an Euler diagram $d = (Curve, l)$ is a connected component of*

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

It is important to be able to identify the interior of closed curves, since it is by containment and overlap that Euler diagrams convey information. A point, $p \in \mathbb{R}^2 - image(c)$, is *interior* to a closed curve, c, if and only if the winding number of c around p is odd; see [24] for more details.

[2]Recall, a closed curve in the plane is a continuous function, $c\colon [a, b] \to \mathbb{R}^2$, where $c(a) = c(b)$. Moreover, $image(c)$ is taken to denote the set of points to which c maps elements of the domain. The *image* notation is used more generally, to denote the set of elements to which some function maps elements of the domain.

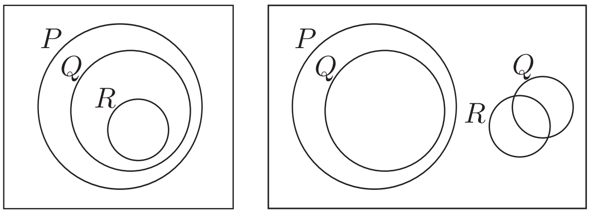

Figure 13: Two nested Euler diagrams.

Curve labels can occur more than once in an Euler diagram and curves with the same label are taken to denote the same set of objects. Thus, we need to identify all curves with the same label in order to have full access to the information provided about that set. We define the set of curves in a diagram with some specified label, λ, to be a **contour** with label λ. A point, p, is inside a contour precisely when the number of the contour's curves that p is is inside is odd. Another important concept is that of *zones*:

Definition 4.3. *A **zone** in an Euler diagram* $d = (Curve, l)$ *is a non-empty set of minimal regions that can be described as being interior to certain contours (possibly no contours) and exterior to the remaining contours. Given a zone,* z*, we denote the area of* z *by* $area(z)$*.*

Thus a zone represents a set intersection. For instance, in figure 10, d_2 contains four zones: three zones are inside the curves (described by P, Q, PQ) and one zone is outside all of the curves (described by $\emptyset$). The zone inside P only represents the set $P \cap \overline{Q}$ whereas the zone PQ represents $P \cap Q$.

Definition 4.4. *An Euler diagram,* d*, is **atomic** if the (images of) its curves form a connected component of the plane, otherwise* d *is **nested** [14]. The connected components formed by (the images of) the curves are called the **atomic components** of* d*.*

Both diagrams in figure 13 are nested and each has three atomic components.

This previous nesting work only considered diagrams where curve labels could not be used more than once. For our purposes, we extend the notion of nested to *contour-nested*, reflecting the potential multiple use of curve labels:

Definition 4.5. *A nested Euler diagram,* d*, is **contour-nested** if each contour appears in exactly one atomic component of* d*.*

The concept of contour-nestedness corresponds to being able to split the diagram up into components where each set is represented in exactly one component. Clearly, when curve labels are not duplicated, the concept of being contour-nested exactly coincides with that of being nested. In figure 13, the lefthand diagram is contour-nested, whereas the righthand diagram is not (the contour Q has curves in two atomic components).

5. Diagram Descriptions and Area Specifications

In order to generate an Euler diagram, d, we start with an abstract description of d together with an area specification, as illustrated informally above (section 3). Note that there is always a zone outside all of the contours, which includes the unbounded minimal region and its guaranteed presence is reflected in abstract descriptions. Since this region is unbounded, it has infinite area and, thus, its area is not specified when we assign areas to zones below. As with drawn diagrams, the labels used in the abstract descriptions are all chosen from $\mathcal{L}$ and the abstract zones are, therefore, chosen from $\mathbb{P}\mathcal{L}$.

Definition 5.1. *An* ***area-proportional abstract description****, D, is a triple, $(L, Z, area)$ 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 we define $Z(D) = Z$.*
3. *$area\colon Z(d) - \{\emptyset\} \to \mathbb{R}^+$ is a function that assigns desired areas to zones.*

We define $area_D = area$, enabling us to distinguish area functions when dealing with more than one abstraction. The ***area*** *of D, denoted $area(D)$, is the sum of the areas of its zones:*

$$area(D) = \sum_{z \in Z - \{\emptyset\}} area(z).$$

We have clearly overloaded the *area* notation, since we have also used it to denote a function that returned the area of a zone in a drawn diagram.

As an example of an area-proportional abstract description, D, the diagram, d, in figure 14 has label set $\{P, Q, R\}$, zone set $\{\emptyset, \{P\}, \{P, Q\}, \{P, Q, R\}\}$ and the areas are $area(P) = 2$, $area(PQ) = 2$, and $area(PQR) = 1$. We say that d is a drawing of D.

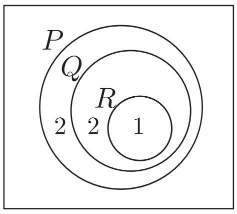

Figure 14: An area-proportional nested Euler diagram.

Definition 5.2. *Given an Euler diagram $d = (Curve, l)$, we map d to an area-proportional abstract description,*

$$abstract(d) = (image(l), Z, area),$$

called the ***abstraction*** *of d where*

1. *Z contains precisely one abstract zone for each zone in d; in particular, given a zone, z, in d, the set Z contains the abstract zone*

$$abstract(z) = \{l(c) : c \in C(z)\}$$

   where $C(z)$ is the set of contours in d that contain z and $l(c)$ returns the label of c, and

2. *for each zone, z, in d, that is inside at least one contour,*

$$area(z) = area(abstract(z)).$$

If D is an abstraction of d then we say d is a ***drawing*** *of D.*

The Euler diagram properties of atomicness and contour-nestedness can be detected at the abstract level. That is, if an area-proportional abstract description has a contour-nested drawing then it is always possible to identify this from an abstract description[3]. We extend the corresponding definitions of atomic and (contour-)nested given in [14] to the area-proportional case. However, we do not only need to identify whether a description is contour-nested, we also need to identify its parts that correspond to atomic components in drawings of it. Thus, we first define a new (non-commutative)

[3]Note that [14] defines nestedness at the abstract level, but considers a restricted class of Euler diagrams where contours comprise single curves. Our concept of contour-nested is equivalent to nested in this case.

operation, sum, on descriptions that simplifies both the presentation of the definition of contour-nested at the abstract level and facilitates the definition of the atomic components of descriptions. The sum operation allows us to 'slot' one abstract description inside a specified zone, $\hat{z}$, of another.

Definition 5.3. *Let $D_1 = (L_1, Z_1, area_1)$ and $D_2 = (L_2, Z_2, area_2)$ be area-proportional abstract descriptions. Let $\hat{z}$ be a zone in Z_1. If $L_1 \cap L_2 = \emptyset$ and either $area(D_2) < area_1(\hat{z})$ or $\hat{z} = \emptyset$ then we define the* ***sum*** *of D_1 and D_2 given $\hat{z}$, denoted $D_1 +_{\hat{z}} D_2$, to be an area-proportional abstract description, $(L, Z, area)$, where*

1. $L = L_1 \cup L_2$,
2. $Z = Z_1 \cup \{z_2 \cup \hat{z} : z_2 \in Z_2\}$,
3. *for each* $z \in Z - \{\emptyset\}$,

$$area(z) = \begin{cases} area_1(z), & \text{if } z \in Z_1 - \{\hat{z}\} \\ area_1(z) - area(D_2) & \text{if } z = \hat{z} \\ area_2(z - \hat{z}) & \text{if } z \in \{z_2 \cup \hat{z} : z_2 \in Z_2\} - \{\hat{z}\}. \end{cases}$$

For example, the area-proportional abstract description, D, of d in figure 14, can be written as $D_1 +_{PQ} D_2$ where D_1 comprises

1. labels $\{P, Q\}$,
2. zones $\{\emptyset, \{P\}, \{P, Q\}\}$, and
3. the area function is defined by $area(\{P\}) = 2$, $area(\{P, Q\}) = 3$,

and D_2 comprises

1. labels $\{R\}$,
2. zones $\{\emptyset, \{R\}\}$, and
3. the area function is defined by $area(\{R\}) = 1$.

