

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>

Contents lists available at [SciVerse ScienceDirect](#)

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlcAutomatically drawing Euler diagrams with circles[☆]Gem Stapleton^{a,*}, Jean Flower^a, Peter Rodgers^b, John Howse^a^a Visual Modelling Group, University of Brighton, UK^b University of Kent, UK

ARTICLE INFO

Article history:

Received 25 July 2011

Accepted 15 February 2012

Available online 3 April 2012

Keywords:

Euler diagrams

Information visualization

Non-hierarchical data visualization

Venn diagrams

ABSTRACT

Euler diagrams are used for visualizing categorized data. These categories, together with information about when categories share some datum, can be turned into a succinct diagram description from which an Euler diagram can be generated. Closed curves represent the categories and the relationships between the curves (such as containment) correspond to relationships between the categories (such as subset). A range of automated Euler diagram drawing methods have been proposed but they often produce diagrams that are aesthetically unpleasing, can be computationally complex and most of them cannot draw a diagram for some (often many) given collections of categories. One such method is capable of drawing aesthetically pleasing Euler diagrams, using only circles, and is computationally efficient (being of polynomial time complexity) but it applies to a very restricted subset of collections of categorized data. This paper substantially extends that method so it can always draw an Euler diagram, that is it applies to all collections of categorized data. In particular, we identify a class of diagram descriptions that can be drawn with circles, generalizing previous work. For diagram descriptions outside of this class, we define transformations that can be used to turn them into descriptions inside the 'drawable with circles' class. We demonstrate how such transformations can be done in a general, a process during which many choices must be made. Further, we provide strategies for making particular choices which ensure desirable properties, such as curve containment, are preserved. We have provided a software implementation of the drawing method, which is freely available from www.eulardiagrams.com/inductivecircles.htm.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

It is commonplace to use visual representations of data since a visualization can give insight into properties of the data. Software tools support information visualization; for instance, Excel incorporates various different options for data visualization, such as the ability to automatically

create a bar chart from appropriate data. In this paper, we are concerned with the automated visualization of data that falls into categories (sets). Fig. 1 displays information concerning three categories, asthma, chronic bronchitis, and emphysema, obtained from [1]. The diagram used is called a Venn diagram, of which Euler diagrams are generalizations; in a Venn diagram all intersections between the categories must be represented, unlike Euler diagrams. In Fig. 1, each category is represented by a unique curve. Fig. 2, also from [1], visualizes five categories, where one of them is represented by seven curves.

Euler diagrams are used for information visualization in a wide range of areas such as: crime control [2], computer file organization [3], classification systems [4],

[☆] This paper has been recommended for acceptance by Shi Kho Chang.

* Corresponding author. Tel.: +44 1273 642410.

E-mail addresses: g.e.stapleton@brighton.ac.uk (G. Stapleton), p.j.rodgers@kent.ac.uk (P. Rodgers), john.howse@brighton.ac.uk (J. Howse).

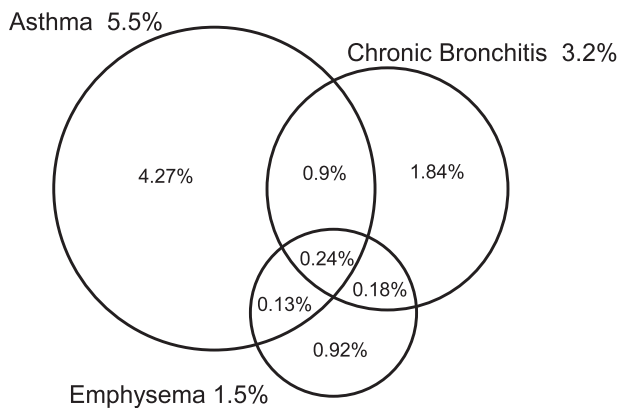


Fig. 1. Visualizing medical data.

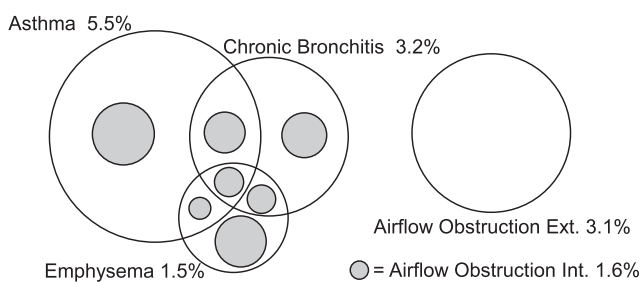


Fig. 2. Using multiple curves to represent a category.

education [5], and genetics [6]. In order to readily visualize information from these domains, automated layout methods are required. In addition, Euler diagrams form the basis of a variety of visual logics, such as Swoboda and Allwein's Euler/Venn logic [7], Shin's Venn-II system [8], Howse et al.'s spider diagrams [9], and Delaney and Stapleton's spider diagrams of order [10]. These visual logics are fairly limited in expressiveness, unlike Oliver et al.'s concept diagrams [11] and Kent's constraint diagrams [12] (later extended to generalized constraint diagrams [13]). In all of these logics, automated diagram layout is also important, such as in the context of automated theorem proving where it is necessary to produce a visualization of an automatically generated proof [14].

Various methods have been devised to draw Euler diagrams automatically, including [15–19]. Some of these methods use only particular geometric shapes such as circles which are typically used in manually drawn Euler diagrams: as an indication of this, Wilkinson identifies that out of 72 Euler diagrams used in articles appearing in *Science*, *Nature* and online affiliated journals during 2009, 65 (90%) use circles [20]. We note that Euler only used circles when drawing Euler diagrams (in fact, they were called Euler circles) [21]. In Euler's original usage of these diagrams, there were expressive limitations in that often diagrams cannot be drawn when each category is represented by a single circle: the geometric constraints placed on the diagrams by the use of circles cannot be satisfied by all collections of intersections. For instance, when four categories are to be represented and all possible intersections between them must be displayed, no four circles can

overlap in the required manner; the required diagram is a Venn diagram with four closed curves, which can be drawn with ellipses [22].

Venn diagrams [23] do not suffer from the same expressiveness limitation as Euler diagrams: the closed curves are not restricted to being circles but can be arbitrary shapes. Venn incorporated shading to allow one to assert that particular intersections are empty. Unfortunately, Venn diagrams too have their problems: the curves can become very convoluted shapes, even when only a limited number of categories are represented.

Thus, it would seem sensible to use the best of both approaches: circles from Euler diagrams, which lack the complexity of arbitrary shaped curves, together with shading to allow one to assert emptiness. This blend permits much better layouts to be achieved than is possible with current drawing methods, as we will demonstrate later in the paper (Section 11). Previously, Stapleton et al. devised a method for drawing a very restricted subset of Euler diagrams using circles [24]; this paper substantially extends that method so that it can always draw a diagram with circles using shading where necessary.

We will now give a section-by-section overview of the paper, with the drawing process being summarized in Fig. 3. In Section 2, we illustrate our drawing method and identify some of the challenges that will be overcome. In Section 3 we define Euler diagrams and a number of associated concepts. We define diagram descriptions in Section 4 along with concepts that are analogous to those defined for Euler diagrams; we prove that the concepts coincide as intended. We define operations that add curves to, and remove curves from, diagrams and their descriptions in Section 5; again we establish that these operations coincide. Section 6 presents definitions of clusters and piercings at both diagram and description levels. These concepts are required for our definition of inductively pierced diagrams and inductively pierced descriptions, which are given in Section 7; inductively pierced diagrams are drawn using circles. A key outcome of Section 7 is a proof that (a) every inductively pierced diagram has an inductively pierced description and (b) every inductively pierced description can be drawn as an inductively pierced diagram. This means that if we transform a non-inductively pierced description into one that is inductively pierced then we can draw it using circles. There are many descriptions that are not inductively pierced and Section 8 demonstrates how to transform any description into an inductively pierced description. We prove that the transformation process respects the semantics: if the description with which we start represents some data then so too does the description that results after the transformation process. Section 9 focuses on the choices that can be made during the transformation process, presenting strategies that can be used to ensure containment and disjointness properties are preserved. Our software implementation of the method is demonstrated in Section 10, where we describe how to draw a diagram given an inductively pierced description. Section 11 gives an overview of existing drawing methods to allow for comparison with our results.

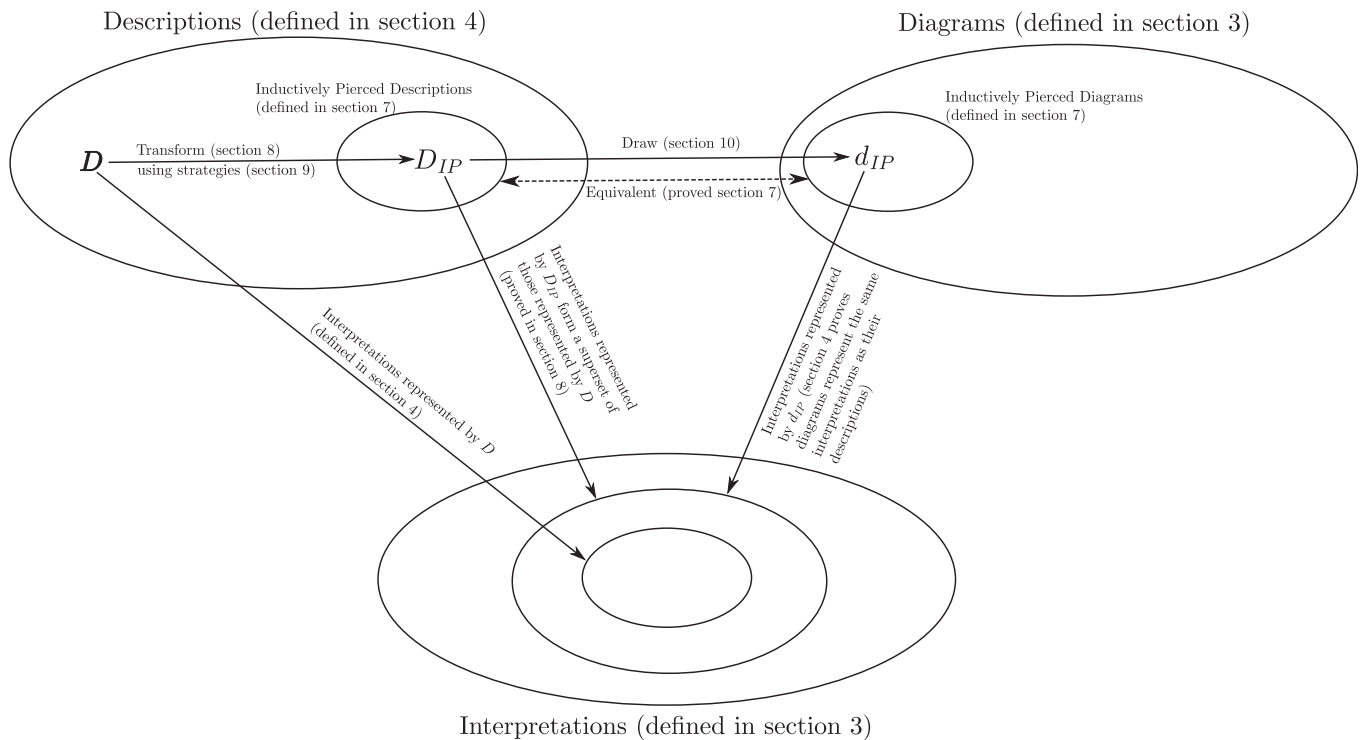


Fig. 3. An overview of the main results in the paper.

2. Overview of our drawing method

Our drawing method uses only circles and, since it draws one curve at a time, is inductive. In addition, we allow the use of multiple curves to represent a given set. Euler diagram drawing methods start with a description of the required diagram, which may be in the form of the sets (categories) to be represented or simply a description of the set intersections to be displayed. To illustrate our approach, suppose we wish to draw a diagram representing sets P , Q , and R with intersections described by \emptyset , $\{P\}$, $\{Q\}$, $\{P,Q\}$, $\{P,R\}$, $\{P,Q,R\}$; \emptyset describes the intersection $(U-P) \cap (U-Q) \cap (U-R)$, where U is the universal set which contains all elements, and $\{P\}$ describes $P \cap (U-Q) \cap (U-R)$ and so forth, with each region in the diagram that represents a set intersection being called a *zone*. Now, since we will sometimes use more than one curve to represent a set, we generalize the (just illustrated) typical notion of a description, D , so that it is more fine-grained. It comprises a set, C , whose elements are called abstract curves, a set, B , of descriptions of regions, called *abstract basic regions*, formed by these curves, and a labelling function, l , that assigns a label to each abstract curve. A basic region in a diagram is a region that can be described as being inside certain curves but outside the rest. In this example, we would instead write D as

1. $C = \{\kappa_1, \kappa_2, \kappa_3\}$,
2. $B = \{\emptyset, \{\kappa_1\}, \{\kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa_1, \kappa_3\}, \{\kappa_1, \kappa_2, \kappa_3\}\}$, and
3. $l(\kappa_1) = P$, $l(\kappa_2) = Q$, and $l(\kappa_3) = R$.

We draw a circle for each abstract curve in some order, say κ_1 then κ_2 then κ_3 . This process would give the sequence of diagrams shown in Fig. 4. We note that each

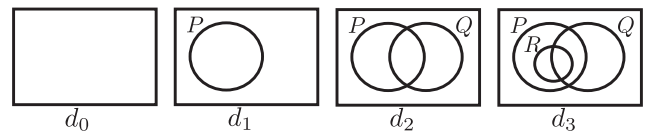


Fig. 4. Drawing one curve at a time.

κ_i carries with it no information about the centre or radius of the to-be-drawn circle; these are determined during the drawing process.

As a slightly more complex example, suppose we start with the description, D_1 , comprising

1. $C_1 = \{\kappa_1, \kappa_2, \kappa_3\}$,
2. $B_1 = \{\emptyset, \{\kappa_1\}, \{\kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa_1, \kappa_3\}, \{\kappa_2, \kappa_3\}\}$, and
3. $l_1(\kappa_1) = P$, $l_1(\kappa_2) = Q$, and $l_1(\kappa_3) = R$.

No diagram drawn with circles has description D_1 . See d_1 in Fig. 5 for a diagram with this description, where R is intended to run concurrently with parts of P and Q . This is clearly an undesirable diagram: the relationship between R and the other curves is somewhat unclear. Since our method produces only diagrams drawn with circles, we transform D_1 into another description, D_2 , that can be drawn with circles. The description is transformed in two ways: abstract curves are ‘split’ into more abstract curves, and abstract basic regions are added; in a drawn diagram, to indicate that the extra basic regions were not requested, they are shaded. Whilst we omit the details, D_2 is

1. $C_2 = \{\kappa_1, \kappa_2, \kappa_4, \kappa_5\}$,
2. $B_2 = \{\emptyset, \{\kappa_1\}, \{\kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa_1, \kappa_4\}, \{\kappa_2, \kappa_5\}\}$, and
3. $l_2(\kappa_1) = P$, $l_2(\kappa_2) = Q$, $l_2(\kappa_4) = R$ and $l_2(\kappa_5) = R$.

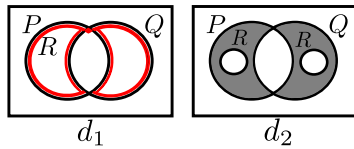


Fig. 5. Problems.

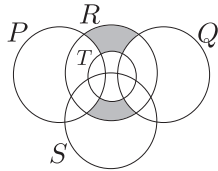


Fig. 6. Adjacency issues.

Essentially, we have split the abstract curve κ_3 into two abstract curves, κ_4 and κ_5 , and added abstract basic regions in order to avoid concurrency between curves. The diagram d_2 in Fig. 5 has description D_2 .

When we split an abstract curve into many such curves, we do so in a manner that guarantees that the result can be drawn with circles. A problem here is that this relies on being able to identify when two abstract basic regions necessarily represent topologically adjacent basic regions in a diagram. If we consider two topologically adjacent basic regions, intuitively one can cross a curve to move from one of these basic regions to the other. For instance, in d_2 of Fig. 5, we can move from the (shaded) basic region inside just P (that with description $\{\kappa_1\}$) to the basic region inside both P and the curve inside P labelled R (the basic region $\{\kappa_1, \kappa_4\}$) by crossing this curve labelled R (corresponding to κ_4). Intuitively it might seem as though basic regions are topologically adjacent whenever their abstractions differ by a single abstract curve, assuming no curves run concurrently. However, there are descriptions that, even when drawn with circles and have no concurrency between the curves, contain abstract basic regions that differ by a single abstract curve but do not correspond to topologically adjacent basic regions. An example is in Fig. 6, where the two shaded basic regions, that inside just R and that inside both R and S , have descriptions that differ by a single abstract curve but they are not adjacent.

In summary, there are a number of challenges:

1. We need to identify when a description can be drawn using circles. This is hard because descriptions do not necessarily capture the topological properties of their drawings and it is these properties that determine whether we can add a circle in the desired manner. We identify a class of abstract descriptions where we can necessarily identify the topological adjacency of basic regions. Descriptions in this class are called *inductively pierced*, generalizing [24,25]. The earlier sections work towards this, culminating in a proof that inductively pierced descriptions can be drawn with circles (Theorem 7.2).
2. When a description has not been identified as drawable with circles (i.e. it is not inductively pierced), we

must transform it into one that can be drawn with circles. Problems include determining how to split up an abstract curve so that each one can be drawn as a circle and knowing which abstract basic regions to add to ensure that this is possible. Sometimes adding regions is sufficient and sometimes we only need to split curves. This was not considered at all in [24]. In [25], which used the less fine-grained descriptions comprising labels and zones, a set of zones to add was identified and we generalize that approach in this paper. Moreover, [25] did not address splitting abstract curves: splitting occurred only when rendering curves during the final stages of the drawing process. Splitting at the abstract level is advantageous in that it allows us to properly analyze the impact of splitting on the properties of the description and, therefore, diagram. A further novel contribution in this paper is a method to transform any given description into one that is inductively pierced. Strategies for making informed choices when adding abstract basic regions and splitting abstract curves are presented in Section 9. We further establish, in Theorem 9.2, that the strategies preserve so-called well-matchedness [26].

3. We aim to produce effective drawings and our algorithm will make informed choices about circle centres and radii, so that the drawn diagrams respect natural alignments of circles where possible. The results we present in Section 10 significantly extend [24,25].

Some automatically drawn diagrams produced by our software can be seen in Figs. 7 and 8. The diagram in Fig. 7

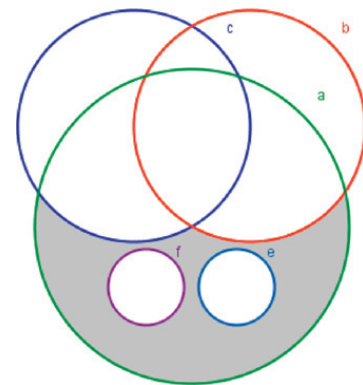


Fig. 7. An extra region.

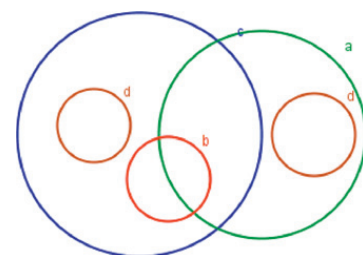


Fig. 8. A split curve.

has a shaded region, indicating it represents the empty set, and the diagram in Fig. 8 has a set represented by two curves.

3. Euler diagrams

We now present a series of definitions that capture a range of concepts that we require. The definitions in this section are, typically, standard and commonly seen in the literature, with the exception being the definition of basic regions which is new to this paper.

An Euler diagram comprises a set of closed curves drawn in \mathbb{R}^2 , where each curve has a label chosen from a set \mathcal{L} . In Fig. 9, the diagram has four closed curves, one is labelled P , two are labelled Q and the other is labelled R . Recall that a closed curve in the plane is a continuous function of the form $c : [a, b] \rightarrow \mathbb{R}^2$ where $c(a) = c(b)$. Given an arbitrary function, $f : A \rightarrow B$, we write $image(f)$ to denote the set of elements in B to which f maps.

Definition 3.1. An Euler diagram is a pair, $d = (Curve, l)$, where $Curve$ is a finite set of closed curves each with codomain \mathbb{R}^2 , and $l : Curve \rightarrow \mathcal{L}$ is a function that returns the label of each curve. If all of the curves in $Curve$ are circles then we say d is drawn with circles.

We observe that the closed curves essentially provide a partition of the plane into *minimal regions*. In Fig. 9, the diagram's curves give rise to seven minimal regions, including the region outside all four curves.

Definition 3.2. A *minimal region*, mr , of an Euler diagram $d = (Curve, l)$ is a connected component of

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

The set of minimal regions in d is denoted $MR(d)$. Given a minimal region, mr , and a curve, c , in d , mr is *inside* c if each point in mr is inside c . We denote the set of curves that mr is inside by $I(mr)$.

The diagram in Fig. 10 has nine minimal regions. There are two minimal regions, mr_1 and mr_2 , that are inside only the leftmost curve labelled R ; that is $I(mr_1) = I(mr_2) = \{c\}$ where c is that curve labelled R . These two minimal

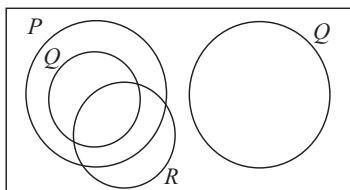


Fig. 9. An Euler diagram.

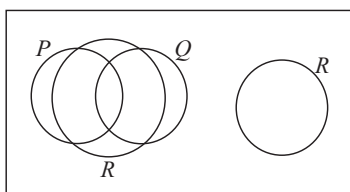


Fig. 10. Basic regions.

regions form a *basic region*: a basic region is a set of minimal regions that are all contained by the same curves. So, Fig. 10 has eight basic regions.

Definition 3.3. A *basic region*, br , in an Euler diagram $d = (Curve, l)$ is a non-empty set of minimal regions for which there exists a subset of $Curve$, say \mathcal{CUR} , such that $br = \{mr \in MR(d) : I(mr) = \mathcal{CUR}\}$.

The set of basic regions in d is denoted $BR(d)$. Given a basic region, br , \mathcal{CUR} is the set of curves that br is inside, denoted $I(br) = \mathcal{CUR}$.

In Fig. 11, the set R is represented by two curves. Since, between them, these two curves are contained by curves labelled P and Q , the diagram expresses that R is a subset of $P \cup Q$. In addition, the diagram expresses that P and Q are disjoint, since no points are inside both P and Q . The basic region inside both P and R represents the set $P \cap R \cap (U - Q)$, where U is called the universal set and contains all elements. The basic region outside all curves represents the set $(U - P) \cap (U - Q) \cap (U - R)$. Given a basic region, we can determine the set represented by the labels of the curves by which it is contained. If a basic region, br , is inside curves c_1, \dots, c_n then we compute the parity of the number of occurrences of each label in the list $l(c_1), \dots, l(c_n)$: if a label λ occurs an odd number of times in the list then the set represented by br , say $set(br)$, is a subset of the set represented by λ , say $set(\lambda)$. However, if λ is in d but does not occur an odd number of times then $set(br)$ is a subset of the complement of the set represented by λ .

Definition 3.4. An *interpretation* is a pair, (U, set) , where U is a set, called the *universal set*, and $set : \mathcal{L} \rightarrow U$ is an assignment of sets to curve labels. Given a disjoint pair, (A_1, A_2) , of sets of labels, set is extended to interpret (A_1, A_2) as

$$set(A_1, A_2) = \bigcap_{\lambda \in A_1} set(\lambda) \cap \bigcap_{\lambda \in A_2} (U - set(\lambda)).$$

Linking interpretations back with our data visualization problem, if we have sets $S_1 = \{Harold, Maude\}$, $S_2 = \{Bill, Ted\}$ (i.e. two disjoint categories in which data lie), and $U = \{Harold, Maude, Bill, Ted, Tom, Jerry\}$ is the universal set, then an interpretation basically assigns names, which are curve labels, to the categories:

1. $set(LikesFunerals) = S_1$
2. $set(HasAdventure) = S_2$.

An Euler diagram representing this interpretation is seen in Fig. 12, where we have annotated the diagram with the

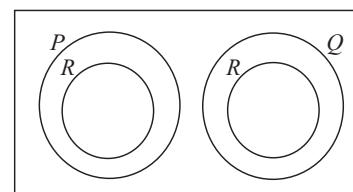


Fig. 11. Depicting sets.

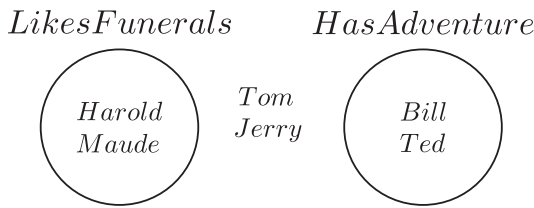


Fig. 12. Representing an interpretation.

names of the individuals represented. In examples, we will typically blur the distinction between λ and $set(\lambda)$.

Definition 3.5. Let $d = (Curve, l)$ be an Euler diagram. Let br be a basic region in d . We define $A_1(br)$ to be the set of labels, λ , such that λ labels an odd number of curves in $l(br)$. We further define $A_2(br)$ to be the set of labels, λ , such that λ labels an even number of curves in $l(br)$, so $A_2(br) = image(l) - A_1(br)$.

Definition 3.6. Let $d = (Curve, l)$ be an Euler diagram and let br be a basic region in d . Let (U, set) be an interpretation. The set denoted by br is

$$set(br) = set(A_1(br), A_2(br))$$

We say that d represents (U, set) provided

$$\bigcup_{br \in BR(d)} set(br) = U.$$

Thus, if we have a collection of sets whose relationships we wish to visualize using an Euler diagram, we must find such a diagram whose basic regions, between them, represent the universal set. Later, we will define analogous concepts for diagram descriptions and show that they coincide. This is important, since later we will be applying transformations to descriptions. We will need to understand how such transformations impact the semantics¹ and that any changes to the semantics are ‘safe’ (i.e. the diagram we finally draw represents the required interpretation).

Given a set, A , of curve labels, the set of basic regions, br , in d with $A_1(br) = A$ is called a *zone*. The basic regions that comprise a zone all represent the same set. Typically, the semantics of Euler diagrams are (equivalently) defined by using the diagram’s zones, but for our purposes using basic regions makes some of the details later in the paper more straightforward.

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

1. All of the curves are simple.
2. No pair of curves runs concurrently.
3. There are no triple points of intersection between the curves (i.e. there are no points that are mapped to more than twice by the curves).
4. Whenever two curves intersect, they cross.

5. Each basic region is connected (i.e. consists of exactly one minimal region).
6. Each curve label is used on at most one curve.

Definitions of these properties can be found in [27], except that for basic region connectedness (property 5) which is more typically stated for zones (a *connected zones* condition); for our purposes weakening the condition to basic regions is helpful. Fig. 9 possesses all properties except that the label Q is used twice. In Fig. 10, R is used twice and the disconnected basic region inside the contour R comprises three minimal regions. Our drawing method allows curve labels to be used more than once (that is, the labelling function l is not necessarily injective), but it ensures that all other properties hold.

Definition 3.7. An Euler diagram that possesses all of the properties except that l is not necessarily injective is *wellformed up to labelling*.

4. Diagram descriptions

In order to draw an Euler diagram automatically, we can start with a description of the sets to be represented and the relationships between those sets. In essence, this is a list of zone descriptions (each zone can be described by the labels of the curves by which it was specified). For example, the diagram in Fig. 11 contains five zones which can be described by: \emptyset (the zone outside all three contours), $\{P\}$ (the zone inside just the contour P), $\{Q\}$ (the zone inside just the contour Q), $\{P, R\}$ (the zone inside both the contours P and R), and $\{Q, R\}$ (the zone inside both the contours Q and R). These zone descriptions are sometimes called *abstract zones*. Other diagrams have the same zone descriptions but contain only one curve labelled R , for example. A zone-based description (sometimes called a set system [28]) is typically used in other methods that draw Euler diagrams [16,29,30].

Our drawing method explicitly identifies the number of curves that will be used to represent each set and, moreover, to which basic regions those curves give rise. Therefore, as stated previously, it is beneficial to extend the notion of a description to identify the curves present, their labels and the basic regions. For example, in Fig. 11, such a description, D , of this diagram, d , would be

1. a set of abstract curves $C = \{\kappa_P, \kappa_Q, \kappa_R, \kappa_{R'}\}$,
2. a set of abstract basic regions
 $B = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_{R'}\}$,
 and
 3. a labelling function, $l: C \rightarrow \mathcal{L}$ where
 (a) $l(\kappa_P) = P$,
 (b) $l(\kappa_Q) = Q$,
 (c) $l(\kappa_R) = R$, and
 (d) $l(\kappa_{R'}) = R$.

We say that D is a *description* of d and that d is a *drawing* of D . From the basic region $\{\kappa_P\}$, we can derive the abstract zone $\{l(\kappa_P)\} = \{P\}$. We see, therefore, that the

¹ The semantics of an Euler diagram can be viewed as the set of interpretations that it represents.

abstract zones arising from D are \emptyset , $\{P\}$, $\{Q\}$, $\{P,R\}$, and $\{Q,R\}$.

We introduce here a countably infinite set, denoted C , whose elements are called *abstract curves*; note that these are not curves in the mathematical or geometrical sense. The set $B = \mathbb{P}(C)$ contains elements called *abstract basic regions*.

Definition 4.1. A *diagram description*, or simply *description*, D , is a tuple, (C,B,l) , where

1. C is a finite set of abstract curves,
2. $B \subseteq \mathbb{P}C$ is a set of abstract basic regions such that $\emptyset \in B$, and
3. $l : C \rightarrow \mathcal{L}$ is a function that returns the label of each abstract curve.

Given a description D , we define $C(D) = C$, $B(D) = B$ and $l_D = l$.

Definition 4.2. Given an Euler diagram, $d = (Curve, l_1)$, and a description $D = (C,B,l_2)$, we say that D is a *description* of d provided there exists a bijection, $f : Curve \rightarrow C$, such that

1. labels are preserved: for each curve $c \in Curve$, $l_1(c) = l_2(f(c))$, and
2. the basic regions are preserved: f induces a bijection, $g : BR(d) \rightarrow B$, defined by

$$g(br) = \{f(c) : c \in I(br)\}.$$

If D is a description of d then d is a *drawing* of D . Such an f and g are said to *identify* D as a description of d and, also, d as a drawing of D .

Any given diagram, therefore, can have many different descriptions. However, these descriptions are unique up to the choice of abstract curves. Such descriptions are *isomorphic* to each other.

Definition 4.3. Let $D_1 = (C_1, B_1, l_1)$ and $D_2 = (C_2, B_2, l_2)$ be descriptions. We say that D_1 and D_2 are *isomorphic* provided there exists a bijection, $\theta : C_1 \rightarrow C_2$, such that

1. the curve labels are preserved: for each $\kappa_1 \in C_1$, $l_1(\kappa_1) = l_2(\theta(\kappa_1))$, and
2. the abstract basic regions are preserved: the induced function $\phi : B_1 \rightarrow B_2$ defined by

$$\phi(\beta) = \{\theta(\kappa) : \kappa \in \beta\}$$

is a bijection.

Lemma 4.1. Let d be a diagram with descriptions $D_1 = (C_1, B_1, l_1)$ and $D_2 = (C_2, B_2, l_2)$. Then D_1 and D_2 are *isomorphic*.

Proof. Suppose that f_1 and g_1 , and f_2 and g_2 , identify D_1 and D_2 , respectively, as descriptions of d . Then the label preserving bijection $\theta : C_1 \rightarrow C_2$ defined by $\theta(\kappa) = f_2(f_1^{-1}(\kappa))$ induces an appropriate bijection between the abstract basic regions. \square

Thus, from now on we will simply refer to *the* description of a diagram, rather than *a* description of a diagram.

Diagram semantics are determined by the basic regions in the diagram; above, we defined $set(br)$ to be the set denoted by basic region br and asserted that the basic regions in the diagram must, between them, represent the universal set, U . For a description, the set represented by an abstract basic region is defined as follows:

Definition 4.4. Let $D = (C,B,l)$ be a description. Let β be an abstract basic region in D . We define $A_1(\beta)$ to be the set of labels, λ , in D such that λ labels an odd number of curves in β . We further define $A_2(\beta)$ to be the set of labels, λ , in D such that λ labels an even, possibly zero, number of curves in β , so $A_2(\beta) = image(l) - A_1(\beta)$. In cases where we have more than one description under consideration, to disambiguate we will write $A_1(\beta, D)$ and $A_2(\beta, D)$ and so forth.

Definition 4.5. Let β be an abstract basic region in description $D = (C,B,l)$ and let (U, set) be an interpretation. Then

$$set(\beta) = set(A_1(\beta), A_2(\beta)).$$

We say that D *represents* (U, set) provided

$$\bigcup_{\beta \in B} set(\beta) = U.$$

Again, in cases where we have more than one description, to disambiguate we will write $set(\beta, D)$.

We will now justify that the mapping between diagrams and descriptions is well-defined with respect to the diagram semantics:

Lemma 4.2. Let $d = (Curve, l_1)$ be a diagram with description $D = (C, B, l_2)$ and let (U, set) be an interpretation. Then d *represents* (U, set) if and only if D *represents* (U, set) , that is:

$$\bigcup_{br \in BR(d)} set(br) = U \Leftrightarrow \bigcup_{\beta \in B(D)} set(\beta) = U.$$

Proof. Take f and g that identify D as a description of d . Let br be a basic region in d . We start by showing that $A_1(br) = A_1(g(br))$ and $A_2(br) = A_2(g(br))$. Let $\lambda \in A_1(br)$. Then λ occurs an odd number of times in the list

$$l_1(c_1), \dots, l_1(c_n),$$

where $I(br) = \{c_1, \dots, c_n\}$. So, because f is bijective and label preserving, λ occurs an odd number of times in the list

$$l_2(f(c_1)), \dots, l_2(f(c_n)).$$

Since $g(br) = \{f(c_1), \dots, f(c_n)\}$ it follows that $\lambda \in A_1(g(br))$. Hence $A_1(br) \subseteq A_1(g(br))$. Similar arguments allow us to deduce that $A_1(g(br)) \subseteq A_1(br)$ and $A_2(br) = A_2(g(br))$. Thus, $set(br) = set(g(br))$. Since g is a bijection, the result follows trivially. \square

Hence, **Lemma 4.2** establishes that the semantics of a diagram are identical to those of its description. This means that, if we start our drawing process with description D_1 ,

derived from the interpretation, I , to be represented, then a drawing of D_1 represents I . If we apply a transformation to D_1 to give description D_2 then we must be sure that D_2 also represents I , so that any drawings of D_2 also represent I . We establish that this is the case for the transformations that we develop in Section 8.