By splitting D up into D_1 and D_2 in this manner, we can see that D is an abstraction of a contour-nested diagram.

Definition 5.4. *Let $D = (L, Z, area)$ be an area-proportional abstract description. Then D is* ***contour-nested*** *if there exist descriptions, $D_1 = (L_1, Z_1, area_1)$ and $D_2 = (L_2, Z_2, area_2)$, and a zone $\hat{z} \in Z_1$ such that*

1. $L_1 \neq \emptyset$ *and* $L_2 \neq \emptyset$,
2. $L_1 \cap L_2 = \emptyset$,

3. $area(D_2) < area(\hat{z})$ *or* $\hat{z} = \emptyset$, *and*
4. $D = D_1 +_{\hat{z}} D_2$.

If D is not contour-nested then D is ***atomic****, adapted and extended from [14].*

The sum of any two descriptions, each containing at least one curve label, is contour-nested. The next theorem tells us that the contour-nestedness at the abstract and drawn diagram levels coincide to some extent; our drawing method will demonstrate that they coincide as intended and, therefore, acts as a proof of theorem 5.1. An alternative proof of theorem 5.1 readily generalizes comparable results in [14] for the non-area-proportional case.

Theorem 5.1. *For each contour-nested area-proportional description, D there exists a contour-nested Euler diagram which is a drawing of D.*

By splitting up a nested area-proportional abstract description, D, into the sum of two parts, D_1 and D_2, we can identify whether it is contour nested. However, D_1 and D_2 need not be atomic parts. More importantly, there are atomic parts of D that cannot be realized as such a D_1 or D_2. For instance, the diagram, d, in figure 14 can be written as D_1 and D_2, given $\{P, Q\}$ as described above; here D_2 is atomic (the single curve R in the drawn diagram) whereas D_1 is not. It can also be written as $D_3 +_{\{P\}} D_4$ where D_3 has label set $\{P\}$ and D_4 has label set $\{Q, R\}$ (the rest of these abstract descriptions are not important for the purpose if this example). Here, D_3 is atomic (the single curve P) whereas D_4 is not. Now, d comprises three atomic parts, but it is not possible for the part consisting of just the curve Q to be in a sum that gives rise to D. We need access to all atomic parts for our drawing method. However, we can still use the sum operation to access this third part, noting that $D = D_3 +_{\{P\}} (D_5 +_{\{Q\}} D_2)$ where D_5 comprises

1. labels $\{Q\}$,
2. zones $\{\emptyset, \{Q\}\}$, and
3. the area function is defined by $area(\{Q\}) = 3$.

Definition 5.5. *Let $D = (L, Z, area)$ be an area-proportional abstract description. The* ***atomic components*** *of D are defined as follows:*

1. *If D is atomic then D is an atomic component of D.*
2. *If D is not atomic and $D = D_1 +_{\hat{z}} D_2$ for some descriptions D_1 and D_2 and zone $\hat{z}$ then the atomic components of D_1 together with those of D_2 are atomic components of D.*

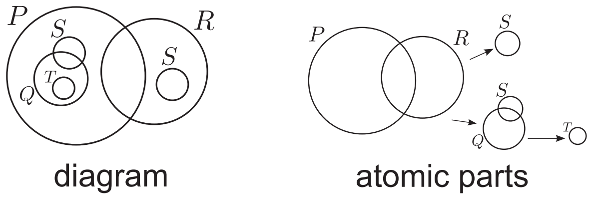

Figure 15: Containment of components.

The first stage in our drawing process takes D and identifies its atomic components. In the non-area-proportional case, the atomic components can be drawn independently, then scaled an merged to produce the final diagram. The motivation behind breaking the problem down in this manner is to due to its computational complexity; drawing the nested parts separately results in the diagram being drawn more quickly. In the area-proportional case, the atomic components cannot be drawn independently: scaling a component so that it can be drawn inside a zone changes the area proportions. However, we would still like to take advantage of the efficiency benefits brought about by the theory of nesting. To effect this, we must specify an order in which to draw the atomic components, so that we no longer draw them independently, but draw them inside the zones in which they are to be placed.

In figure 15, the diagram has 4 atomic components. The representation of these components on the right depicts a partial ordering. Our drawing method will start by drawing the leftmost component. We can then draw one of the next two components (that containing just the curve S or that containing the curves S and Q) inside the appropriate zones. Any component is drawn after any predecessor. We can compute an order in which to draw the atomic components in a manner that reflects their containment: we draw D_1 before D_2 if D_1 'contains' D_2. When drawing D_2 we are, thus, able to draw it in the zone that contains it and avoid scaling. In practice, the 'shape' of the zone will impact the choices of curve routings available. The routing techniques we provide handle being confined to remaining within a particular shape and are, therefore, immediately applicable to the contour-nested case.

6. Decomposing Abstract Descriptions

The area-proportional Euler diagram generation problem can be summarized as 'given an area-proportional abstract description, D, find an Euler diagram, d, that is a drawing of D. Our inductive approach will add curves successively until the generated Euler diagram has the required abstraction and zone areas. The routing of the curves will reflect the required areas, directly extending the methods of [25]. We need to know:

1. how to decompose D into a sequence of atomic descriptions
2. how to decompose D into a sequence $(D_0, D_1, ..., D_n)$ where D_0 contains no curve labels, D_{i-1} is obtained from D_i by removing a label, and $D_n = D$,

reflecting the next two subsections. We could restrict the second case, which identifies an order in which to draw the contours, to considering only atomic diagrams (because of case 1). However, we set up this decomposition processes more generally, since it can equally well be applied to non-atomic descriptions. Indeed, case 1 can easily be omitted from our drawing method, but the drawings we produce will not, then, reflect any nestedness present.

6.1. Component Decomposition

We want to produce a sequence of atomic descriptions that allows us to draw D by drawing one atomic component at a time.

Definition 6.1. *Given an area-proportional abstract description $D = (L, Z, area)$ a* ***component decomposition*** *is a sequence of area-proportional abstract descriptions, $decC(D) = (D_0, ..., D_n)$, which generates a sequence of atomic area-proportional abstract descriptions $decA(D) = (D'_0, ..., D'_{n-1})$ where*

1. *D_0 contains no curve labels,*
2. *$D_n = D$, and*
3. *for each i, where $0 \leq i < n$, $D_i = D_{i-1} +_{z_i} D'_{i-1}$ for some zone z_i.*

So, given such a $decC(D) = (D_0, ..., D_n)$ and $decA(D) = (D'_0, ..., D'_{n-1})$, our drawing method, after drawing D_i, will draw D'_i inside the zone z'_i of D_i. To produce a component decomposition, one simply identifies an atomic component of D, removes it, identifies an atomic component of the result, an so forth, until there are no non-trivial components remaining.