The transformations potentially change properties like curve containment. One of our transformations adds abstract basic regions to a description. For instance, if we add a single abstract basic region to the description of d_1 in Fig. 13 to give the description of d_2 then the curve labelled Q is no longer contained by the curve labelled P . Similarly, if two curves have interiors with an empty intersection (i.e. they represent disjoint sets) then adding regions could actually make them have a non-empty intersection. As we will demonstrate, it is possible to transform a description, D_1 , into an inductively pierced description, D_2 (defined later), in such a way that containment and disjointness properties are preserved. To this end, we need to define a notion of containment and disjointness for abstract curves.

There is a well-understood notion of what it means for one curve to contain another: curve c_1 contains c_2 in an Euler diagram, d , if and only if all of the basic regions inside c_2 are also inside c_1 . In addition, two curves are disjoint when there is no region inside them both. We use these observations to define abstract curve containment and disjointness.

Definition 4.6. Let $D = (C, B, l)$ be a description and let κ_1 and κ_2 be abstract curves in D . Then κ_1 contains κ_2 , denoted $\kappa_2 \subseteq \kappa_1$, in D provided

$$\{\beta \in B : \kappa_2 \in \beta\} \subseteq \{\beta \in B : \kappa_1 \in \beta\}.$$

If

$$\{\beta \in B : \kappa_2 \in \beta\} \subset \{\beta \in B : \kappa_1 \in \beta\}$$

then κ_1 properly contains κ_2 , denoted $\kappa_2 \subset \kappa_1$. In addition, κ_1 and κ_2 are disjoint, denoted $\kappa_1 \cap \kappa_2 = \emptyset$, provided

$$\{\beta \in B : \kappa_1 \in \beta\} \cap \{\beta \in B : \kappa_2 \in \beta\} = \emptyset.$$

The following lemma, the proof of which is straightforward, establishes that the notions of containment and disjointness for abstract curves in descriptions coincide with those for curves in diagrams.

Lemma 4.3. Let $d = (Curve, l)$ be an Euler diagram with description D identified by f and g . Let c_1 and c_2 be curves in d . Then c_1 is contained (disjoint from) by c_2 if and only if $f(c_1)$ is contained by (disjoint from) $f(c_2)$.

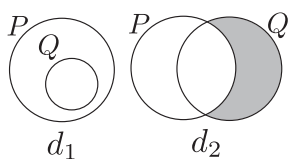


Fig. 13. Adding regions changes containment properties.

5. Adding and removing curves

Since the drawing method that we present is inductive, adding one curve at a time to the diagram, it is helpful to define two operations on diagrams: one for adding curves and another for removing curves.

Definition 5.1. Let $d = (Curve, l)$ be an Euler diagram and let $c \in Curve$. Then

$$d - c = (Curve - \{c\}, l - \{(c, l(c))\}).$$

Definition 5.2. Let $d = (Curve, l)$ be an Euler diagram. Let c be a curve that is not in $Curve$ and let $\lambda \in \mathcal{L}$. Then

$$d + (c, \lambda) = (Curve \cup \{c\}, l \cup \{(c, \lambda)\}).$$

We require analogous operations on descriptions. For example, taking the description $D = (C, B, l)$ where

1. $C = \{\kappa_P, \kappa_Q, \kappa_R\}$,
2. $B = \{\emptyset, \{\kappa_P\}, \{\kappa_P, \kappa_Q\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_R\}\}$ and
3. $l(\kappa_P) = P$, $l(\kappa_Q) = Q$ and $l(\kappa_R) = R$,

removing κ_R yields the description $D - \kappa_R = (C', B', l')$ where

1. $C' = \{\kappa_P, \kappa_Q\}$,
2. $B' = \{\emptyset, \{\kappa_P\}, \{\kappa_P, \kappa_Q\}, \{\kappa_Q\}\}$ and
3. $l'(\kappa_P) = P$, and $l'(\kappa_Q) = Q$.

See that, when forming B' , each occurrence of κ_R is removed from the abstract basic regions in B . The function l' is simply l with the domain restricted to C' , denoted $l|_{C'}$.

Definition 5.3. Given a description, $D = (C, B, l)$, and $\kappa \in C$, we define $D - \kappa$ to be

$$D - \kappa = (C - \{\kappa\}, B - \kappa, l|_{C - \{\kappa\}})$$

where

$$B - \kappa = \{\beta - \{\kappa\} : \beta \in B\}.$$

Removing a curve from a diagram corresponds to removing an abstract curve from its description:

Lemma 5.1. Let $d = (Curve, l_1)$ be a diagram with description $D = (C, B, l_2)$ identified by $f : Curve \rightarrow C$ and $g : BR(d) \rightarrow B$. Let c be a curve in d . Then $d - c$ has description

$$D - f(c) = (C - \{f(c)\}, B - f(c), l_2|_{C - \{f(c)\}}).$$

Proof (Sketch). An appropriate mapping between the curves of $d - c$ and the abstract curves of $D - f(c)$ is obtained by restricting the domain of f to $Curve - \{c\}$. \square

When adding an abstract curve to a description, we need to specify some information in order to be able to perform the addition. For example, if we wanted to add κ_R to $D - \kappa_R$ (from the previous example), in order to yield D (the original description), then we would need to specify that κ_R has label R . Moreover, we would need to ensure that the abstract basic regions in B' are altered correctly to give B . In particular, each abstract basic region in B' gives

rise to either one or two abstract basic regions in B : if it gives rise to one abstract basic region then that region is either ‘inside’ κ_R or ‘outside’ κ_R , whereas if it gives rise to two abstract basic regions then one of them is inside κ_R and the other is outside. In our running example, we have the following:

$$\emptyset \mapsto \emptyset \quad \{\kappa_P\} \mapsto \{\kappa_P, \{\kappa_P, \kappa_R\}\}$$

$$\{\kappa_P, \kappa_Q\} \mapsto \{\kappa_P, \kappa_Q\} \quad \{\kappa_Q\} \mapsto \{\kappa_Q, \kappa_R\}.$$

We can specify this addition of κ_R by saying that $\emptyset, \{\kappa_P\}$ and $\{\kappa_P, \kappa_Q\}$ each give rise to an abstract basic region outside κ_R and that $\{\kappa_P\}$ and $\{\kappa_Q\}$ each give rise to an abstract basic region inside κ_R . We write

1. $in = \{\emptyset, \{\kappa_P\}, \{\kappa_P, \kappa_Q\}\}$, and
2. $out = \{\{\kappa_P\}, \{\kappa_Q\}\}$.

In this example, we can see that

1. $in = \{\beta \in B - \kappa_R : \beta \cup \{\kappa_R\} \in B\}$, and
2. $out = \{\beta \in B - \kappa_R : \beta \in B\}$.

So, if we remove an abstract curve and then add it back in a manner that yields the original description, D , we can compute the sets in and out from the abstract basic regions in D . We will make use of this observation later.

Definition 5.4. Given $D = (C, B, l)$, let $\kappa \in C - C$, $\lambda \in \mathcal{L}$ and let $in \subseteq B$ and $out \subseteq B$ such that $\emptyset \in out$ and $in \cup out = B$. We define $D + (\kappa, \lambda, in, out)$ to be

$$D + (\kappa, \lambda, in, out) = (C \cup \{\kappa\}, out \cup (in + \kappa), l \cup \{(\kappa, \lambda)\}),$$

where $in + \kappa = \{\beta \cup \{\kappa\} : \beta \in in\}$.

Definition 5.5. Given $D = (C, B, l)$, and $\kappa \in C$, we define:

1. $in(\kappa, D) = \{\beta \in B - \kappa : \beta \cup \{\kappa\} \in B\}$ and
2. $out(\kappa, D) = \{\beta \in B - \kappa : \beta \in B\}$.

Lemma 5.2. Given $D = (C, B, l)$, and $\kappa \in C$,

$$(D - \kappa) + (\kappa, l(\kappa), in(\kappa, D), out(\kappa, D)) = D.$$

Proof. Set

$$D' = (D - \kappa) + (\kappa, l(\kappa), in(\kappa, D), out(\kappa, D)) = (C', B', l')$$

Trivially,

$$C' = (C - \{\kappa\}) \cup \{\kappa\}$$

and

$$l' = l|_{C - \{\kappa\}} \cup \{(\kappa, l(\kappa))\} = l.$$

All that remains is to show that $B' = B$. Given $out(\kappa, D)$ and $in(\kappa, D)$,

$$B' = out(\kappa, D) \cup (in(\kappa, D) + \kappa)$$

by definition. It is straightforward to show that

$$out(\kappa, D) = \{\beta \in B : \kappa \notin \beta\}$$

and

$$in(\kappa, D) + \kappa = \{\beta \in B : \kappa \in \beta\}.$$

Thus, $B' = out(\kappa, D) \cup (in(\kappa, D) + \kappa) = B$ as required. \square

Again, the notion of adding a curve to a diagram is consistent with the notion of adding an abstract curve to a description, captured by:

Lemma 5.3. Let $d = (Curve, l)$ be a diagram and let c be a curve that is not in $Curve$. Let $\lambda \in \mathcal{L}$. Let D be a description of $d + (c, \lambda)$ identified by f and g . Then $D - f(c)$ is a description of d .

Proof. The result follows by Lemma 5.1. \square

6. Clusters and piercings

Our diagram drawing method adds curves that pass through specified minimal regions. We want to ensure that we add only circles, so we cannot stipulate that the to-be-added curve passes through an arbitrary set of minimal regions: some ‘minimal region configurations’ do not admit the addition of a circle. However, if we have a set of minimal regions, \mathcal{MR} , that all meet at a point, p , and no other minimal regions meet at p , then we can add a circle, c , around p , passing through all and only the regions in \mathcal{MR} ; we call \mathcal{MR} a *cluster* and c a *piercing*, the idea of which was introduced, but not defined, in [24].

In Fig. 14, we can add a circle to the lefthand diagram, around p , passing through the four minimal regions around p ; such an addition yields the righthand diagram. We could not add a circle to the lefthand diagram that passes through precisely the minimal region inside just P and the minimal region inside just Q .

Definition 6.1. Let $d = (Curve, l)$ be an Euler diagram and let $\mathcal{MR} = \{mr_1, \dots, mr_k\}$ be a set of minimal regions in d . The set \mathcal{MR} is a *cluster* if there exists a point, p , in \mathbb{R}^2 and $\epsilon > 0$ such that for all $0 < \delta \leq \epsilon$, the δ -neighbourhood of p , $N_\delta(p)$, satisfies:

1. $N_\delta(p) - \bigcup_{c \in Curve} im(c) \subseteq mr_1 \cup \dots \cup mr_k$, and
2. for each $mr_i \in \mathcal{MR}$, $mr_i \cap N_\delta(p) \neq \emptyset$.

We say \mathcal{MR} is a cluster around p . Given such a δ -neighbourhood, $N_\delta(p)$, a circle, c , whose image is the boundary of $N_\delta(p)$ is a *piercing* for d around p .

So, a piercing curve is a circle and we can identify a cluster in order to see how to add such a circle.

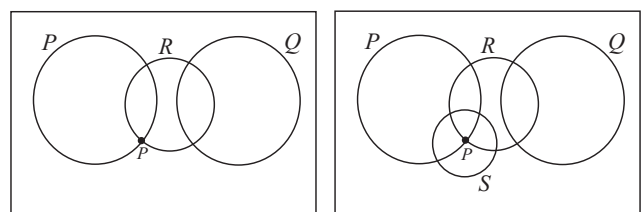


Fig. 14. Adding circles using clusters.

In diagrams produced using our drawing method, the so-called *inductively pierced diagrams*, the basic regions are connected, meaning that they include exactly one minimal region. Thus, for this class of diagrams, Definition 6.1 could be restated in terms of basic regions.

We now give a definition of a similar concept at the abstract level. Consider Fig. 14, where the point p is next to four basic regions that form a cluster in the lefthand diagram, d . A description of d includes three curves, κ_P labelled P , κ_Q labelled Q , and κ_R labelled R ; that is, we define $f(c_P) = \kappa_P$, $f(c_Q) = \kappa_Q$ and $f(c_R) = \kappa_R$ where c_λ is the curve in d labelled by λ . The basic regions in d around the point p therefore have descriptions

$$g(br_0) = \emptyset, \quad g(br_1) = \{\kappa_P\}, \quad g(br_2) = \{\kappa_R\}, \quad g(br_3) = \{\kappa_P, \kappa_R\}$$

where g is induced by the function f . These four abstract basic regions form a cluster and are each of the form $g(br_0) \cup K$ where $K \subseteq \{\kappa_P, \kappa_R\}$. In addition, since piercing curves split all zones through which they pass, we also see that $out(\kappa_R, D)$, where D is the description of d , is equal to $B(D - \kappa_R)$. This insight leads to:

Definition 6.2. Let β be an abstract basic region and let \mathcal{K} be a finite set of abstract curves disjoint from β , that is $\mathcal{K} \subseteq C - \beta$. The set $\{\beta \cup K : K \subseteq \mathcal{K}\}$ is a *cluster* for β , denoted $\mathcal{CL}(\beta, \mathcal{K})$. An abstract curve, κ , in an abstract description, D , is a *piercing* for $D - \kappa$ if $in(\kappa, D)$ is a cluster and $out(\kappa, D) = B(D - \kappa)$.

7. Inductively pierced diagrams and descriptions

We now define the class of inductively pierced Euler diagrams. They can be built inductively, adding one circle at a time. We make use of clusters in order to identify places in which circles can be added, as illustrated in Fig. 14.

Definition 7.1. A diagram, $d = (Curve, l)$, is *inductively pierced* if d is wellformed up to labelling and either d contains no curves, that is $Curve = \emptyset$, or d contains a curve, c , such that

1. c is a piercing for $d - c$, and
2. $d - c$ is inductively pierced.

Such a curve c is said to *identify* d as inductively pierced.

To illustrate the idea, the diagrams in Fig. 15 are all inductively pierced. The concept of being an inductively pierced diagram was introduced in [24], but was not defined and was restricted to the case where diagrams had injective labelling functions (no curve label could be used more than once).

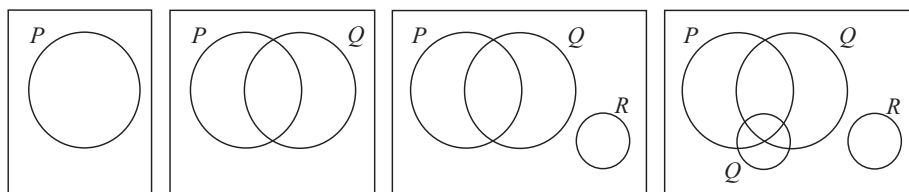


Fig. 15. Inductively pierced diagrams.

We now present a result that will be helpful to us later in the paper and provide some insight into restrictions that are placed on piercings in inductively pierced diagrams. In particular, we observe that in an inductively pierced diagram, the number of minimal regions and, therefore, basic regions (since basic regions are connected) inside a piercing curve is restricted to being 1, 2, or 4. The cases are illustrated in Fig. 16, with the dashed curve representing a δ -neighbourhood around p and each br_i is a basic region; the boundary of br_i is indicated by the irregular curve since this boundary may be formed from pieces of many circles. In the lefthand diagram, the point p does not lie on any curve, in which case it is inside a basic region, br_0 , so br_0 can be the only such region that satisfies condition 2 of Definition 6.1. If p lies on a single circle, c_1 , as shown in the middle diagram, then ‘small’ neighbourhoods around p can only contain points from the two basic regions, br_0 (outside c_1) and br_1 (inside c_1). Indeed, small neighbourhoods must include points from each of br_0 and br_1 . The only other option for p is that it lies on exactly two circles, since no more than two circles, c_1 and c_2 , pass through any point (no triple points). Moreover, since these two circles must cross at p (whenever two curves intersect they cross), p is next to exactly four basic regions. These arguments prove the following lemma:

Lemma 7.1. If d is identified as inductively pierced by c then, in d , c contains exactly 1, 2, or 4 basic regions.

We can make use of Lemma 7.1 to derive some properties of clusters in inductively pierced diagrams:

Lemma 7.2. Let $d = (Curve, l)$ be an inductively pierced diagram identified by curve c . Let \mathcal{MR} be the set of minimal regions in $d - c$ that contain points inside c in d . Let $D - c$ be the description of $d - c$ identified by f and g . Then

1. the set $\{g(\{mr\}) : mr \in \mathcal{MR}\}$ is a cluster in $D - c$
2. Specifically, the cluster in $D - c$ is

$$\{g(\{mr\}) : mr \in \mathcal{MR}\} = \mathcal{CL}(g(\{mr_s\}), \mathcal{K})$$

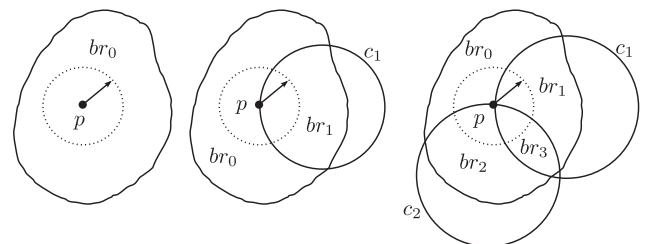


Fig. 16. Piercing curves around p .

where mr_s is the minimal region in \mathcal{MR} contained by fewest curves ($I(mr_s)$ is smallest), and

$$\mathcal{K} = \{f(c) : c \in I(mr_l) - I(mr_s)\}$$

where mr_l is the minimal region in \mathcal{MR} contained by most curves ($I(mr_l)$ is largest).

Proof. By Lemma 7.1, \mathcal{MR} contains 1, 2, or 4 basic regions. We make reference to Fig. 16.

1. In the case where \mathcal{MR} contains a single minimal (and, therefore, basic) region, namely br_0 , trivially we have $\{g(\{br_0\})\} = \mathcal{CL}(g(\{br_0\}), \emptyset)$.
2. In the case where \mathcal{MR} contains exactly two basic regions, we have $\{g(\{br_0\}), g(\{br_1\})\} = \mathcal{CL}(g(\{br_0\}), \{f(c_1)\})$.
3. In the case where \mathcal{MR} contains exactly four basic regions, we have $\{g(\{br_0\}), g(\{br_1\}), g(\{br_2\}), g(\{br_3\})\} = \mathcal{CL}(g(\{br_0\}), \{f(c_1), f(c_2)\})$.

In each case, we can readily show that \mathcal{K} is as required and we are done. \square

We now provide a definition of an inductively pierced description that precisely coincides with the notion of an inductively pierced diagram. Consider the obvious definition of an inductively pierced description: a description, D , is inductively pierced if it has an abstract curve, κ , that is a piercing for $D - \kappa$ and $D - \kappa$ is inductively pierced. Unfortunately, given this simple definition, there would be inductively pierced descriptions that do not have any drawing that is inductively pierced. For example, consider $D = (C, B, l)$ where

1. $C = \{\kappa_P, \kappa_Q, \kappa_R, \kappa_S, \kappa_T\}$,
2. $B = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}, \{\kappa_R\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}, \{\kappa_S\}, \{\kappa_P, \kappa_S\}, \{\kappa_Q, \kappa_S\}, \{\kappa_P, \kappa_Q, \kappa_S\}, \{\kappa_T\}, \{\kappa_P, \kappa_T\}, \{\kappa_Q, \kappa_T\}, \{\kappa_P, \kappa_Q, \kappa_T\}\}$,
3. $l(\kappa_P) = P$, $l(\kappa_Q) = Q$, $l(\kappa_R) = R$, $l(\kappa_S) = S$, and $l(\kappa_T) = T$.

The abstract curve κ_T is a piercing for $D - \kappa_T$, where $in(\kappa_T, D) = \mathcal{CL}(\emptyset, \{\kappa_P, \kappa_Q\})$.

In addition, κ_S and κ_R are also piercings with $in(\kappa_S, D) = in(\kappa_R, D) = \mathcal{CL}(\emptyset, \{\kappa_P, \kappa_Q\})$.

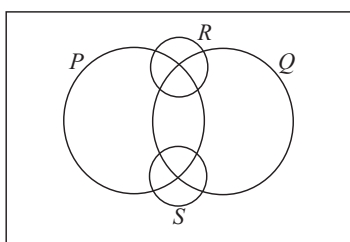


Fig. 17. Inductively pierced.

Removing κ_T from D yields a description, $D - \kappa_T$, that has an inductively pierced drawing, as shown in Fig. 17. It is not possible to add a piercing to this diagram in a manner that yields a diagram with description D : such a curve would need to pass through the basic regions corresponding to the abstract cluster $\mathcal{CL}(\emptyset, \{\kappa_P, \kappa_Q\})$. The issue here is that any pair of circles intersect at exactly two points. In order to add a curve labelled T in the required manner, it would necessarily need to enclose one of these points. In this case, one of those points is enclosed by R and the other is enclosed by S but we do not want the undrawn curve T to contain any points that are inside R or S .

In order to identify situations where this kind of issue arises, it is useful for us to define *pseudo-piercings*. Intuitively, in a diagram, a pseudo-piercing is a curve that would become a piercing if other curves were removed, but we need to define this concept for descriptions. For example, in Fig. 18, suppose that the curves map to the abstract curves κ_P , κ_Q , and so forth, in the obvious manner. We denote the description of this diagram by D . If we want to add a curve that pierces P and Q but is not to be drawn inside S or R then we cannot do so. However, in this case, the two points where P and Q intersect are not both enclosed by piercings (only one of them is and it is contained by κ_S). The abstract curve κ_T is not a piercing of κ_P and κ_Q since

$$in(\kappa_T, D) = \mathcal{CL}(\emptyset, \{\kappa_P, \kappa_Q\}) \cup \mathcal{CL}(\{\kappa_S\}, \{\kappa_P, \kappa_Q\})$$

and this is not a cluster. In the case of κ_R ,

$$in(\kappa_R, D) = \mathcal{CL}(\emptyset, \{\kappa_P, \kappa_Q\}) \cup \mathcal{CL}(\{\kappa_P, \kappa_U\}, \{\kappa_Q\})$$

which again is not a cluster. The abstract curves κ_R , κ_S and κ_T are all pseudo-piercings of κ_P and κ_Q . We can add a piercing of P and Q to this diagram provided it is contained by either R or both S and T . We would not be able to add a piercing if it was required to be contained by T but not by S , however. We say that S and R are *minimal pseudo-piercings* since they do not contain any pseudo-piercings; new piercings should be contained by minimal pseudo-piercings where they exist. The above examples have demonstrated that if we want to add a curve that pierces two other curves then whether that is possible relies on which curves are to contain it.

Definition 7.2. Let $D = (C, B, l)$ be a description and let κ , κ_1 , and κ_2 be distinct abstract curves in D . If there exists an abstract basic region, β , in B such that

1. $\kappa \in \beta$,
2. $\beta \cap \{\kappa_1, \kappa_2\} = \emptyset$,
3. $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\}) \subseteq B$, and
4. $\mathcal{CL}(\beta - \{\kappa\}, \{\kappa_1, \kappa_2\}) \subseteq B$

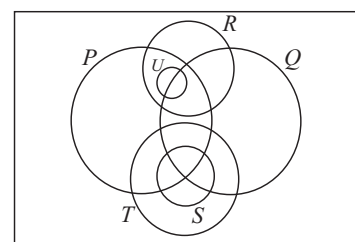


Fig. 18. Pseudo-piercings.

then κ is a pseudo-piercing of κ_1 and κ_2 . If such a κ does not contain any pseudo-piercings of κ_1 and κ_2 then κ is a minimal pseudo-piercing of κ_1 and κ_2 .

Given κ_1, κ_2 , and κ as in Definition 7.2, we can deduce that, if we remove all curves from D except κ_1 and κ_2 to yield description D' , the basic regions in D' would be $\emptyset, \{\kappa_1\}, \{\kappa_2\}$, and $\{\kappa_1, \kappa_2\}$; that is, in a drawing of D' , the two curves corresponding to κ_1 and κ_2 intersect. In particular, the following lemma establishes that κ is essentially a piercing of κ_1 and κ_2 :

Lemma 7.3. *Let $D = (C, B, l)$ be a description and let κ, κ_1 and κ_2 be distinct abstract curves in D . Suppose that κ is a pseudo-piercing of κ_1 and κ_2 in D . If we remove all abstract curves from D , except κ, κ_1 and κ_2 to give $D' = (C', B', l')$ then $B' = \{\emptyset, \{\kappa\}, \{\kappa_1\}, \{\kappa, \kappa_1\}, \{\kappa_2\}, \{\kappa, \kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa, \kappa_1, \kappa_2\}\}$.*

That is, D' is a description of a Venn diagram with three curves.

Proof. Consider

$$\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$$

and

$$\mathcal{CL}(\beta - \{\kappa\}, \{\kappa_1, \kappa_2\}),$$

as in Definition 7.2. On removing curves from D to create D' , these sets of abstract basic regions reduce to

$$\mathcal{CL}(\beta \cap \{\kappa, \kappa_1, \kappa_2\}, \{\kappa_1, \kappa_2\}) = \mathcal{CL}(\{\kappa\}, \{\kappa_1, \kappa_2\})$$

and

$$\mathcal{CL}((\beta - \{\kappa\}) \cap \{\kappa, \kappa_1, \kappa_2\}, \{\kappa_1, \kappa_2\}) = \mathcal{CL}(\emptyset, \{\kappa_1, \kappa_2\}).$$

Therefore

$$\mathcal{CL}(\{\kappa\}, \{\kappa_1, \kappa_2\}) \cup \mathcal{CL}(\emptyset, \{\kappa_1, \kappa_2\}) \subseteq B'.$$

Since

$$\mathcal{CL}(\{\kappa\}, \{\kappa_1, \kappa_2\}) \cup \mathcal{CL}(\emptyset, \{\kappa_1, \kappa_2\})$$

is equal to

$$\{\emptyset, \{\kappa\}, \{\kappa_1\}, \{\kappa, \kappa_1\}, \{\kappa_2\}, \{\kappa, \kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa, \kappa_1, \kappa_2\}\}$$

and these are all possible abstract basic regions that can be formed over $C' = \{\kappa, \kappa_1, \kappa_2\}$, we see that

$$B' = \{\emptyset, \{\kappa\}, \{\kappa_1\}, \{\kappa, \kappa_1\}, \{\kappa_2\}, \{\kappa, \kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa, \kappa_1, \kappa_2\}\}$$

thus completing the proof. \square

We can in fact make a stronger claim than Lemma 7.3:

Lemma 7.4. *Let $D = (C, B, l)$ be a description of an inductively pierced diagram, d , and let $\kappa, \kappa', \kappa_1$ and κ_2 be distinct abstract curves in D . Suppose that κ and κ' are minimal pseudo-piercings of κ_1 and κ_2 in D . If we remove all abstract curves from D , except $\kappa, \kappa', \kappa_1$ and κ_2 to give $D' = (C', B', l')$ then*

$$B' = \{\emptyset, \{\kappa\}, \{\kappa_1\}, \{\kappa, \kappa_1\}, \{\kappa_2\}, \{\kappa, \kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa, \kappa_1, \kappa_2\}\} \cup \{\emptyset, \{\kappa'\}, \{\kappa_1\}, \{\kappa', \kappa_1\}, \{\kappa_2\}, \{\kappa', \kappa_2\}, \{\kappa_1, \kappa_2\}, \{\kappa', \kappa_1, \kappa_2\}\}.$$

Lemma 7.4 is illustrated in Fig. 19. It should be relatively clear that, in an inductively pierced diagram, d , with description D , there cannot be more than two minimal pseudo-piercings of κ_1 and κ_2 .

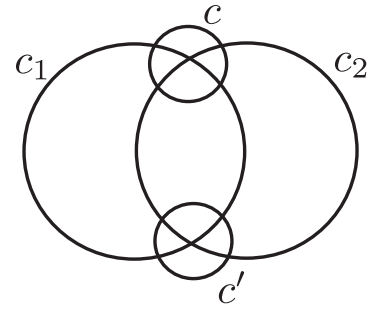


Fig. 19. Two minimal pseudo-piercings.

Definition 7.3. Let $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ be a cluster in description $D = (C, B, l)$. The cluster $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ is used if there exist two distinct minimal pseudo-piercings, κ_3 and κ_4 , of κ_1 and κ_2 such that $\kappa_3 \notin \beta$ and $\kappa_4 \notin \beta$. Otherwise $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ is available. In addition, a cluster in D containing 1 or 2 abstract basic regions is also available.

In Fig. 18, $\mathcal{CL}(\{\kappa_T\}, \{\kappa_P, \kappa_Q\})$ is used since κ_R and κ_S are both minimal pseudo-piercings of κ_P and κ_Q and neither are in the abstract basic region $\{\kappa_T\}$. However, the cluster $\mathcal{CL}(\{\kappa_S, \kappa_T\}, \{\kappa_P, \kappa_Q\})$ is available since only one of the minimal pseudo-piercings of κ_P and κ_Q , namely κ_R , is not in the abstract basic region $\{\kappa_S, \kappa_T\}$.

Lemma 7.5. *Let d be an inductively pierced diagram with a cluster, \mathcal{MR} , that contains four minimal regions. Let D be a description of d identified by f and g . Then the abstract cluster arising from \mathcal{MR} , namely,*

$$\mathcal{CL} = \{g(\{mr\}) : mr \in \mathcal{MR}\},$$

is available in D .

Proof. By Lemma 7.2, \mathcal{CL} is indeed a cluster in D , say $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ for some β, κ_1 and κ_2 ; so \mathcal{MR} is a cluster around a point where $f^{-1}(\kappa_1)$ and $f^{-1}(\kappa_2)$ intersect. Suppose, for a contradiction, that $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ is not available in D . Then it is used, so there exist two distinct minimal pseudo-piercings, κ and κ' , of κ_1 and κ_2 . By Lemma 7.4 in d we therefore have the configuration of curves as shown in Fig. 19, where $f^{-1}(\kappa) = c, f^{-1}(\kappa') = c', f^{-1}(\kappa_1) = c_1$ and $f^{-1}(\kappa_2) = c_2$. But then \mathcal{MR} is not a cluster, reaching a contradiction. Hence $\mathcal{CL} = \{g(\{mr\}) : mr \in \mathcal{MR}\}$ is available in D . \square

Lemma 7.6. *Let D be a description of an inductively pierced diagram, d , identified by f and g . If $\mathcal{CL}(\beta, \mathcal{K})$ is an available cluster in D then*

$$\mathcal{MR} = \{mr : \{mr\} = g^{-1}(\beta_i) \wedge \beta_i \in \mathcal{CL}(\beta, \mathcal{K})\},$$

is a cluster in d .

Proof (Sketch). Since available clusters contain 1, 2, or 4 abstract basic regions, the cardinality of \mathcal{K} is between 0 and 2. First, suppose that $\mathcal{K} = \emptyset$, so $\mathcal{CL}(\beta, \mathcal{K}) = \{\beta\}$. It is trivial that \mathcal{MR} is a cluster in this case.

Suppose instead that $\mathcal{K} = \{\kappa_1\}$. Then $\mathcal{CL}(\beta, \mathcal{K}) = \{\beta, \beta \cup \{\kappa_1\}\}$. It can be shown, by a relatively simple induction argument (over the number of curves in d), that $g^{-1}(\beta)$ and $g^{-1}(\beta \cup \{\kappa_1\})$ are topologically adjacent, separated by the circle $f^{-1}(\kappa_1)$. Hence \mathcal{MR} is a cluster in d .

Alternatively, $\mathcal{K} = \{\kappa_1, \kappa_2\}$, where κ_1 and κ_2 are two distinct curves. For this case, the strategy is to prove the contrapositive: if \mathcal{MR} is not a cluster in d then $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ is not an available cluster in D . In particular, if \mathcal{MR} is not a cluster in d then the minimal regions it contains are not joined at a point, p . In which case, since $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ contains four abstract basic regions (implying that \mathcal{MR} contains four minimal – essentially basic – regions) the two points at which $f^{-1}(\kappa_1)$ and $f^{-1}(\kappa_2)$ intersect are enclosed by curves whose image, under f , are not in β . Because d is inductively pierced, it can be shown, in particular, that there are two curves as in Fig. 19, whose abstractions are distinct minimal pseudo-piercings of κ_1 and κ_2 in D . But then $\mathcal{CL}(\beta, \{\kappa_1, \kappa_2\})$ would not be available. Hence, in all cases \mathcal{MR} is a cluster. \square

Definition 7.4. A description, $D = (C, B, l)$, is *inductively pierced* if either D contains no abstract curves, that is $C = \emptyset$, or D contains an abstract curve, κ , such that

1. κ is a piercing for $D - \kappa$,
2. $in(\kappa, D)$ is an available cluster in $D - \kappa$, and
3. $D - \kappa$ is inductively pierced.

We say that κ *identifies* D as inductively pierced.

As mentioned in Section 2, the definition we give of an inductively pierced description is more general than that in [24], which did not include any notion of pseudo-piercing or available clusters. Even when we consider only the case where the labelling function is injective (as had to be the case in [24]), our new approach is more general. To illustrate briefly, the inductively pierced diagram in Fig. 20 has a description that satisfies Definition 7.4 that would not be identified as inductively pierced in [24].