6.2. Contour Decomposition

Given an atomic description, D, our generation problem will find an embedding of D_1 by adding a curve to D_0 (which contains no curves), then D_2 (which contains 2 curves) and so forth, ending up with an embedding of $D_n = D$. As noted above, this process works for non-atomic descriptions, too, so the framework is set up for the general case.

Definition 6.2. *Given an area-proportional abstract description, $D = (L, Z, area)$, and $\lambda \in L$, we define $D - \lambda$ to be*

$$D - \lambda = (L - \{\lambda\}, \{z - \{\lambda\} : z \in Z\}, area')$$

where for each $z \in Z(D - \lambda) - \{\emptyset\}$

$$area'(z) = area(z) + area(z \cup \{\lambda\})$$

where we have extended the domain of area to include all zones, taking $area(z')$ to be 0 if z' is not in D.

Definition 6.3. *Given an area-proportional abstract description, $D = (L, Z, area)$, a* ***decomposition*** *of D is a sequence, $dec(D) = (D_0, D_1, ..., D_n)$ where each D_{i-1} $(0 < i \leq n)$ is obtained from D_i by the removal of some label, λ_i, from D_i (so, $D_{i-1} = D_i - \lambda_i$) and $D_n = D$. If D_0 contains no labels then $dec(D)$ is a* ***total decomposition****.*

The notion of a decomposition is similar to an alternative abstraction of Euler diagrams in [11], although there the authors do not consider areas.

Given a decomposition, $dec(D) = (D_0, D_1, ..., D_n)$, we need to be able to describe how to add a new curve label, λ_i, to D_i to obtain D_{i+1} (so, $D_{i+1} - \lambda_i = D_i$). In particular, we need to be able to identify the zones to be completely contained by λ_i, those to be completely outside λ_i, and those to be 'split' (i.e. partly inside and partly outside) by λ_i. We can deduce this information from the decomposition: considering D_i and D_{i+1}, we need to identify two sets of zones, that we call *in* and *out*, where

1. $in - out$ contains abstract zones whose drawn counterpart is to be completely inside the contour with label λ_i,
2. $out - in$ contains abstract zones whose drawn counterpart is to be completely outside the contour with label λ_i, and
3. $in \cap out$ contains abstract zones whose drawn counterpart is to be split by the contour with label λ_i.

It can easily be shown that

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

For notational convenience, we define $in(D_i, D_{i+1}) = in$ and $out(D_i, D_{i+1}) = out$.

We further need to describe how to subdivide the areas of the split zones when describing how to add λ_i to D_i. We define, for each zone, z, in $in(D_i, D_{i+1}) \cap out(D_i, D_{i+1})$, a pair of areas reflecting how z is to be subdivided at the drawn level:

$$splitArea(z) = (area_{D_{i+1}}(z), area_{D_{i+1}}(z \cup \{\lambda_i\}))$$

with $splitArea(D_i, D_{i+1})$ denoting the set of all such pairs. There is one exception to the above: if the zone $\emptyset$ is in in (it is necessarily in out since it is present in every abstract description) then the zone $\emptyset \cup \{\lambda_i\}$ can take any non-zero area. Finally, the addition of λ_i to D_i in order to obtain D_{i+1} is specified by

$$D_{i+1} = D_i + (\lambda_i, in(D_i, D_{i+1}), in(D_i, D_{i+1}), splitArea(D_i, D_{i+1})).$$

Of course, it would be a trivial matter to set up this framework for curve label addition in a more general manner, allowing the specification of the addition of a label without reference to a target area-proportional abstract description. However, our chosen presentation of the definitions reflects the primary aim of the paper: to provide a method that will facilitate the automated drawing of an area-proportional Euler diagram given any area-proportional abstract description.

7. Graphs for Contour Addition

Euler diagrams are associated with various graphs, some of which play an instrumental role in their automated layout; see [2, 13, 25] for more details. In this paper, we are interested in four of these associated graphs: the Euler graph, the Euler graph dual, the modified Euler dual, and the hybrid graph. Since these graphs were previously defined, we refer the reader to [25] for illustrative examples and associated discussion.

We can take an Euler diagram and construct its *Euler graph* which, roughly speaking, has a vertex at each point where two curves meet and the edges are the curve segments that connect the vertices. As a special case, any simple curve that does not intersect with any other curve has exactly one vertex placed on it. The Euler graph was originally defined in [2], but the definition relies on certain wellformedness conditions holding. We require a definition that applies in the general case:

Definition 7.1. *An **Euler graph** of Euler diagram $d = (Curve, l)$ is a plane graph, denoted $EG(d)$, whose embedded edges and vertices have image $\bigcup_{c \in Curve} image(c)$ and $EG(d)$ has a minimal number of vertices out of all the graphs to which it is homeomorphic (i.e. $EG(d)$ has no unnecessary vertices of degree two).*

Given a vertex, v, in an Euler graph dual, we write $z(v)$ to mean the zone of d in which v is embedded. We will talk about the images of the edges and vertices of these embedded graphs as simply the edges and vertices respectively. Later, we will define further graphs, which are also embedded in $\mathbb{R}^2$, associated with Euler diagrams and again blur the distinction between the edges (vertices) and the embedding of those edges (vertices).

Our generation method will add curves, respecting area constraints, by finding appropriate cycles in graphs. We could choose to use the dual graph to achieve this, but there would then be curves that we wish to add that could not necessarily follow a cycle in the dual. The issue arises, in atomic diagrams, because a dual graph does not allow us to route a curve in an arbitrary direction through the zone outside all of the curves. Following [25], we use the *modified Euler dual* as a basis for the graph that we will use to add contours. However, this graph will be extended, giving a so-called hybrid graph (see figure 11), in order to allow contours to be added in more ways.

Definition 7.2. *Let $d = (Curve, l)$ be an atomic Euler diagram. A **modified Euler dual** of d, denoted $MED(d)$, is a plane graph obtained from the Euler graph dual of d by carrying out the following sequence of transformations:*

1. *for each edge, e, incident with the vertex, v, placed in the unbounded face, f, of $EG(d)$ insert a new vertex of degree two onto e placed in f; the new vertex splits e into two edges in the obvious manner,*
2. *delete v along with all its incident edges; if this leaves any isolated vertices then delete those also,*
3. *add edges, embedded in f, connecting the newly inserted vertices (which have degree 1 after deleting v) so that the newly inserted vertices together with these new edges form a simple plane cycle[4] that properly encloses the Euler graph [25].*

[4] A simple cycle is one which does not pass through any vertex more than once (except the start and end vertex).