We will now establish that there is consistency between the notions of being an inductively pierced diagram and an inductively pierced description. As we have just seen there were inductively pierced diagrams that did not have inductively pierced descriptions under the less general definition given in [24]. Theorem 7.1 establishes that this is no longer the case: every inductively pierced diagram has an inductively pierced description. Theorem 7.2 extends a result in [24] which established that every inductively pierced description, under the less general definition given in that paper, can be drawn with circles.

Theorem 7.1. Let $d = (Curve, l)$ be an inductively pierced diagram with description $D = (C, B, l)$. Then D is inductively pierced.

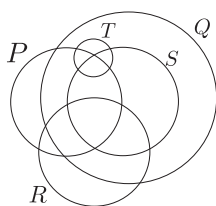


Fig. 20. An inductively pierced diagram whose description is not inductively pierced as defined in [24].

Proof. The proof proceeds by induction on the number of curves in d . For the base case, $|Curve| = 0$. Here, the only description of d is $D = (\emptyset, \{\emptyset\}, \emptyset)$, which is trivially inductively pierced. Now, suppose that for all inductively pierced diagrams, d , with $|Curve| = k$, the description, D , of d is inductively pierced. Consider a diagram, d , with $k + 1$ curves that is identified as inductively pierced by curve c . Then $d - c$ is inductively pierced, by definition. Now, given a description, D , of d , we know that $D - f(c)$ is a description of $d - c$, by Lemma 5.1. By assumption, $D - f(c)$ is inductively pierced. Moreover, the cluster, \mathcal{MR} , in $d - c$ that gave rise to the piercing, c , in d , is available in $D - f(c)$, by definition if \mathcal{MR} contains fewer than four basic regions, otherwise by Lemma 7.5. It is straightforward to show that $in(f(c), D)$ is the abstract cluster arising from \mathcal{MR} and that $out(f(c), D) = B(D - f(c))$. Hence, D is inductively pierced, as required. \square

Theorem 7.2. Let $D = (C, B, l)$ be an inductively pierced description. There exists an inductively pierced diagram, $d = (Curve, l)$, that is a drawing of D .

Proof. The proof also proceeds by induction. We just include the inductive step. Suppose $D = (C, B, l)$ is inductively pierced. Then $D - \kappa$ is inductively pierced, for some piercing κ for $D - \kappa$ that identifies D as inductively pierced. Moreover, $D - \kappa$ has an inductively pierced drawing, say $d' = (Curve, l)$, by assumption. We show that we can add a curve to d' in such a manner that it yields an inductively pierced drawing of D . Now, since D is inductively pierced, we know that $in(\kappa, D)$ is an available cluster in $D - \kappa$. By Lemma 7.6, the set of minimal regions, \mathcal{MR} , in d' that corresponds to $in(\kappa, D)$ is a cluster around some point, p . We can draw a piercing curve around p , to give a diagram, d , splitting each of the minimal regions in \mathcal{MR} . Since $out(\kappa, D) = B(D - \kappa)$, it is then straightforward to verify that d is an inductively pierced drawing of D , as required. \square

Hence, we have established that any inductively pierced description can be drawn with circles. Many descriptions are not inductively pierced; the focus of the next section is on converting descriptions into inductively pierced descriptions.

8. Transforming descriptions into inductively pierced descriptions

As identified in Fig. 5, we sometimes need to (a) add extra abstract basic regions to descriptions, and (b) split abstract curves into multiple abstract curves, to turn a description into an inductively pierced description. Given $D = (C, B, l)$, there is not necessarily a unique choice of abstract basic regions to add. One question that we must answer is: how do we identify a sufficient set of abstract basic regions to add? Trivially, we could just add ‘all possible’ basic regions given the abstract curve set (so, set $B = \{\beta : \beta \subseteq C\}$) but this is far from ideal.

To illustrate, consider the description $D = (C, B, l)$ where

1. $C = \{\kappa_P, \kappa_Q, \kappa_R, \kappa_S\}$,
2. $B = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_R\}, \{\kappa_P, \kappa_Q\}, \{\kappa_P, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_S\}, \{\kappa_Q, \kappa_S\}\}$, and
3. $l(\kappa_P) = P$, $l(\kappa_Q) = Q$, $l(\kappa_R) = R$ and $l(\kappa_S) = S$.

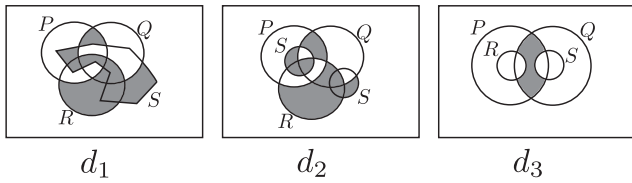


Fig. 21. Adding abstract basic regions.

This description is not inductively pierced. If we add all possible basic regions to D then we obtain a description that can be drawn as d_1 in Fig. 21. This is not inductively pierced, so we would also need to split curves in this case. One way of splitting the curves would yield d_2 . We argue that d_3 is a better diagram, which is a drawing of D with just a single additional abstract basic region (that which is shaded in d_3). In particular, d_3 is well-matched to its semantics [26], since

1. R is enclosed by P (the set R represents is a subset of that represented by P),
2. S is enclosed by Q (the set S represents is a subset of that represented by Q), and
3. R and S have disjoint interiors (R and S represent disjoint sets).

The inductively pierced diagram d_2 does not exhibit any of these three features. In addition, the number of shaded basic regions is considerably lower in d_3 (where shading is used to assert that the represented set is empty).

We want to be able to choose sensibly a set of abstract basic regions so, for example, if one abstract curve, κ_1 , is contained by another, κ_2 , in the original description, D , then κ_1 is still contained by κ_2 after we have added abstract basic regions. If we merely inspect a description it is not obvious how we identify such a set. Furthermore, the curves may need splitting up, but the manner in which any given abstract curve, κ , is split is, in part, determined by the abstract basic regions that include κ ; adding abstract basic regions may well enlarge the set of such regions that contain κ .

Now, for simplicity, suppose we have a description, D , which contains a curve, κ , where $D-\kappa$ is inductively pierced. In order to identify any abstract basic regions that we need to add to D , we only need to consider κ . To illustrate, if $D = (C, B, I)$ where

1. $C = \{\kappa_P, \kappa_Q, \kappa_R\}$
2. $B = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_R\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}$, and
3. $l(\kappa_P) = P$, $l(\kappa_Q) = Q$, and $l(\kappa_R) = R$.

Then $D-\kappa_R$ is inductively pierced but D is not inductively pierced. Given $D-\kappa_R$, in order to obtain D , we have $in(\kappa_R, D) = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}\}$

and

$$out(\kappa_R, D) = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}\}.$$

If κ_R was a piercing that identified D as inductively pierced then $out(\kappa_R, D)$ would equal $B(D-\kappa_R)$ (see

Definition 7.4. So, we add abstract basic regions to $out(\kappa_R, D)$ by setting $out(\kappa_R, D) = B(D-\kappa_R)$. We split κ_R into piercings, by essentially partitioning $in(\kappa_R, D)$ into available clusters, \mathcal{CL}_i , each containing 1, 2 or 4 abstract basic regions. In our example, $D-\kappa_R$ can be drawn as shown in the top of Fig. 22. There are many partitions of $in(\kappa_R, D)$ that give rise to a set of piercing curves that can be added to obtain an appropriate diagram. Three of these partitions, represented in the three diagrams in the bottom of Fig. 22, are:

1. $\{in(\kappa_R, D)\}$
2. $\{\{\kappa_P\}, \{\kappa_P, \kappa_Q\}\}, \{\emptyset, \{\kappa_Q\}\}$, and
3. $\{\{\kappa_P\}\}, \{\{\kappa_P, \kappa_Q\}\}, \{\emptyset, \{\kappa_Q\}\}$.

In general, we add abstract basic regions by setting $out(\kappa, D) = B(D-\kappa)$. This adds sufficient abstract basic regions so that we can split κ into piercings and ensure that the description becomes inductively pierced, provided $in(\kappa, D) \neq \emptyset$. If $in(\kappa, D) = \emptyset$ then we change it to any non-empty subset of $B(D-\kappa)$. We can also choose to add more abstract basic regions by arbitrarily enlarging $in(\kappa, D)$ (in our example above we kept $in(\kappa_R, D)$ fixed). Regardless of which abstract basic regions are in $in(\kappa, D)$, we can always find a partition into available clusters provided $in(\kappa, D)$ is non-empty, as we will demonstrate later.

Our approach breaks up the process of adding abstract basic regions and splitting abstract curves in to three stages:

1. Produce a *decomposition* of D . A decomposition is a sequence of descriptions, (D_n, \dots, D_0) where $D_n = D$, produced by removing abstract curves from D until there are no curves left.
2. Produce a *region-adding recomposition*. This takes a decomposition and adds abstract basic regions to produce a sequence of descriptions, (D'_0, \dots, D'_n) ; we precisely specify the regions that must be added, as well as allowing further regions to be added.
3. Produce a *piercing recomposition*. This takes a region-adding recomposition and turns it into a sequence of inductively pierced descriptions, thus splitting the abstract curves into piercings.

These three stages are presented in the next three subsections. Before presenting the details of each of these

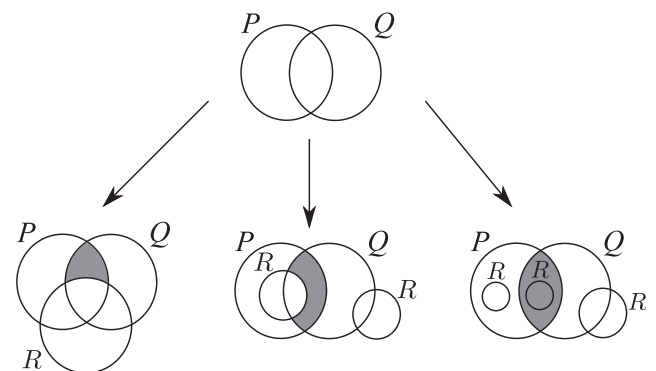


Fig. 22. Adding regions and splitting a curve.

stages, we provide a top-level algorithm that takes as input a description D and produces as output a drawing, d , of D . This is **Algorithm 1** and it calls further algorithms that will be detailed later in the paper.

Algorithm 1. Drawing Algorithm

Input: A description $D = (C, B, l)$.
Output: An Euler diagram d , which represents all interpretations that D represents.

1. Call **Algorithm 2**: Produce Decomposition, with
Input: The description $D = (C, B, l)$.
Output: A decomposition, $dec(D)$.
2. Call **Algorithm 3**: Produce Region-Adding Recomposition, with
Input: The decomposition, $dec(D)$.
Output: A region-adding recombination, $RArec(dec(D))$.
3. Call **Algorithm 5**: Produce Piercing Recomposition, with
Input: The region-adding recombination, $RArec(dec(D))$.
Output: A piercing recombination, $P(RArec(dec(D)))$.
4. Call **Algorithm 6**: Draw Diagram
Input: The piercing recombination, $P(RArec(dec(D)))$.
Output: An Euler diagram, d' .

Set $d = d'$ and terminate.

Three choices can be made during the transformation process, each of which can have a profound impact on the to-be-drawn diagram: the order of abstract curve removal when producing a decomposition, the set of added abstract basic regions when producing a region-adding recombination, and the choice of how to split up an abstract curve into a set of piercings when producing a piercing recombination. The choices are discussed in **Section 9**, where we show how to make them sensibly.

8.1. Decompositions

Producing a decomposition identifies an order in which to draw the curves, modulo any curve splitting that takes place later in the process. The curves are drawn in the opposite order to that in which their corresponding abstract curves are removed.

Definition 8.1. A decomposition of description D is a sequence, $dec(D) = (D_n, D_{n-1}, \dots, D_0)$, where

1. $D_n = D$,
2. $D_i = D_{i+1} - \kappa_{i+1}$, that is D_i is obtained from D_{i+1} by the removal of some abstract curve, κ_{i+1} , from D_{i+1} , and
3. D_0 contains no curves.

Algorithm 2. Produce Decomposition

Input: A description $D = (C, B, l)$
Output: A decomposition, $dec(D)$
 Initialization: Set $i = |C|$, $D_i = D$, and $dec(D_i) = (D_i)$.
while $i \geq 1$ **do**
 Choose $\kappa_i \in C(D_i)$, and set
 $D_{i-1} = D_i - \kappa_i$, $dec(D_{i-1}) = dec(D_i) \circ (D_{i-1})$
 where \circ denotes concatenation of sequences.
 Decrease i by 1.
Set $dec(D_i) = dec(D)$.

Later we will introduce a strategy for choosing decompositions. Thus, **Algorithm 2** which produces a decomposition can be refined to make sensible choices. As an

example, to which we will return when considering region-adding recompositions, consider $D = (C, B, l)$ where

1. $C = \{\kappa_P, \kappa_Q, \kappa_R\}$,
2. $B = \{\emptyset, \{\kappa_P, \kappa_R\}, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}\}$

with l defined in the obvious way. A drawing of D is d in **Fig. 23**, where R runs concurrently with the part of P that is not inside Q and the part of Q that is inside P . To produce a decomposition, one abstract curve removal order is $\kappa_R \rightarrow \kappa_Q \rightarrow \kappa_P$, which would result in the curve drawing order $\kappa_P \rightarrow \kappa_Q \rightarrow \kappa_R$. This would give decomposition

$$dec_1(D) = (D, D - \kappa_R, D - \kappa_R - \kappa_Q, D - \kappa_R - \kappa_Q - \kappa_P).$$

An alternative abstract curve removal order is $\kappa_Q \rightarrow \kappa_P \rightarrow \kappa_R$, which would result in the curve drawing order $\kappa_R \rightarrow \kappa_P \rightarrow \kappa_Q$. This would give decomposition

$$dec_2(D) = (D, D - \kappa_Q, D - \kappa_Q - \kappa_P, D - \kappa_Q - \kappa_P - \kappa_R).$$

The strategy we present, in for choosing decompositions will produce $dec_1(D)$ rather than $dec_2(D)$, for reasons that will become clear later; this is **Strategy 1**, given in **Section 9**.

8.2. Region-adding recompositions

Next, we take a decomposition and produce a recombination from it that includes, possibly, extra abstract basic regions.

Definition 8.2. Given a decomposition of description $D_n = (C, B, l)$, say $dec(D_n) = (D_n, D_{n-1}, \dots, D_0)$, the region-adding recombination of $dec(D_n)$ is a sequence of descriptions, denoted

$$RArec(dec(D_n)) = (RD_0, RD_1, \dots, RD_n)$$

such that

1. $RD_0 = D_0$, and
2. for each i where $0 < i \leq n$,
 $RD_{i+1} = RD_i + (\kappa_{i+1}, l(\kappa_{i+1}), in_{i+1}, out_{i+1})$

where

- (a) κ_{i+1} is the abstract curve removed from D_{i+1} to yield D_i in $dec(D_n)$,

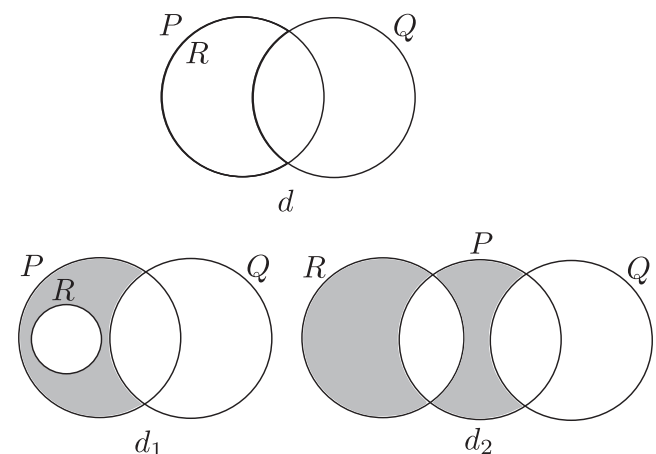


Fig. 23. Producing decompositions and region-adding recompositions.

- (b) if $in(\kappa_{i+1}, D_{i+1}) \neq \emptyset$ then
 $in_{i+1} = in(\kappa_{i+1}, D_{i+1}) \cup in_{a,i}$
for some $in_{a,i} \subseteq B(RD_i)$
- (c) if $in(\kappa_{i+1}, D_{i+1}) = \emptyset$ then
 $in_{i+1} = in_{b,i}$
for some non-empty $in_{b,i}$ where $in_{b,i} \subseteq B(RD_i)$.
- (d) $out_{i+1} = B(RD_i)$.

If each $in_{a,i}$ is empty and each $in_{b,i}$ is $\{\emptyset\}$ then we say that $RArec(dec(D_n))$ is a *minimal region-adding recomposition*.

Algorithm 3. Produce Region-Adding Recomposition

Input: A decomposition $dec(D_n) = (D_n, D_{n-1}, \dots, D_0)$ of a description, $D_n = (C, B, l)$.

Output: A region-adding recomposition, $RArec(dec(D))$.

Initialization: Set $i=0$, $RD_0 = D_0$, and $RArec_0 = (RD_0)$.

while $i < n$ **do**

if $in(\kappa_{i+1}, D_{i+1}) \neq \emptyset$ **then**

 choose any $in_{a,i} \subseteq B(RD_i)$ and set

$$in_{i+1} = in(\kappa_{i+1}, D_{i+1}) \cup in_{a,i}.$$

else

$in(\kappa_{i+1}, D_{i+1}) = \emptyset$ so choose any

 non – empty $in_{b,i} \subseteq B(RD_i)$ and set

$$in_{i+1} = in_{b,i}.$$

 Set $out_{i+1} = B(RD_i)$ and

$$RD_{i+1} = RD_i + (\kappa_{i+1}, l(\kappa_{i+1}), in_{i+1}, out_{i+1})$$

 where κ_{i+1} is the abstract curve removed from

D_{i+1} to yield D_i (in $dec(D_n)$). Also set

$$RArec_{i+1} = RArec_i \circ (RD_{i+1}).$$

 Increase i by 1.

To finish, set $RArec(dec(D_n)) = RArec_n$.

To produce a region-adding recomposition, follow [Algorithm 3](#). When producing a region-adding recomposition, there are two ways in which regions are added: by the enlargement of *in* and the enlargement of *out*. We have no choice about how to enlarge *out*, since we must ensure it is equal to $B(RD_i)$. For the set *in*, we can either leave it unchanged (provided it is non-empty) or add any abstract basic regions that we like from $B(RD_i)$.

Returning to the example given in [Section 8.1](#), producing a minimal region-adding recomposition of $dec_1(D)$, obtained from $D = (C, B, l)$, gives

$$RArec(dec_1(D)) = (D_1 - \kappa_R - \kappa_Q - \kappa_P, D_1 - \kappa_R - \kappa_Q, D_1 - \kappa_R, D_1),$$

where $D_1 = (C, B_1, l)$ with

$$B_1 = B(D) \cup \{\kappa_P\}.$$

Producing a minimal region-adding recomposition of $dec_2(D)$, obtained from $D = (C, B, l)$, gives

$$RArec(dec_2(D)) = (D_1 - \kappa_Q - \kappa_P - \kappa_R, D_2 - \kappa_Q - \kappa_P, D_2 - \kappa_Q, D_2),$$

where $D_2 = (C, B_2, l)$ with

$$B_2 = B(D) \cup \{\kappa_P, \kappa_R\}.$$

Drawings of D_1 and D_2 can be seen in [Fig. 23](#), as d_1 and d_2 respectively, where the added regions are shaded. We see that d_1 is well-matched to the semantics of d , since the curve labelled R is contained by the curve labelled P in both diagrams. By contrast, d_2 is not well-matched to the semantics of d since R is not contained by P in d_2 .

We will now establish that a region-adding recomposition does indeed result in a description, RD_n , that contains a superset of the abstract basic regions of D_n :

Lemma 8.1. Let D_n be a description with decomposition

$$dec(D_n) = (D_n, \dots, D_0)$$

and region-adding recomposition.

$$RArec(dec(D_n)) = (RD_0, \dots, RD_n).$$

Then $B(D_n) \subseteq B(RD_n)$.

Proof. We show that $B(D_{i+1}) \subseteq B(RD_{i+1})$, from which it follows that $B(D_n) \subseteq B(RD_n)$. We observe that $B(D_{i+1})$ is given by

$$B(D_{i+1}) = (in(\kappa_{i+1}, D_{i+1}) + \kappa_{i+1}) \cup out(\kappa_{i+1}, D_{i+1}), \quad (1)$$

where the abstract curve κ_{i+1} is that removed from D_{i+1} in $dec(D_n)$. Now

1. $in(\kappa_{i+1}, D_{i+1}) \subseteq in_{i+1} = in(\kappa_{i+1}, RD_{i+1})$, and
2. $out(\kappa_{i+1}, D_{i+1}) \subseteq out_{i+1} = out(\kappa_{i+1}, RD_{i+1})$.

Therefore, from (1) we have

$$B(D_{i+1}) \subseteq (in(\kappa_{i+1}, RD_{i+1}) + \kappa_{i+1}) \cup out(\kappa_{i+1}, RD_{i+1}) = B(RD_{i+1}).$$

Hence, $B(D_n) \subseteq B(RD_n)$. \square

Corollary 8.1. Let D_n be a description with decomposition $dec(D_n) = (D_n, \dots, D_0)$ and let (U, set) be an interpretation. If D_n represents (U, set) then RD_n represents (U, set) , where $RArec(dec(D)) = (RD_0, \dots, RD_n)$.

This means that if we draw RD_n instead of D_n then we have not reduced the information conveyed by the diagram. Of course, if RD_n contains more abstract basic regions then we may have enlarged the set of represented interpretations. However, we overcome this by shading any additional basic regions in the drawn diagram that are intended to represent the empty set.

8.3. Piercing recompositions

Our attention now turns to how we covert a region-adding recomposition into a piercing recomposition. For this purpose, it is helpful to introduce notation that denotes a basic region, β , with one abstract curve, κ_2 , substituted for another, κ_1 :

$$\beta[\kappa_2/\kappa_1] = \begin{cases} \beta & \text{if } \kappa_1 \notin \beta, \\ (\beta - \{\kappa_1\}) \cup \{\kappa_2\} & \text{otherwise.} \end{cases}$$

We read $\beta[\kappa_2/\kappa_1]$ as ‘ β with κ_2 for κ_1 ’. For example, if $\beta = \{\kappa_1, \kappa_2, \kappa_3\}$ then $\beta[\kappa_4/\kappa_2] = \{\kappa_1, \kappa_4, \kappa_3\}$. Generalizing this to a set of abstract basic regions, BR ,

$$BR[\kappa_2/\kappa_1] = \{\beta[\kappa_2/\kappa_1] : \beta \in BR\}.$$

To illustrate, if

$$BR = \{\{\kappa_1, \kappa_2\}, \{\kappa_1, \kappa_3\}\}$$

then

$$BR[\kappa_4/\kappa_2] = \{\{\kappa_1, \kappa_4\}, \{\kappa_1, \kappa_3\}\}.$$

Using this notation, given a set of abstract basic regions that contain some abstract curve, κ_1 , we can select a subset, BR , in which to replace κ_1 with κ_2 , which corresponds to ‘splitting an abstract curve into two abstract curves’. For example, in Fig. 22, we can split the curve labelled R in the bottom lefthand diagram into the two curves labelled R in the bottom middle diagram. The set of abstract basic regions, from the lefthand diagram, that include κ_R is

$$\{\{\kappa_R\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}$$

with the abstract curves corresponding to the diagram’s curves in the obvious manner. Choosing $BR = \{\{\kappa_P, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}$, and a new abstract curve $\kappa_{R'}$, we see

$$BR[\kappa_{R'}/\kappa_R] = \{\{\kappa_P, \kappa_{R'}\}, \{\kappa_P, \kappa_Q, \kappa_{R'}\}\}.$$

This describes the splitting of κ_R to give the two curves labelled R in the middle (bottom) diagram: in terms of the diagram on the left, the effect of the curve-split on the abstract basic region set, say $B(D)$, is to remove the elements of BR and add the elements of $BR[\kappa_{R'}/\kappa_R]$ to obtain the abstract basic region set, $B(D')$, for the diagram in the middle. So,

$$B(D') = (B(D) - BR) \cup BR[\kappa_{R'}/\kappa_R].$$

Definition 8.3. Let $D_1 = (C_1, B_1, l_1)$ be a description, let κ_1 be an abstract curve in D_1 and let κ_2 be a fresh abstract curve. Let BR be a proper, non-empty subset of $\{\beta \in B : \kappa_1 \in \beta\}$. Then the description obtained from D_1 by **splitting** κ_1 given BR and κ_2 , denoted

$$D_1[BR[\kappa_2/\kappa_1]/BR] = (C_2, B_2, l_2)$$

is

1. $C_2 = C_1 \cup \{\kappa_2\}$,
2. $B_2 = (B_1 - BR) \cup BR[\kappa_2/\kappa_1]$, and
3. $l_2 = l_1 \cup \{(\kappa_2, l_1(\kappa_1))\}$.

In terms of our drawing process, the construction of a region-adding recomposition ensures that each abstract curve, κ_1 , is inside some abstract basic region, β , that is $\kappa_1 \in \beta$; in a diagram, $f^{-1}(\kappa_1)$ is then a curve with a non-empty interior. When we perform splits, we use the set $in(\kappa_1, D)$ to split κ : we choose BR to be a subset of $in(\kappa_1, D)$. Requiring BR to be a proper subset of $\{\beta \in B : \kappa_1 \in \beta\}$ ensures that κ_1 is still inside some abstract basic region after the split occurs. Similarly, requiring BR to be non-empty ensures that the new curve is inside some abstract basic region. We observe that splitting abstract curves preserves semantics:

Lemma 8.2. Let $D = (C, B, l)$ be a description, let κ_1 be an abstract curve in D and let κ_2 be a fresh abstract curve. Let BR be a proper, non-empty subset of $\{\beta \in B : \kappa_1 \in \beta\}$. Let $I = (U, set)$ be an interpretation. Then D represents I if and only if $D[BR[\kappa_2/\kappa_1]/BR]$ represents I .

Proof. Let β be an abstract basic region in $D[BR[\kappa_2/\kappa_1]/BR]$. If β is also in D then, trivially,

$$A_1(\beta, D) = A_1(\beta, D[BR[\kappa_2/\kappa_1]/BR])$$

and

$$A_2(\beta, D) = A_2(\beta, D[BR[\kappa_2/\kappa_1]/BR])$$

from which it follows that

$$set(\beta, D) = set(\beta, D[BR[\kappa_2/\kappa_1]/BR]).$$

Alternatively, β is not in D so β is in $BR[\kappa_2/\kappa_1]$ and arose from substituting κ_2 for κ_1 in some abstract basic region, β' , in BR . In this case, since the label of κ_2 is the same as that of κ_1 , the parity of the number of occurrences of each label, λ , in the list of labels arising from the curves in β is the same as the parity in β' . Thus,

$$set(\beta) = set(\beta').$$

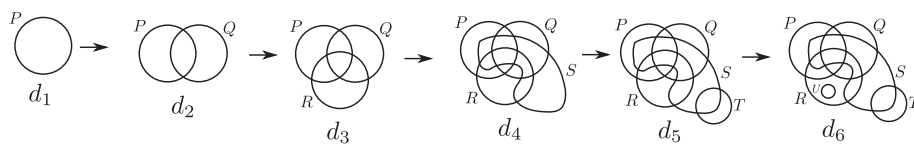


Fig. 24. A sequence of diagrams corresponding to a region-adding recomposition.

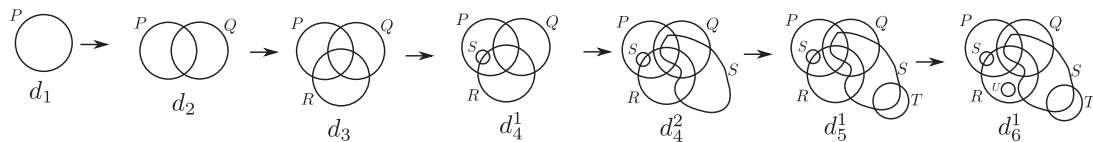


Fig. 25. Filtering a curve split along a sequence of diagrams.

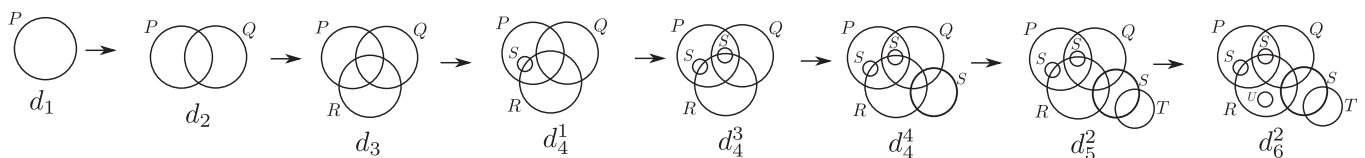


Fig. 26. A piercing recomposition.

It readily follows that

$$\bigcup_{\beta \in B(D[BR[\kappa_2/\kappa_1]/BR])} \text{set}(\beta, D) \subseteq \bigcup_{\beta \in B(D)} \text{set}(\beta, D).$$

A similar argument shows that

$$\bigcup_{\beta \in B(D)} \text{set}(\beta, D) \subseteq \bigcup_{\beta \in B(D[BR[\kappa_2/\kappa_1]/BR])} \text{set}(\beta, D[BR[\kappa_2/\kappa_1]/BR]).$$

Hence D represents I if and only if $D[BR[\kappa_2/\kappa_1]/BR]$ represents I . \square

Lemma 8.2 tells us that when we produce a piercing recomposition (which is derived from a region-adding recomposition by splitting abstract curves) we do not alter the semantics. The next stage in our process concerns how we split an abstract curve within the context of a region-adding recomposition. In particular, we observe that any abstract curve-splitting needs to filter along the recomposition, so it is reflected from the point at which the split takes place right up to the final description in the recomposition.

To illustrate the idea, we refer to Figs. 24 and 25. In particular, suppose that a region-adding recomposition comprises the sequence of descriptions of the diagrams in Fig. 24, namely

$$\text{Rec} = (D_0, D_1, D_2, D_3, D_4, D_5, D_6),$$

where D_0 is the description containing no abstract curves, and D_i is the description of d_i . Clearly, D_4 is not inductively pierced (this is the first such description in Rec). We can split κ_S (the abstract curve corresponding to S in d_4) into two abstract curves, namely κ_S and $\kappa_{S'}$, as shown in d_4^2 of Fig. 25. The diagram d_4^2 is that which we obtain from the split, and the diagram d_4^1 is inserted into the sequence to ensure we still have a recomposition. The effect of the split is also filtered all the way along the sequence, as shown in d_5^1 and d_6^1 .

We now proceed to establish how splitting an abstract curve into two abstract curves affects the next description in the recomposition. That is, given $D_1 = (C, B_1, I_1)$, where we have

- (a) added an abstract curve, κ , with label λ given sets in and out to yield

$$D_2 = D_1 + (\kappa, \lambda, in, out)$$

- (b) split an abstract curve, κ_1 , into κ_1 and κ_2 given some $BR \subseteq \{\beta \in B_1 : \kappa_1 \in \beta\}$ to yield

$$D_3 = D_1[BR[\kappa_2/\kappa_1], BR],$$

and we

- (c) identify sets in' and out' so that we can add κ to D_3 , respecting the addition of κ to D_1 , giving

$$D_4 = D_3 + (\kappa, \lambda, in', out'),$$

- (d) identify a set BR' so that we can split κ_1 into κ_1 and κ_2 in D_2 , respecting the split in D_1 , giving

$$D_4' = D_2[BR'[\kappa_2/\kappa_1]/BR].$$

We will show that $D_4 = D_4'$; the diagram in Fig. 27 illustrates the idea.

With reference to Figs. 24 and 25, considering the relationship between D_4 , D_5 , and D_4^2 (the description with κ_S split into κ_S and $\kappa_{S'}$), we have:

- (a) added κ_T to D_4 with $in = \{\emptyset, \{\kappa_S\}\}$ and $out = B(D_4)$ to yield D_5 , and
 (b) split the abstract curve κ_S in D_4 into κ_S and $\kappa_{S'}$ given $BR = \{\{\kappa_P, \kappa_S\}, \{\kappa_P, \kappa_R, \kappa_S\}\}$ to yield D_4^2 with the abstract curves corresponding to the curves in the obvious manner. Now, to obtain D_5^1 from D_4^2 , we add the abstract curve κ_T given some appropriate sets in' and out' . Equally, we can view D_5^1 as being obtained from D_5 by splitting κ_S given some appropriate set BR' . In particular, we have
 (c) sets $in' = \{\emptyset, \{\kappa_S\}\}$ and $out' = (B(D_4^2) - BR) \cup \{\{\kappa_P, \kappa_S\}, \{\kappa_P, \kappa_R, \kappa_S\}\}$ are used to add κ_T to D_4^2 giving

$$D_5^1 = D_4^2 + (\kappa_T, T, in', out'),$$
 and
 (d) the set $BR' = \{\{\kappa_P, \kappa_S\}, \{\kappa_P, \kappa_R, \kappa_S\}\}$ splits κ_S into κ_S and $\kappa_{S'}$ in D_5 to give

$$D_5^1 = D_5[BR'[\kappa_{S'}/\kappa_S]/BR].$$

In this example, $BR' = BR$ because the part of κ_S that we split off to form $\kappa_{S'}$ (the curve labelled S in d_4^1) does not intersect with, or contain any part of, κ_T .