Definition 7.3. *Let $d = (Curve, l)$ be an atomic Euler diagram. A **hybrid graph** for d, denoted $HG(d) = (V, E)$, is a plane graph obtained from $EG(d)$ and $MED(d)$ by carrying out the following sequence of transformations:*

1. *take the embeddings of $EG(d)$ and $MED(d)$ as one embedded graph, G_1, (i.e. union the vertex sets and union the edge sets),*
2. *for each edge, e, in G_1 that is in $MED(d)$ and completely embedded in the unbounded face, f, of $EG(d)$ insert a new vertex onto e; the new vertex splits e into two edges in the obvious manner and we call the created graph G_2,*
3. *for each pair of edges, e_1 and e_2, in G_2, if e_1 and e_2 cross then insert a new vertex at the point where they cross; the new vertex splits each of e_1 and e_2 into two edges in the obvious manner, and we call the resulting graph G_3,*
4. *add edges to G_3 which are incident with a vertex in $MED(d)$ and a vertex in $EG(d)$ to create a graph, G_4, so that*
   (a) *all the new edges in G_4 are in the subgraph, SG_4, of G_4 generated by deleting the vertices of G_4 that are embedded in the unbounded face of $EG(d)$, and*
   (b) *SG_4 is triangulated except for its unbounded face,*
5. *add edges, e, to G_4, so that*
   (a) *e is incident with a vertex in $EG(d)$,*
   (b) *e is incident with a vertex in G_2 that is not in $MED(d)$ or in $EG(d)$, and*
   (c) *every vertex in G_2 that is not in $MED(d)$ or in $EG(d)$ is incident with exactly one new edge.*

   The resulting graph is $HG(d)$ [25].

Given a hybrid graph for d, we partition the set of edges as follows. Any edge in the hybrid graph that arose from the Euler graph is in the set $EulerEdges(HG(d))$. Any edge in the hybrid graph that arose from the modified Euler dual is in the set $DualEdges(HG(d))$. The remaining edges in the hybrid graph are in the set $NewEdges(HG(d))$. We call edges in the set $EulerEdges(HG(d))$ Euler edges and use similar terminology for elements of the other sets defined here. Moreover, the vertices are similarly partitioned into the sets $EulerVertices(HG(d))$, $DualVertices(HG(d))$, and $NewVertices(HG(d))$.

8. Adding Contours

Using a decomposition of an area-proportional abstract description, D, we can solve the drawing problem by identifying how to add a contour to a diagram in order to obtain some specified description. We demonstrate (a) how to identify a region of the diagram that can contain the required contour, and (b) that it is possible to draw the contour in a manner that guarantees the correct areas. To identify how we can draw the to-be-added contour, we use cycles in the hybrid graph. Recall, a **cycle**, C, in a graph $G = (V, E)$ is a non-empty sequence of edges, $C = (e_0, ..., e_n)$ in E, where no edge in E occurs more than once in C together with a sequence of vertices, $(v_0, ..., v_n, v_{n+1})$ such that $v_0 = v_{n+1}$ and each edge, e_i, in C is incident with v_i and v_{i+1}; such a sequence of vertices is **associated** with C. The set of edges in C is denoted $E(C)$ and the set of vertices in the vertex sequence associated with C is denoted $V(C)$.

Given a cycle in a hybrid graph, $HG(d)$, of some Euler diagram d, we can identify whether zones are inside, outside, or 'split' by the cycle. The concept of being inside a cycle will be defined by appealing to face-colouring. We observe that any cycle in a graph gives rise to an Eulerian subgraph (i.e. by deleting all edges not in the cycle and any isolated vertices). It was shown in [25] that any Eulerian, plane graph has a face-colouring that uses most two colours. Therefore, we can use such a colouring to identify the points inside (and outside) a cycle. Adding a curve to a diagram corresponds to traversing a cycle in the hybrid graph. In general, we want to describe how to add a contour to a diagram, d which can be done, in part, using a multi-set (sometimes called a bag) of cycles, $\mathcal{C}$, with each cycle giving rise to one of the contour's curves. Recall that a zone is inside a contour precisely when it is inside an odd number of its curves. Thus, the zones inside the new contour, the zones outside the new contour and the zones split by the contour can be immediately identified from the cycles and face colourings. We assume, without loss of generality, that given any pair of cycles in $\mathcal{C}$, each of their infinite faces has the same colour. This means that the minimal region, m, outside all curves in d, is coloured the same in each cycle. Taking d and such a two face colouring of each cycle in $\mathcal{C}$ we construct a marking of the points, p, of which the zones in d comprise:

1. p is marked $in(\mathcal{C})$ if the number of cycles in which p is coloured the same as m is odd, and

2. p is marked $out(\mathcal{C})$ if the number of cycles in which p is coloured the same as m is even.

We then define:

1. the abstract description of a zone z in d is defined to be **inside** $\mathcal{C}$ if any one of its points is in $in(\mathcal{C})$, the set of which is denoted $inZones(\mathcal{C})$, and
2. the abstract description of a zone z in d is defined to be **outside** $\mathcal{C}$ if any one of its points is in $in(\mathcal{C})$, the set of which is denoted $outZones(\mathcal{C})$.

The zones which are split are those whose abstract descriptions occur in both $inZones(\mathcal{C})$ and $outZones(\mathcal{C})$.

We want to be able to add a contour in order to obtain some specified abstraction augmented with an area specification, D. Consider, then, a decomposition, $dec(D) = (D_0, ..., D_n)$, of D where we have found an embedding of D_i, say d_i. We demonstrated above that we could compute appropriate sets, $in(D_i, D_{i+1})$ and $out(D_i, D_{i+1})$. Thus, to add an appropriate contour to d_i in order to obtain a drawing of D_{i+1}, we first seek a multi-set of cycles, $\mathcal{C}$, in $HG(d_i)$ such that

1. $in(D_i, D_{i+1}) = inZones(\mathcal{C})$, and
2. $out(D_i, D_{i+1}) = outZones(\mathcal{C})$.

To ensure the resulting diagram is atomic, each cycle must pass through an Euler vertex (a vertex in the Euler graph). It is relatively straightforward to justify the existence of an appropriate $\mathcal{C}$ given in and out. Intuitively:

1. For each abstract zone, z, in $in - out$, and for each minimal region, m, of d_i of which the drawn zone, $\hat{z}$, with abstraction z comprises, add a minimal set of cycles to $\mathcal{C}$ formed by traversing the edges of $HG(d_i)$ that bound m, such that each edge is included in exactly one cycle.
2. For each abstract zone, z, in $in \cap out$, and for each minimal region, m, of d_i of which the drawn zone, $\hat{z}$, with abstraction z comprises, add a simple cycle formed by traversing some of the edges of $HG(d_i)$ that bound m together with two new edges that are incident with the dual vertex inside m.

The contour added consists of curves that are formed by traversing each of the cycles in $\mathcal{C}$, ignoring the area constraints. This naive way to choose a multi-set of cycles can yield unattractive diagrams and, typically, much better choices can be made. For instance, one heuristic is to minimize the number of cycles in $\mathcal{C}$, but other heuristics may be more appropriate depending on the circumstances. Nonetheless, we can now state the following theorem:

Theorem 8.1. *Let $D_{i+1} = (L, Z, area)$ be an area-proportional abstract description and let $\lambda_i \in L$ and suppose $D_i = D_{i+1} - \lambda_i$ has drawing d_i. There exists a multi-set of cycles, $\mathcal{C}$ in $HG(d_i)$ such that*

1. $inZones(\mathcal{C}) = in(D_i, D_{i+1})$,
2. $outZones(\mathcal{C}) = out(D_i, D_{i+1})$, *and*
3. *each cycle in $\mathcal{C}$ passes through an Euler vertex.*

To summarize, we can now identify how to add a contour to a diagram in order to obtain the correct abstraction, ignoring area specifications.