Definition 8.4. Let in , out and BR be sets of abstract basic regions. Let κ , κ_1 and κ_2 be abstract curves. We define

- in' to be the set of abstract basic regions derived from BR and in given κ_1 and κ_2 where

$$in' = (in - BR) \cup \{\beta[\kappa_2/\kappa_1] : \beta \in in \cap BR\}.$$
- out' to be the set of abstract basic regions derived from BR and out given κ_1 and κ_2 where

$$out' = (out - BR) \cup \{\beta[\kappa_2/\kappa_1] : \beta \in out \cap BR\}.$$
- BR' to be the set of abstract basic regions derived from BR , in and out to be

$$BR' = \{\beta \cup \{\kappa\} : \beta \in BR \cap in\} \cup \{\beta : \beta \in BR \cap out\}.$$

Definition 8.5. Let $D_1 = (C_1, B_1, I_1)$ be a description. Let D_2 be a description obtained from D_1 by adding a fresh abstract curve, κ , given sets of abstract basic regions in and out , and label λ , so

$$D_2 = D_1 + (\kappa, \lambda, in, out).$$

Let $D_3 = D_1[BR[\kappa_2/\kappa_1]/BR] = (C_3, B_3, I_3)$ be a description obtained from D_1 by splitting κ_1 given BR and κ_2 , where κ_2 is fresh abstract curve.

- Then $D_4 = D_3 + (\kappa, \lambda, in', out')$ where
 - in' is the set of abstract basic regions derived from BR and in , and
 - out' is the set of abstract basic regions derived from BR and out

is the description that is obtained from D_3 by adding κ respecting the addition of κ to D_1 to obtain D_2 .

2. In addition, $D'_4 = D_2[BR'[\kappa_2/\kappa_1]/BR']$ where
 - (a) BR' is the set of abstract basic regions derived from BR , *in* and *out*

is the description that is obtained from D_2 by splitting κ_1 respecting the splitting of κ_1 in D_1 to obtain D_3 .

For the remainder of this section, we take D_4 and D'_4 to be as in Definition 8.5, given a suitable D_1 , D_2 and D_3 .

Lemma 8.3. *The descriptions D_4 and D'_4 are equal: $D_4 = D'_4$.*

Thus, from now on we will simply talk about D_4 and say that D_4 respects D_1 , D_2 and D_3 .

The following is required to establish that if we have a region-adding recomposition and we split an abstract curve, filtering through the effect of the split along the recomposition, then we have not changed the essential properties with respect to the *in* and *out* sets that describe the abstract curve additions.

Lemma 8.4. *Let D_4 be the description that respects D_1 , D_2 and D_3 . Then*

1. if $in \neq \emptyset$ then $in' \neq \emptyset$, and
2. if $out = BR(D_1)$ then $out' = BR(D_3)$.

In order to produce a piercing recomposition, we now capture the effect, on a recomposition, of splitting one of the abstract curves in some description, D_i . To ensure that we add only one abstract curve at a time, in the recomposition, the description D_i gives rise to two descriptions, one with κ_2 added, the other with κ_1 added (the split abstract curve plus a fresh abstract curve arising from the split). We have already seen this in Figs. 24 and 25: splitting κ_5 in d_4 gave d_4^1 and d_4^2 .

Definition 8.6. Let $R = (D_0, \dots, D_i, D_{i+1}, \dots, D_n)$ be recomposition such that κ_i is added to D_{i-1} to give D_i . Let κ be a fresh abstract curve. A *curve-splitting recomposition* obtained from R by splitting the curve κ_i in D_i is a sequence of descriptions

$$R' = (D_0, \dots, D_{i-1}, D'_i, D'_{i+1}, \dots, D'_n, D'_{n+1}),$$

such that

1. an abstract curve, κ_i , is split in D_i given some proper, non-empty subset, BR , of $in(\kappa_i, D_i)$ and κ , giving
 - (a) $D'_{i+1} = D_i[BR[\kappa/\kappa_i]/BR]$, and
 - (b) $D'_i = D'_{i+1} - \kappa_i$

and

2. for all k where $i+k+1 < n$, the description D'_{i+k+2} respects D_{i+k} , D_{i+k+1} and D'_{i+k+1} .

We say that R' is *piercing respecting* whenever BR is an available cluster in $D_i - \kappa_i$.

The sequence of descriptions associated with the diagrams in Fig. 25 is a piercing respecting, curve splitting recomposition given the recomposition associated with the diagrams in Fig. 24. We are only interested in

producing piercing respecting curve-splitting recompositions, and Algorithm 4 demonstrates how to create them.

Algorithm 4. Produce a Piercing Respecting Curve-Splitting Recomposition

Input: A recomposition

$$R = (D_0, D_1, \dots, D_{i-1}, D_i, \dots, D_n),$$

a curve, κ_i , in D_i that is not in D_{i-1} , such that $|in(\kappa_i, D_i)| \geq 2$, a proper non-empty subset, BR , of $in(\kappa_i, D_i)$ that is an available cluster in D_{i-1} , and a fresh abstract curve, κ .

Output: A piercing respecting, curve-splitting recomposition,

$$R' = (D_0, D_1, \dots, D_{i-1}, D'_i, D'_{i+1}, \dots, D'_n, D'_{n+1}).$$

Initialization: Set $j = i + 1$ and

$$R_j = (D_0, D_1, \dots, D_{i-1}, D'_i, D'_{i+1})$$

where $D'_{i+1} = D_i[BR[\kappa/\kappa_i]/BR]$ and $D'_i = D'_{i+1} - \kappa_i$.

while $j \leq n$ **do**

Set $R_{j+1} = R_j \circ (D'_{j+1})$ where D'_{j+1} respects D_{j-1} , D_j and D'_j .

Increment j by 1.

To finish, set $R' = R'_{n+1}$.

Given this notion, we can now produce a piercing recomposition. Intuitively, given a region-adding recomposition of D_n , say $RArec(dec(D_n)) = (D_0, \dots, D_n)$, if D_i is inductively pierced but D_{i+1} is not then we ‘split’ an abstract curve by selecting an available cluster, BR , with which to perform the split.

Definition 8.7. Let D_n be a description with decomposition $dec(D_n) = (D_n, \dots, D_0)$. A *piercing recomposition* for D_n is a recomposition, R , for which there exists a sequence, (R_0, \dots, R_m) of recompositions where

1. $R_0 = (D_{0,0}, D_{0,1}, \dots, D_{0,n})$ is a region-adding recomposition obtained from $dec(D_n)$,
2. for each i , where $1 \leq i \leq m$, R_i is a piercing respecting, curve splitting recomposition obtained from R_{i-1} , and
3. $R_m = R$ and contains only inductively pierced descriptions.

Continuing with the examples in Figs. 24 and 25, a piercing recomposition can be seen in Fig. 26. The following theorem is a key result of this section:

Theorem 8.1. *Let D_n be a description. There exists a piercing recomposition for D_n .*

Proof (Sketch). Given a decomposition, $dec(D_n)$, of D_n let $RArec(dec(D_n))$ be a region-adding recomposition. We start the process of splitting curves with the first non-inductively pierced description, RA_i , in $RArec(dec(D_n))$. We note that RA_i is obtained from RA_{i-1} by adding some abstract curve, κ_i , given some non-empty sets *in* and *out*, where $out = B(RA_{i-1})$. Now, we wish to split κ_i into κ_i and κ . Since $RArec(dec(D_n))$ is a region-adding recomposition, we know that $in(\kappa_i, RA_i)$ is not empty and $out(\kappa_i, RA_i) = B(RA_i - \kappa_i)$. Since κ_i is not a piercing, it follows that $in(\kappa_i, RA_i)$ contains at least two abstract basic regions. We choose $BR \subset in(\kappa_i, RA_i)$ to be an available cluster. Now, the first of the

newly inserted diagrams in the curve-splitting recomposition obtained by applying the split induced by BR is inductively pierced because $\mathcal{C}\mathcal{L}$ is available and out contains all abstract basic regions. Moreover, the act of performing this split and filtering it up the recomposition ensures that the essential properties of the in sets and out sets at each step in the sequence are maintained, that is, in' is non-empty and out' contains all abstract basic regions, from Lemma 8.4. We can, therefore, continue performing splits in this manner to give a sequence of curve-splitting recompositions until we have only inductively pierced descriptions. \square

Algorithm 5. Produce Piercing Recomposition.

Input: A region-adding recomposition, $RArec(dec(D)) = (RD_0, \dots, RD_n)$, formed from a decomposition $dec(D) = (D_n, \dots, D_0)$ of a description, $D = (C, B, l)$.

Output: a piercing recomposition, R .

Initialization: Set $i=0$ and set $S_j=RD_j$, for each $j \leq n$. Set $R_0 = (S_0, \dots, S_n)$ and $SeqR_0 = (R_0)$. Set $splitCount = 0$

while $S_{n+splitCount}$, in R_i , is not inductively pierced **do**

if S_{i+1} is obtained from S_i by adding κ_i with $in(\kappa_i, S_{i+1})$ being an available cluster in S_i **then**

 set

$R_{i+1} = R_i$ and $SeqR_{i+1} = SeqR_i$.

 and increment i by one.

else

S_{i+1} is obtained from S_i by adding κ_i where $in(\kappa_i, S_{i+1})$ is not an available cluster. In this case, $in(\kappa_i, S_{i+1})$

 must contain at least two abstract basic regions, since any singleton set is an available cluster.

 Choose a non-empty, proper subset, BR of $in(\kappa_i, S_{i+1})$ such that BR is an available cluster. Choose a fresh abstract curve κ .

 Call Algorithm 4, with, as input, the recomposition R_i , the abstract curve κ_i ,

 the available cluster BR and the fresh curve κ . Denote the output of Algorithm 4 by

$R_{i+1} = (S_0, \dots, S_i, S_{i+1}, \dots, S_{n+splitCount+1})$.

 Set $SeqR_{i+1} = SeqR_i \circ (R_{i+1})$.

 Increment i and $splitCount$ by 1.

To finish, we observe that $SeqR_i$ is a sequence of recompositions where the first sequence is $R_0 = RArec(dec(D))$

and each successive sequence is a piercing respecting recomposition obtained from the previous recomposition. Our termination condition for the While loop ensures that the final recomposition, R_i , in $SeqR_i$ is a piercing recomposition, as required. Thus, set $R = R_i$.

Algorithm 5 demonstrates how to produce a piercing recomposition from a region-adding recomposition. We have seen earlier that splitting abstract curves preserves semantics and that the act of filtering a split along a recomposition (for example, going from D_3 to D_4 in Fig. 27) preserves the semantics of the final description in the recomposition. Each abstract curve addition that occurs in the recomposition after the point at which the abstract curve split takes place corresponds to an abstract curve splitting operation applied to the corresponding description in the original sequence. We know, by Lemma 8.2, that splitting preserves semantics. In other words, the final description, P_m , in the piercing-recomposition represents the same interpretations as the final

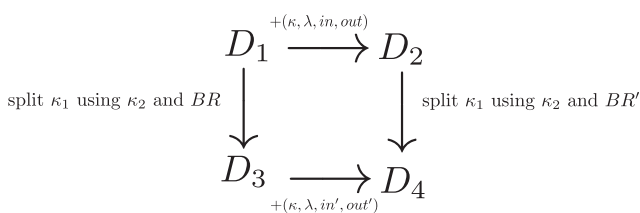


Fig. 27. Splitting and adding abstract curves.

description, RA_n , in the region-adding recomposition. By Corollary 8.1, we deduce that P_m represents all of the interpretations represented by the description, D_n , with which we started. This is captured by:

Theorem 8.2. Let D_n be a description and let P_m be the final description in some piercing recomposition obtained from D_n . Then any interpretation represented by D_n is also represented by P_m .

To conclude, we have demonstrated how to transform a description, D_n , into an inductively pierced description, P_m , in a manner that respects the semantics (Theorems 8.1 and 8.2). To draw P_m , we take the piercing recomposition and draw each circle one at a time, that is one circle for each curve addition in the recomposition, in the appropriate

manner; Section 10 sketches the algorithm used for this process and demonstrates our implementation of the drawing method. Section 9 gives details of the strategies we use to choose decompositions, region-adding recompositions, and piercing recomposition.

9. Choosing decompositions and recompositions

This section focuses on the choices we make when identifying a decomposition, converting the decomposition into a region-adding recomposition and then transforming this recomposition into a piercing recomposition. We devise various strategies that relate to well-matchedness, as described in Section 8, page 14. In particular, we will establish that our strategies preserve well-matched features of the description with which we start, relating to containment and disjointness.

9.1. Well-matchedness via contours

We wish to establish that the strategies we develop preserved the well-matchedness of the drawn diagram

with respect to the semantics of the original description. To illustrate, in Fig. 28, the curve labelled R is contained by the curve labelled P in d_1 but not in d_2 , so d_2 is not well-matched to the semantics of d_1 . To formally capture the well-matchedness of one description with respect to another description, we start by defining *contours*.

In a diagram, the set of curves with some given label, λ , is a *contour* with label λ . A contour tells us about the relationship of the set represented by λ with the other contours in the diagram. For instance, if a contour labelled P is contained by a contour labelled Q then we know that P is a subset of Q . A basic region, br , is *inside* a contour, CON with label λ if $\lambda \in A_1(br)$, otherwise br is *outside* CON . At the description level, the abstract curves in $D = (C, B, l)$ with some given label, λ , is the *abstract contour* in D with label λ . An abstract basic region, β , is *inside* an abstract contour, CON , with label λ if $\lambda \in A_1(\beta)$, otherwise β is *outside* CON .

Definition 9.1. Let $D = (C, B, l)$ be a description with abstract contours $CON_1 = \{\kappa \in C : l(\kappa) = \lambda_1\}$ and $CON_2 = \{\kappa \in C : l(\kappa) = \lambda_2\}$ for some $\lambda_1 \in im(l)$ and $\lambda_2 \in im(l)$. Then CON_1 contains CON_2 if

$$\{\beta \in B : \beta \text{ is inside } CON_2\} \subseteq \{\beta \in B : \beta \text{ is inside } CON_1\}.$$

If the above relationship is \subset then CON_1 properly contains CON_2 . In addition, CON_1 is disjoint from CON_2 if

$$\{\beta \in B : \beta \text{ is inside } CON_1\} \cap \{\beta \in B : \beta \text{ is inside } CON_2\} = \emptyset.$$

Again, contour containment and disjointness at the diagram level correspond to abstract contour containment and disjointness at the description level. Since the abstract curves of which an abstract contour comprises, between them, assert the relationship between the set denoted by that contour and the sets represented by the other contours in the diagram, the notion of being well-matched is encompassed by the contours. Clearly, if CON_1 contains CON_2 then $set(\lambda_1)$ is a superset of $set(\lambda_2)$ in any interpretation represented by D . Similarly, disjoint contours represent disjoint sets. Our strategies will ensure that contour containment and disjointness are preserved when transforming a description into an inductively pierced description.

Definition 9.2. Let $D_1 = (C_1, B_1, l_1)$ and $D_2 = (C_2, B_2, l_1)$ be descriptions that contain the same labels, that is

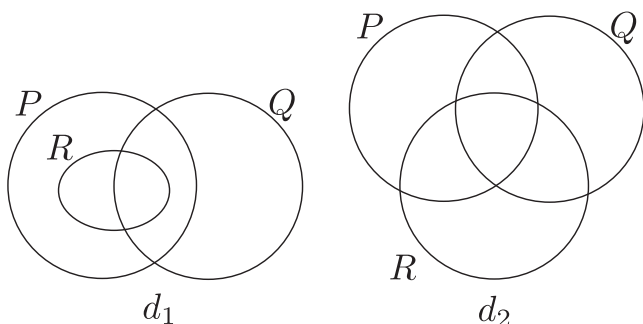


Fig. 28. Contour containment and well-matchedness.

$image(l_1) = image(l_2)$. Then description D_2 is *well-matched to the semantics of D_1* if, for all abstract contours $CON_{1,1}$ and $CON_{1,2}$ in D_1 , with labels $\lambda_{1,1}$ and $\lambda_{1,2}$ respectively,

1. if $CON_{1,1}$ properly contains $CON_{1,2}$ then $CON_{2,1}$ properly contains $CON_{2,2}$, and
2. if $CON_{1,1}$ is disjoint from $CON_{1,2}$ then $CON_{2,1}$ is disjoint from $CON_{2,2}$.

where $CON_{2,1}$ and $CON_{2,2}$ are the contours in D_2 with labels $\lambda_{1,1}$ and $\lambda_{1,2}$ respectively.

9.2. Choosing a decomposition

When producing a decomposition there are different choices about the order in which curves are removed. The ordering can profoundly impact on the description that we end up drawing. In addition to preserving contour containment and disjointness, our strategy ensures that if we start with an inductively pierced description then producing a minimal recomposition does not alter the description at all.

To illustrate the choices, consider the (not inductively pierced) diagram d in Fig. 29. Removing the abstract curves from the description, D , of d in the order $\kappa_R \rightarrow \kappa_Q \rightarrow \kappa_P$ and then producing a minimal recomposition yields a description of d_1 . Alternatively, removing the abstract curves from D in the order $\kappa_P \rightarrow \kappa_Q \rightarrow \kappa_R$ yields d_2 . Clearly, d_1 is a preferable diagram, given d , than d_2 , since the curve R is properly contained by P in both d and d_1 but not in d_2 . Consequently, d_2 is not well-matched with respect to the semantics of d , because d asserts that R is a subset of P . We observe that the order of curve removal that yielded d_1 respected curve containment (contained curves are removed earlier), whereas that which yielded d_2 did not.