Given $D_{i+1} = D_i + (\lambda_i, in, out, splitArea)$ we must consider how to obtain the correct areas. To demonstrate that this is always possible, suppose that $\mathcal{C}$ is a multi-set of cycles such that if we traverse each of the cycles to give rise to a curve then we obtain a diagram, d_{i+1}, with the correct zone set (although the areas may well be incorrect).

Consider a zone, z_{i+1}, in d_{i+1} with the wrong area. Then z_{i+1} must be a split zone, since the areas of the non-split zones will be unaltered and, by assumption, the areas are correct in d_i. The minimal regions of which z_{i+1} comprises will be either inside $\mathcal{C}$, outside $\mathcal{C}$ or passed through (i.e. split) by at least one cycle in $\mathcal{C}$. Figure 16 illustrates the three possibilities for six minimal regions of z_{i+1}, here depicted as circles; note that the figure assumes z_{i+1}'s minimal regions are simply connected, which is necessary in atomic diagrams unless z_{i+1} is outside all of the curves. The diagram depicts the part of the hybrid graph that includes edges in two of the cycles in $\mathcal{C}$, (partially) shown as C_1 and C_2. The shaded parts of the six minimal regions that form z_{i+1} are assumed to be inside $\mathcal{C}$.

Suppose that if our to-be-added contour traversed C_1 and C_2 then the area of z_{i+1} is too large; this is the shaded area. This implies that, in d_{i+1}, the area of the zone, z', with abstraction $abstract(z_{i+1}) - \{\lambda_i\}$ has too small an area, since the sum of the two areas must be $area_{d_i}(z)$, where z is the zone that was split to give z_{i+1} and z'_{i+1}, that is

$$area_{d_i}(z) = area_{d_{i+1}}(z_{i+1}) + area_{d_{i+1}}(z'_{i+1}).$$

Thus, we need to adjust the path taken by the new contour so that the areas are as required, reducing $area_{d_{i+1}}(z_{i+1})$ and increasing $area_{d_{i+1}}(z'_{i+1})$.

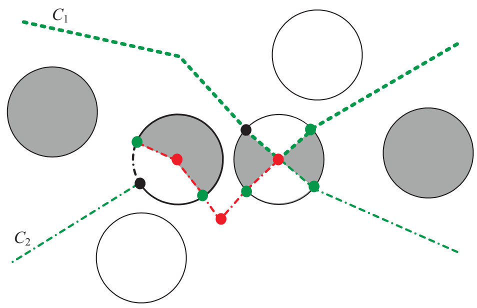

Figure 16: Adding a contour by traversing cycles

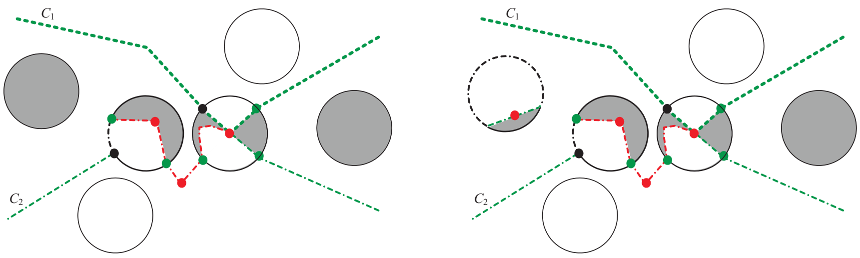

Figure 17: Re-routing the contour and adding new curves to achieve the correct areas.

This can be achieved by (a) re-routing the new or dual edges of the hybrid graph through the minimal regions that the cycle(s) passes through – these are the minimal regions of z that contain some vertex associated with at least one of the cycles in $\mathcal{C}$, or (b) adding further cycles to $\mathcal{C}$ that pass around or through the other minimal regions of which z comprises. In our example, suppose then that we cannot obtain the correct areas by process (a) alone, although we might still make use of process (a), as shown in the lefthand side of figure 17. To further decrease the area of z_{i+1}, we can add a further curve around one of its minimal regions and possibly, as shown in the figure, passing through that minimal region. Such curves can again be found by identifying particular cycles in the hybrid graph and adjusting the routing as in process (a). If the area of z_{i+1} is too small in d_i then the process for enlarging the areas is similar.

In this section, we have, therefore, described how to add a contour to obtain a drawing of D_{i+1} from a drawing of D_i, ensuring that the required areas are achieved.

9. Inductive Drawing Method

Given $D = (L, Z, area)$, follow the process below to draw D:

1. **Atomic Components** Identify the atomic components of D, by producing a component decomposition $decC(D) = (D_0, ..., D_n)$ and its associated sequence of atomic components, $decA(D) = (D'_0, ..., D'_{n-1})$
2. **Decompose Atomic Components** For each D'_i, find a total decomposition, say $dec(D'_i) = (D_{i,0}, D_{i,1}, ..., D_{i,m_i})$. Recall that $D_{i,0}$ has no curve labels and we denote the Euler diagram with abstraction $D_{i,0}$ by $d_{i,0}$. Here we can choose to use heuristics, when producing a decomposition, that may yield better drawn diagrams.
3. **Draw Atomic Parts** Draw each of the atomic components of D in order, starting with D'_1. Set $i = 1$.
   (a) *Draw a Contour for the Atomic Part, $D_{i,1}$, in $dec(D'_i)$* The description $D_{i,1}$ has a single curve label, $\lambda_{i,1}$. Draw a closed curve, that has label $\lambda_{i,1}$, with the correct area inside it. Ensure that this curve is drawn inside the zone, z, that identifies $D_{i,1}$ as an atomic component. This can trivially be achieved. Call this drawn diagram $d_{i,1}$. Set $j = 1$.
   (b) *Hybrid Graph* Construct the hybrid graph of $d_{i,j}$, ensuring that the routing of the edges remains inside the zone that contains $d_{i,j}$.
   (c) *Find Cycles* Identify a multi-set of cycles, $\mathcal{C}$, in $HG(d_{i,j})$ where
   $$inZones(\mathcal{C}) = in(D_{i,j}, D_{i,j+1}),$$
   $$outZones(\mathcal{C}) = out(D_{i,j}, D_{i,j+1}),$$
   and each cycle in $\mathcal{C}$ passes through an Euler vertex. Here, we can also choose to use heuristics, such as minimize the number of cycles in $\mathcal{C}$, when identifying an appropriate multi-set of cycles. Moreover, we can choose to place some constraints on the cycles in $\mathcal{C}$, as will be discussed below, but these constraints might render the description undrawable, if strictly enforced.
   (d) *Draw Contour* Using the cycles in $\mathcal{C}$ and the area adjustment method described above, add a contour to $d_{i,j}$ to give a drawing of $D_{i,j+1}$, say $d_{i,j+1}$.
   (e) *Iteration and Termination* Increment j by 1. If $j < m_i$ then return to the beginning of step 3(b). If $j = m_i$ and $i < n$ then increment i by 1 and return to the beginning of step 3(a). Otherwise, $i = n$ and we have drawn D.

Some of the steps in the above drawing method are not necessary. In particular, step 1 can be omitted, but this would result in contour-nested descriptions being drawn in an atomic manner. If step 1 is omitted, step 2 would find a decomposition of D, the original abstract description, with similar knock on effects to the rest of the drawing processes. In any case, to conclude this section we present our main theorem, the proof of which readily follows from the arguments presented throughout the paper:

Theorem 9.1. *Let $D = (L, Z, area)$ be an area-proportional abstract description. Applying the inductive drawing method produces a diagram d that is a drawing of D, provided no constraints are placed on the cycles in $\mathcal{C}$.*

9.1. Choosing a Decomposition

When obtaining a decomposition, there are choices about the order in which the curve labels are removed from an area-proportional abstract description, D. The order in which the labels are removed will impact the appearance of the final drawn diagram. As noted above, we can use heuristics when choosing a decomposition of an abstraction in order to achieve better drawings. Here we outline one such heuristic.

Our method for adding a contour will always result in an atomic diagram, but when we remove a label from area-proportional abstract description D we may create a contour-nested description, even if D is atomic. The removal of a contour from a drawn diagram that results in an increase in the number of nested components is *disconnecting*, the theory of which, when contours consist only of single curves, is developed by Fish and Flower in [12]. At the abstract level, it is trivial to identify whether a curve label is disconnecting:

Definition 9.1. *Let $D = (L, Z, area)$ be an area-proportional abstract description and let $\lambda \in L$. If $D - \lambda$ has more atomic components than D then λ is called a **disconnecting** curve label for D.*

We suggest prioritizing the removal of non-disconnecting curve labels when constructing a decomposition because we can then draw diagrams that better reflect nestedness, as we now explain. Suppose that D has a disconnecting curve label, λ, and, moreover, suppose that we have a drawing, $d-\lambda$, of $D - \lambda$ produced using our inductive method. Then $d - \lambda$ is atomic, since our drawing method only produces atomic drawings. Thus, when we add a curve, c, labelled λ to $d - \lambda$ to give a drawing, d, of D, c will not be a

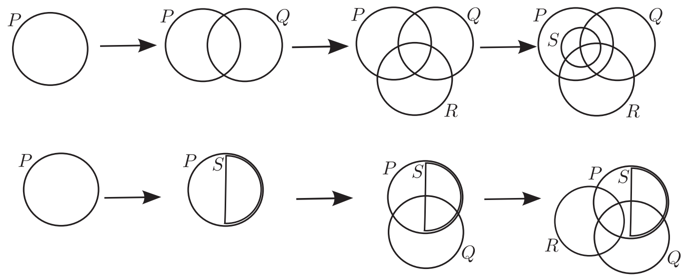

Figure 18: Prioritizing the removal of non-disconnecting curves.

disconnecting curve in d. However, choosing a different decomposition of D could have resulted in c being a disconnecting curve and, hence, the drawing would better reflect the 'lower level' nestedness present in the diagram.

To illustrate, consider the two diagram sequences in figure 18. The final diagrams in each sequence have the same abstract description. However, the order in which the curves were drawn was different. In the top sequence, the curves are added in the order $P \to Q \to R \to S$, meaning that the decomposition was created by removing them in the opposite order, $S \to R \to Q \to P$. At each stage, the curve removed was not disconnecting. For the bottom diagram, the curves are removed in the order $R \to Q \to S \to P$. The curve Q is disconnecting: the abstract diagram description of the second diagram in the bottom sequence is nested. Since the drawing method produces only atomic diagrams, this nestedness is not reflected in the drawing: P and S touch. By contrast, P properly contains S in the final diagram of the top sequence.

A claim in [12] is that if one removes a disconnecting curve, draws the resulting atomic parts, and then draws the disconnecting curve, c, then it might decrease the time taken to draw diagrams; as stated above, in [12] curve labels cannot be used more than once, so every contour is a curve. However, this motivation for developing the theory of disconnecting curves has yet to be justified. In particular, the choice of embeddings of the atomic parts will have to be given careful consideration (that is, the parts cannot be drawn without considering the fact that c has to be added) if one wishes to enforce particular well-formedness conditions or obtain the 'best' drawing of the diagram. Whether drawing efficiency is indeed improved will almost certainly depend on the drawing method employed and the abstract description in question; the impact on efficiency remains the subject of future work.

10. Choosing Cycles for Contour Addition

Sometimes we wish to add contours under certain constraints to achieve better drawings. The constraints we consider correspond to properties, often called wellformedness conditions, that the drawn diagrams posses.

Definition 10.1. *Given an Euler diagram $d = (Curve, l)$, the following are properties that d may posses.*

1. *If the labelling function, l, is injective then d possesses the **unique labels property**.*
2. *If all of the curves in $Curve$ are simple then d possesses the **curve simplicity property**.*
3. *If no pair of curves in $Curve$ run concurrently then d possesses the **no concurrency property**.*
4. *If there are no points in $\mathbb{R}^2$ that are passed through more than twice by the curves in $Curve$ then d possesses the **no triple points property**[5].*
5. *If whenever two curves in $Curve$ intersect, they cross then d possesses the **crossings property**.*
6. *If each zone in d is connected (i.e consists of exactly one minimal region) then d possesses the **connected zones property**.*

Formalizations of these properties can be found in [24]. To the best of our knowledge, no area-proportional Euler diagram drawing method allows the to-be-possessed properties to be specified in advance of attempting to find a drawing of an area-proportional abstract description: each method produces diagrams with some fixed set of properties or cannot guarantee that particular properties hold. Our drawing method can be used in such a manner as to ensure that any specified set of the above properties hold. However, a chosen collection of properties may result in a set of constraints for which there is no satisfying multi-set of cycles in the hybrid graph. Here, we present the constraints, with the formal details extending those in [25], except for the unique labels property which is entirely new (since that drawing method did not allow contours to consist of multiple curves).

We now consider each of the properties in turn, providing conditions on the cycles in $\mathcal{C}$ that correspond to the diagram that results from using $\mathcal{C}$ to add a contour possessing those properties. In the following subsections, we assume $\mathcal{C}$ contains cycles that permit the correct areas to be achieved.

[5]A triple point is one that is passed through (at least) three times by the curves.

10.1. Unique labels

Ensuring that a diagram possesses unique labels we simply ensure that we use exactly one cycle to add a contour. This is because having unique labels is directly equivalent to having only contours that consist of single curves.

Definition 10.2. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$. Then $\mathcal{C}$ possesses the **unique labels property** whenever $\mathcal{C}$ contains only one cycle, that is $|\mathcal{C}| = 1$.*

Theorem 10.1. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$ and let λ be a label that is not in d, $\lambda \notin image(l)$. Then the contour added to d using $\mathcal{C}$ consists of only one curve if and only if $\mathcal{C}$ possesses the unique labels property.*

Proof Trivially, if $\mathcal{C}$ possesses the unique labels property then a unique curve is obtained by traversing the edges in the single cycle in $\mathcal{C}$. For the converse, each cycle in $\mathcal{C}$ gives rise to a curve labelled λ, Since there is only one such curve, $\mathcal{C}$ possess the unique labels property. $\square$

If the required diagram is to have the unique labels property then every diagram we create in the inductive construction must have the unique labels property. A similar observation applies to all of the other properties except the connected zones property; we discuss this issue for that property in more detail below.

10.2. Curve Simplicity

The simplicity condition is very easy to enforce when adding a curve using a cycle: the cycle must not pass through any vertex more than once. Such a cycle is called *simple*.

Definition 10.3. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$. Then $\mathcal{C}$ possesses the **curve simplicity property** whenever each cycle, C, in $\mathcal{C}$ is simple.*

The contour added to d using $\mathcal{C}$ consists only of simple curves if and only if $\mathcal{C}$ possesses the curve simplicity property.

In addition, we could decide that we want all contours to be simple; a **simple contour** is a contour that consists of only simple curves and no pair of these curves intersect with each other. Extending the curve simplicity property for a multi-set of cycles to a contour simplicity property is straightforward: we add the constraint that no pair of cycles in $\mathcal{C}$ have a common vertex in their associated vertex sequences.

10.3. No Concurrency

The no concurrency condition requires the added contour does not run concurrently with any other contour or itself. To avoid concurrency with existing curves, we do not use any Euler edges in the cycles in $\mathcal{C}$. To avoid self-concurrency, we ensure that no pair of cycles in $\mathcal{C}$ traverse a common edge. From this it follows that $\mathcal{C}$ is a set, rather than a multi-set.

Definition 10.4. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$. Then $\mathcal{C}$ possesses the* ***no concurrency property*** *whenever*

1. *each cycle, C, in $\mathcal{C}$ does not contain any edges in $EulerEdges(HG(d))$, and*
2. *for each pair of cycles, C_1 and C_2, there is no edge that occurs in them both.*

The contour added to d using $\mathcal{C}$ does not run concurrently with any contour in d or itself if and only if $\mathcal{C}$ possess the no concurrency property.

10.4. No Triple Points

In order to enforce the no triple points condition, we must ensure that the added contour does not increase the multiplicity of any points. The **multiplicity** of a point, p, in a diagram, d, is the number of times to which p is mapped by the curves in d and if p has multiplicity 3 or greater then p is a **triple point**. We need access to the multiplicity of any points of intersection in order to identify whether the to-be-added contour creates a triple point. For each vertex, v, in $EulerVertices(HG(d)) \cup NewVertices(HG(d))$, we label that vertex by the multiplicity of that point in d, denoted $mul(v, d)$. We note that for any diagram, d, constructed using our inductive method which possesses the no concurrency property, any Euler vertex, v, has $mul(v, d) = \frac{deg(v)}{2}$ and for any new vertex, v, $mul(v, d) = 1$. For dual vertices, we set $mul(v, d) = 0$, since no curves in d pass through them.

When identifying triple point creation, we assume that the diagram to which the contour is to be added was drawn using the inductive method.

This is because, using our method, the multiplicity of a point on any Euler edge, e, in the hybrid graph is at most that of any vertex incident to e. Thus, we can identify the creation of a triple point by considering the vertices alone.

Definition 10.5. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$. Then $\mathcal{C}$ possesses the **no triple points property** whenever, for any vertex, v, in $HG(d)$, if it is the case that $mul(v, d)$ plus half the number of edges in cycles in $\mathcal{C}$ that are incident with v is at most two.*

The contour added to d using $\mathcal{C}$ does not introduce any triple points if and only if $\mathcal{C}$ has the no triple points property, provided d was drawn using the inductive method.

10.5. Crossings

There are various properties that our cycle must posses if it is to yield a curve that ensures the crossings property holds in the embedded diagram. First, we observe that any diagram that contains concurrency does not possess the crossings property. Second, suppose that the cycle contains an edge, e_i, that is incident with an Euler vertex, v_{i+1}, (e_i must be a new edge, since it cannot be an Euler edge or we would have concurrency). Then the next edge in the cycle (which must also be a new edge) must ensure that the cycle crosses all of the curves that give rise to Euler edges incident with v_{i+1}. The notion of a crossing arising from a cycle can be captured relatively straightforwardly: the cycle, when passing through an Euler vertex, v_{i+1}, must have exactly half of the Euler edges incident with v_{i+1} on one side of it, as illustrated in figure 19.

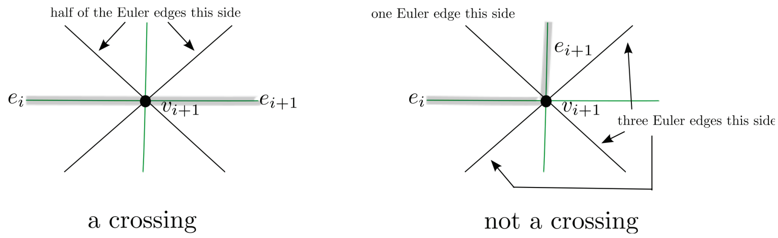

Figure 19: Detecting crossings.

A pair of consecutive edges, e_i and e_{i+1}, in a cycle, therefore, gives rise to a two way partition of the edges, excluding e_i and e_{i+1}, incident with the vertex v_{i+1} that joins e_i and e_{i+1}. We denote the two sets in this partition by $E_1(e_i, e_{i+1}, v_{i+1})$ and $E_2(e_i, e_{i+1}, v_{v+1})$. Thus, for crossings we require

$$|E_1(e_i, e_{i+1}, v_{i+1}) \cap EulerEdges(HG(d))| = |E_2(e_i, e_{i+1}, v_{i+1}) \cap EulerEdges(HG(d))|$$

for every pair of consecutive edges e_i and e_{i+1} in C that are incident with an Euler vertex v_{i+1}. We must also ensure that each cycle in $\mathcal{C}$ does not create a non-crossing point with itself and, moreover, that any pair of cycles do not create non-crossing points with each other.

Definition 10.6. *Let $d = (Curve, l)$ be an atomic Euler diagram with hybrid graph $HG(d)$. Let $\mathcal{C}$ be a multi-set of cycles in $HG(d)$. Then $\mathcal{C}$ possess the* ***crossings property*** *whenever the following all hold.*

1. *The multi-set $\mathcal{C}$ possesses the no concurrency property.*
2. *For each cycle, $C = (e_0, ..., e_n)$, in $\mathcal{C}$, with associated vertex sequence $(v_0, ..., v_n, v_0)$, for any pair of consecutive edges, e_i and e_{i+1} in C*

   $$|E_1(e_i, e_{i+1}, v_{i+1}) \cap EulerEdges(HG(d))| = |E_2(e_i, e_{i+1}, v_{i+1}) \cap EulerEdges(HG(d))|$$

   and

   $$|E_1(e_i, e_{i+1}, v_{i+1}) \cap E(C)| = |E_2(e_i, e_{i+1}, v_{i+1}) \cap E(C)|$$

   where we take $e_{n+1} = e_0$.
3. *For each pair of distinct cycles, $C = (e_0, ..., e_n)$ and $C' = (e'_0, ..., e'_m)$, in $\mathcal{C}$, with associated vertex sequences $(v_0, ..., v_n, v_0)$ and $(v'_0, ..., v'_m, v'_{m+1})$ respectively, given any pair of consecutive edges, e_i and e_{i+1}, in C if there exists an edge, e'_j in C' such that $v_{i+1} = v'_{j+1}$ then*

   $$e'_j \in E_1(e_i, e_{i+1}, v_{i+1}) \quad \text{and} \quad e'_{j+1} \in E_2(e_i, e_{i+1}, v_{i+1}).$$

The diagram obtained by adding a contour to d using $\mathcal{C}$ possesses the crossings property if and only if $\mathcal{C}$ and d possess their respective crossings property.

10.6. Connected Zones

Our final wellformedness condition is that of connected zones and it is linked to when we split a zone. This wellformedness condition is different from all of the others: it may be broken by a diagram in the inductive construction, but the final (required) diagram may still have connected zones. For each of the other conditions, once they are broken in the construction they remain broken. We describe how to add a contour in such a manner that the resulting diagram has connected zones.

Definition 10.7. *Let $d = (Curve, l)$ be an atomic Euler diagram all of whose zones comprise at most two minimal regions. Let $\mathcal{C}$ be a multi-set of cycles in the hybrid graph, $HG(d)$. Then $\mathcal{C}$ possesses the* ***connected zones property*** *whenever:*

1. *for each zone, z, in d that comprises a single minimal region, m,*
   (a) *there is at most one cycle, C, in $\mathcal{C}$ that contains an edge incident with a (dual) vertex of $HG(d)$ embedded inside m, and*
   (b) *if there is a cycle, C, that passes through m then exactly one subsequence of edges in C pass through m, and*
2. *for each zone, z, in d that comprises two minimal regions, m_1 and m_2,*
   (a) *no cycle in $\mathcal{C}$ passes through any vertex of $HG(d)$ embedded in z, and*
   (b) *exactly one of m_1 and m_2 is inside $\mathcal{C}$.*

In part 1(b) of the above definition, if $C = (e_0, ..., e_n)$ then $(e_i, e_{i+1}, ..., e_{i+m})$ is a subsequence of C of length $m+1$ whereas, for example, (e_1, e_3). The only minimal region that contains such a sequence of edges of length more than 2 is the unbounded one outside all of the curves. To conclude, the diagram obtained from d by adding a contour using $\mathcal{C}$ possesses the connected zones property if and only if $\mathcal{C}$ possesses the connected zones property, provided each zone in d comprises at most two minimal regions.

11. Conclusion

In this paper we have provided a novel area-proportional Euler diagram drawing method. It advances the existing state-of-the-art, in that it can draw every area-proportional abstract description. Moreover, our method can readily incorporate user preferences as to which well-formedness properties are to be possessed by the drawn diagram. Previously developed area-proportional drawing methods could not draw the majority of abstract descriptions and were very limited.

There are various avenues for future work. In particular, we seek further heuristics to help identify cycles for contour addition so that we can (a) improve the efficiency of the drawing method, and (b) produce better diagrams. Along the same lines, we plan to give a more detailed consideration as to how to choose a decomposition that results in better drawn diagrams being produced. In addition, there are likely to be efficiency savings if we only produce a subgraph of the hybrid graph (step 4(b) in the drawing algorithm), including only vertices and edges that may be included in part of a cycle for the new contour.

We plan to provide an implementation of the drawing method, which will allow more users to access the power of area-proportional Euler diagrams as a visualization technique. The feasibility of this plan is demonstrated by the (partial) implementation of the non-area-proportional inductive method. That implementation allows the addition of contours that consist of a single, simple closed curve [25]. A further significant avenue of research includes devising layout improvement methods, following [15]. Such methods will be an important addition to the drawing method, since the aesthetic quality of the diagrams is likely to impact their ability to convey information effectively. The existing layout improvement methods move the curves of the diagram, measuring the quality of the diagram at each stage to ensure that improved layouts result. However, moving curves changes the areas of the diagram's regions and, thus, extending to the area-proportional case will be challenging. We envisage that techniques which improve the diagram layout after each curve addition, altering only the most recently added curve, are most likely to be practically implementable.

Acknowledgements The authors are supported by the EPSRC for the *Visualization with Euler diagrams* project (EP/E011160/1 and EP/E010393/1).

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] S. Chow and F. Ruskey. Drawing area-proportional Venn and Euler diagrams. In *Graph Drawing*, pages 466–477. Springer, 2003.

[5] S. Chow and F. Ruskey. Towards a general solution to drawing area-proportional Euler diagrams. In *Euler Diagrams,* pages 3–18, ENTCS 134, 2005.

[6] R. DeChiara, U. Erra, and V. Scarano. VennFS: A Venn diagram file manager. In *Information Visualisation*, pages 120–126. IEEE, 2003.

[7] Anthony W.F. Edwards. Venn diagrams for many sets. *New Scientist*, 7:51–56, 1989.

[8] L. Euler. Lettres a une princesse d'allemagne sur divers sujets de physique et de philosophie. *Letters*, 2:102–108, 1775.

[9] M. Farfel et al., An overview of 9/11 experiences and respiratory and mental health conditions among world trade center health registry enrolles. *Journal of Urban Health*, 85(6):880–909, 2008.

[10] 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.

[11] A. Fish and J. Flower. Abstractions of Euler diagrams. In *Euler Diagrams*, pages 77–101, ENTCS 134, 2005.

[12] A. Fish and J. Flower. Euler diagram decomposition. In *Diagrams*. Springer, 2008.

[13] J. Flower and J. Howse. Generating Euler diagrams. In *Diagrams*, pages 61–75, Springer, 2002.

[14] J. Flower, J. Howse, and J. Taylor. Nesting in Euler diagrams: syntax, semantics and construction. *Software and Systems Modelling*, 3:55–67, 2004.

[15] J. Flower, P. Rodgers, and P. Mutton. Layout metrics for Euler diagrams. In *Information Visualisation*, pages 272–280. IEEE, 2003.

[16] E. Ip. Visualizing multiple regression. *Journal of Statistics Education*, 9(1), 2001.

[17] H. Kestler et al., Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9(67), 2008.

[18] 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.

[19] P. Rodgers, J. Flower, G. Stapleton, and J. Howse. Some results for drawing area-proportional Venn-3 with convex curves. In *Information Visualization*, pages 667–672. IEEE, 2009.

[20] P. Rodgers, L. Zhang, and A. Fish. General Euler diagram generation. In *Diagrams*. Springer, September 2008.

[21] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. *Computer Graphics Forum*, 28(3), 2009.

[22] J. Soriano, K. Davis, B. Coleman, G. Visick, D. Mannino, and N. Pride. The proportional Venn diagram of obstructive lung disease. *Chest*, 124, 2003.

[23] G. Stapleton, J. Howse, and P. Rodgers. A graph theoretic approach to general Euler diagram drawing. *Theoretical Computer Science*, 411(1):91–112, 2010.

[24] G. Stapleton, P. Rodgers, J. Howse, and J. Taylor. Properties of Euler diagrams. In *Proceedings of Layout of Software Engineering Diagrams*, pages 2–16. EASST, 2007.

[25] G. Stapleton, P. Rodgers, J. Howse, and L. Zhang. Inductively generating Euler diagrams. *IEEE Transactions of Visualization and Computer Graphics*, 17(1):88-100, 2011.

[26] J. Thièvre, M. Viaud, and A. Verroust-Blondet. Using Euler diagrams in traditional library environments. In *Euler Diagrams*, pages 189–202. ENTCS 134, 2005.

[27] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *Phil.Mag*, 1880.

[28] A. Verroust and M.-L. Viaud. Ensuring the drawability of Euler diagrams for up to eight sets. In *Diagrams* pages 128–141, Springer, 2004.