In general, if we remove an abstract curve, κ_1 , before removing an abstract curve, κ_2 , contained by κ_1 then, when producing a region-adding recomposition, all of the abstract basic regions inside κ_2 will be split (i.e. added to *out*, so $in \subseteq out$) when adding κ_1 back. Clearly, this will not preserve curve containment: κ_1 will not contain κ_2 in the resulting recomposed description. In fact, removing only non-containing curves is sufficient to ensure that containment is preserved when producing a minimal region-adding recomposition. The situation with regard to curve disjointness is simpler, in that producing a minimal recomposition is sufficient to ensure that this holds, regardless of curve

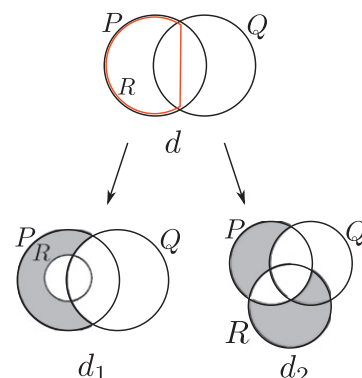


Fig. 29. Choices of curve removal order.

removal order. Finally, we note that when there is a choice of non-containing curve to be removed, we prioritize removing piercing curves. This ensures that if we start with an inductively pierced description then producing a minimal region-adding recomposition keeps the description unchanged (no regions are added).

Definition 9.3. Let $D = (C, B, l)$ be a description. We define the set of piercings in D to be

$$\mathcal{P}(D) = \{\kappa \in C : \kappa \text{ is a piercing for } D - \kappa\}.$$

We further define the set of abstract curves in D that do not properly contain any abstract curves in D to be

$$\mathcal{NC}(D) = \{\kappa_1 \in C : \forall \kappa_2 \in C - \kappa_2 \subset \kappa_1\}.$$

Obviously, a piercing cannot properly contain any other curves, so $\mathcal{P}(D) \subseteq \mathcal{NC}(D)$.

Strategy 1 (Decomposition Strategy). Let D_n be a description. A chosen decomposition, $dec(D_n) = (D_n, \dots, D_0)$, of D_n , is one where the following hold:

1. If $\mathcal{P}(D_i) \neq \emptyset$ then $D_{i-1} = D_i - \kappa_i$ for some $\kappa_i \in \mathcal{P}(D_i)$.
2. Otherwise $\mathcal{P}(D_i) = \emptyset$ and $D_{i-1} = D_i - \kappa_i$ for some $\kappa_i \in \mathcal{NC}(D)$.

Thus, in Algorithm 2, we use Strategy 1 when choosing abstract curve removal order. The process of producing a chosen decomposition is not deterministic, for instance there could be many piercing curves that could be removed at each step. In our implementation we must choose a specific order. In particular, we will refine this strategy in Section 9, to make an even more informed choice (where possible) given the approach we take to draw each circle.

9.3. Choosing a region-adding recomposition

Our recomposition strategy is very simple and ensures we add as few as possible abstract basic regions given the chosen decomposition:

Strategy 2 (Recomposition Strategy). Let D_n be a description. A chosen region-adding recomposition, $RArec(dec(D_n)) = (RD_0, \dots, RD_n)$, of D_n , is one where the following hold:

1. $dec(D_n)$ is a chosen decomposition, and
2. $RArec(dec(D_n)) = (RD_0, \dots, RD_n)$ is a minimal region-adding recomposition.

We use Strategy 2 in Algorithm 3. The following lemma trivially establishes that, once we have chosen a decomposition, there is no choice about how to produce a minimal (i.e. chosen) region-adding recomposition:

Lemma 9.1. Let D_n be a description. Let $dec(D_n)$ be a chosen decomposition of D_n . Then there exists a unique chosen region-adding recomposition, $RArec(dec(D_n)) = (RD_0, \dots, RD_n)$, obtained from $dec(D_n)$.

Thus, no refinements to the region-adding recomposition strategy are possible and this stage of the transformation process is deterministic.

Finally, we consider the relationships between the curves in D_n and those in RD_n in the context of well-matchedness. The following theorem establishes that RD_n is well-matched with respect to the semantics of D_n in terms of contour containment and disjointness:

Theorem 9.1. Let $D_n = (C, B, l)$ be a description and let $RArec(dec(D_n)) = (RD_0, \dots, RD_n)$ be a chosen recomposition. Then RD_n is well-matched to the semantics of D_n .

9.4. Choosing a piercing recomposition

The purpose of the piercing recomposition strategy is to minimize the number of curves that arise when splitting a curve into pieces. When determining how to split a curve, κ_i , we start by finding a valid partition of $in(\kappa_i, D_i)$:

Definition 9.4. Let $D = (C, B, l)$ be a description and let in be a subset of B . A partition, \mathcal{P} , of in is valid provided each set, p_i , in \mathcal{P} is an available cluster in D . A valid partition, \mathcal{P} , of in is minimal if there is no other valid partition of in containing fewer elements.

With reference to Figs. 24–26, given the curve S added to d_3 to give d_4 , in the chosen region-adding recomposition (shown in Fig. 24), we have

$$in(\kappa_S, D_4) = \{\emptyset, \{\kappa_P\}, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}, \{\kappa_R\}, \{\kappa_P, \kappa_R\}, \{\kappa_Q, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}.$$

A valid, but not minimal, partition of $in(\kappa, D_4)$ is

$$\{\{\kappa_P\}, \{\kappa_P, \kappa_R\}\}, \{\{\kappa_P, \kappa_Q\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}, \{\emptyset, \{\kappa_Q\}, \{\kappa_R\}, \{\kappa_Q, \kappa_R\}\}.$$

We can use elements of this partition to split κ_S into piercing curves. In particular, $BR = \{\{\kappa_P, \kappa_S\}, \{\kappa_P, \kappa_R, \kappa_S\}\}$ gives rise to the split as shown in Fig. 25. We observe that

$$in(\kappa_S, D_4^2) = \{\emptyset, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}, \{\kappa_R\}, \{\kappa_Q, \kappa_R\}, \{\kappa_P, \kappa_Q, \kappa_R\}\}.$$

That is,

$$in(\kappa_S, D_4^2) = in(\kappa_S, D_4) - (BR - \kappa_S).$$

Moreover, the given partition of $in(\kappa_S, D_4)$ corresponds to the splitting of κ_S as illustrated in Fig. 26. Here, the diagrams d_4^1 , d_4^3 and d_4^4 are those which arose from spitting κ_S .

This example illustrates that we can easily compute, in general, the effect of splitting κ_1 on $in(\kappa_1, D_i)$. In particular, given $in(\kappa_1, D_i)$ and set BR that we use to split κ_1 into κ_1 and κ_2 , the ‘new’ $in(\kappa_1, D_{i+1}')$ is equal to $in(\kappa_1, D_i) - (BR - \kappa_1)$. Thus, we can find a valid partition to identify a sequence of curve splits, breaking κ_1 up into piercings. Moreover, finding a minimal valid partition results in κ_1 being split into a minimal number of abstract curves, given the region-adding recomposition and the splits that have already occurred earlier in the process of obtaining a piercing recomposition. We use this insight to formulate our piercing recomposition strategy.

Strategy 3 (Piercing Recomposition Strategy). Let $D_n = (C, B, l)$ be a description. A piercing recomposition, $P = (P_1, \dots, P_q)$, of D_n , derived from a sequence of

recompositions, $(R_0, \dots, R_m = R)$, is chosen provided it arose by splitting curves using only minimal valid partitions.

Theorem 9.2. *Let $D_n = (C, B, l)$ be a description and let $P = (P_1, \dots, P_q)$ be a chosen piercing recombination. Then P_q is well-matched to the semantics of D_n .*

Proof. This follows immediately from Theorem 9.1, since contour containment and disjointness are invariant under curve splitting. \square

Finally, to conclude this section, we include some remarks on the number of splits that can occur. This is inherently linked to the number of abstract basic regions present in the description. In an inductively pierced description, D , containing at least two abstract curves, the largest number of abstract basic regions in D is captured by:

$$|B(D)| \leq 4|C(D)| - 4;$$

for the other cases, when $|C(D)| = 0$ we have $|B(D)| = 1$, and when $|C(D)| = 1$ we have $|B(D)| \leq 2$. When $|B(D)| = 4|C(D)| - 4$, there is a subset $B = \{\beta_1, \dots, \beta_n\}$ of $B(D)$ where

$$n = \frac{4|C(D)| - 4}{2} = 2|C(D)| - 2$$

for which

$$P = \{\{\beta_1\}, \dots, \{\beta_n\}\}$$

is a minimal valid partition. We demonstrate that a minimal, valid partition has cardinality at most n in Theorem 9.3.

Theorem 9.3. *Let $D = (C, B, l)$ be an inductively pierced description, containing at least two abstract curves. Let in be a non-empty subset of B . Then a minimal valid partition, P , of in satisfies*

$$|P| \leq 2|C(D)| - 2.$$

Proof. The proof is by induction on $|C(D)|$. For the base case, D has two abstract curves and it follows that $|B| \leq 4$. Trivially, any valid partition of a non-empty subset of B contains one element or two elements. We also see that $2|C(D)| - 2 = 2 \times 2 - 2 = 2$, so the result holds. Assume that if D contains $k \geq 2$ abstract curves then the result holds. Let $D' = (C', B', l')$ be an inductively pierced description containing $k + 1$ abstract curves, identified as inductively pierced by abstract curve κ' . Then, by assumption, the result holds for $D' - \kappa'$, that is for all non-empty subsets, in , of $B' - \kappa'$, any minimal valid partition, P , satisfies

$$|P| \leq 2|C(D' - \kappa)| - 2.$$

Let in' be a subset of B' . Then

$$in' \subseteq (B' - \kappa) \cup \mathcal{CL}(\beta', \mathcal{K}),$$

where

$$\mathcal{CL}(\beta, \mathcal{K}) = in(\kappa', D') + \kappa' = \{\beta' \cup \{\kappa'\} : \beta' \in in(\kappa', D')\}.$$

To create a valid partition of in' , start by splitting in' into two sets:

$$in'_{old} = in' \cap (B' - \kappa'),$$

$$in'_{new} = in' \cap (in(\kappa', D') + \kappa').$$

Since $in'_{old} \subseteq B' - \kappa'$, there is a minimal valid partition, P , of in'_{old} in the description $D' - \kappa'$. We show that P is a valid partition of in'_{old} in the description D' . Let $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ be a cluster in P , in which case $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ is available in $D' - \kappa'$. If $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ contains at most two abstract basic regions then, trivially, it is also available in D' . Otherwise, $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ contains exactly four abstract basic regions. Suppose, for a contradiction, that $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ is not available in D' . Then there exist two distinct minimal pseudo-piercings κ_3 and κ_4 , of κ_1 and κ_2 , where $\mathcal{K}_i = \{\kappa_1, \kappa_2\}$. Then it can be shown that either κ_3 and κ_4 are both minimal pseudo-piercings in $D' - \kappa'$ or that one of them is κ' . The former case cannot be so since this would imply $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ is not available in $D' - \kappa'$. In the latter case, $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ would contain an abstract basic region, namely $\beta_i \cup \{\kappa'\}$, contradicting the fact that $in'_{old} \subseteq B' - \kappa'$. Hence, $\mathcal{CL}(\beta_i, \mathcal{K}_i)$ is available in D' . Therefore, P is valid partition of in'_{old} .

Consider now the set in'_{new} , which is a subset of $in(\kappa', D') + \kappa$. If in'_{new} contains one, or two abstract basic regions then define

$$P' = P \cup \{\{\beta'\} : \beta' \in in'_{new}\};$$

here, P' has cardinality

$$|P| + |in'_{new}| \leq 2|C(D' - \kappa')| - 2 + 2 = 2|C(D')| - 2$$

by assumption. If in'_{new} contains four abstract basic regions then define

$$P' = P \cup \{in'_{new}\};$$

here, P' has cardinality

$$|P| + 1 \leq 2|C(D' - \kappa')| - 2 + 1 < 2|C(D')| - 2$$

by assumption. Otherwise, in'_{new} contains three abstract basic regions. Since

$$in'_{new} \subseteq \{\beta' \cup \{\kappa'\} : \beta' \in in(\kappa', D')\}$$

we can split it into two clusters, one containing one abstract basic region, say \mathcal{CL}_1 , and the other containing two abstract basic regions, say \mathcal{CL}_2 . We define

$$P' = P \cup \{\mathcal{CL}_1, \mathcal{CL}_2\};$$

here, P' has cardinality

$$|P| + 2 \leq 2|C(D' - \kappa')| - 2 + 2 = 2|C(D')| - 2$$

again by assumption. Hence, in all cases, we have constructed a partition, P' , of in' , where

$$|P'| \leq 2|C(D')| - 2.$$

It can readily be shown that P' is valid. Therefore, there exists a minimal valid partition of in' whose cardinality is at most $2|C(D')| - 2$, as required. \square

In terms of splitting, Theorem 9.3 tells us that abstract curves are split into at most $2|C(D - \kappa)| - 2$ curves when using the piercing recombination strategy. More precisely:

Corollary 9.1. *Let $D = (C, B, l)$ and let κ be an abstract curve in D such that $D - \kappa$ is an inductively pierced description.*

Then a minimal valid partition, P , of $in(\kappa, D)$ satisfies

$$|P| \leq 2|C(D - \kappa)| - 2.$$

10. Diagram drawing software

Here, we describe how to produce a drawing of an inductively pierced description, P_q , obtained from a description, D_n , via a chosen piercing recomposition. We will describe various layout choices that have been made. Moreover, we describe refinements to the strategies for producing chosen decompositions and piercing recompositions within the context of how our method draws the diagram. The method has been implemented and the software can be freely downloaded from www.eulerdiagrams.com/inductive_circles.html.

In the implementation, we make an assumption that the starting description, D , does not use curve labels more than once. To motivate this decision, we consider the information that Euler diagrams represent, which concerns sets and their relationships. In particular, the diagram should include zones for each required set intersection. For instance, to draw a diagram representing $P \subseteq Q$, we would want to represent the following set intersections:

1. the set intersection $\overline{P} \cap \overline{Q}$, corresponding to a basic region with $A_1 = \emptyset$,
2. the set intersection $Q \cap \overline{P}$, corresponding to a basic region with $A_1 = \{Q\}$, and
3. the set intersection $P \cap Q$, corresponding to a basic region with $A_1 = \{P, Q\}$.

Such a list of set intersections can be easily converted into a description, $D = (C, B, l)$, where the labelling function is injective:

1. $C = \{\kappa_P, \kappa_Q\}$,
2. $B = \{\emptyset, \{\kappa_Q\}, \{\kappa_P, \kappa_Q\}\}$, and
3. $l(\kappa_P) = P$ and $l(\kappa_Q) = Q$.

Since we are aiming to draw an Euler diagram to represent this kind of information, it is reasonable to make the assumption that the starting description, D , does not use curve labels more than once.

The implementation incorporates some refinements to the strategies outlined above and makes informed choices for circle centres and radii; these strategies and choices are discussed in the next three subsections. The last subsection demonstrates the associated software.

10.1. Extension of the decomposition strategy

As noted in Section 9, it is possible to extend the decomposition strategy, adding in extra constraints to determine a good order in which to remove the curves. In our implementation, we place additional constraints on the ordering of abstract curve removal when there is a choice.

First, if the decomposition strategy allows us to choose between removing κ_1 and κ_2 (so they are both piercings or they are both non-containing abstract curves) then we use a notion of abstract curve ‘size’ to, possibly, determine which to remove:

Definition 10.1. Let $D = (C, B, l)$ be a description and let κ_1 and κ_2 be abstract curves in D . We say that κ_1 and κ_2 are the *same size* in D if

$$|in(\kappa_1, D)| = |in(\kappa_2, D)|.$$

By removing abstract curves of the same size in sequence, we can make more informed choices about the relative positioning of their centres and sizes of their radii, allowing better layouts to be obtained; we describe our layout method in Section 10.3. Thus, we prioritize removing such abstract curves in sequence.

Second, if there is still a choice of abstract curves for removal, we prioritize removing them in order of the size of the set in : abstract curves with smaller in are removed first. The basis for this extension is derived from the intuition that smaller in sets are more likely to have smaller valid partitions. In addition, removing abstract curves either maintains or reduces the size of the in sets for the remaining curves, so making them more likely to have smaller valid partitions too. Clearly, this intuition has led to a heuristic method of choosing between curves and it is possible that more sophisticated approaches could be developed. However, comparing the sizes of the in sets is computationally simple and we do not want to implement overly complex algorithms to give better layouts at the expense of being able to draw diagrams in real times. In our implementation, we use the following extension of the decomposition strategy:

Strategy 4 (Decomposition Strategy: Extended). Let D_n be a description. If $|C(D_n)| = 0$ then $dec(D_n) = (D_n)$ is a chosen decomposition of D_n . If $|C(D_n)| > 0$ then $dec(D_n) = (D_n, \dots, D_0)$ is a *chosen* decomposition of D_n provided

1. if κ_n is a piercing removed from D_n to give D_{n-1} then all other piercings in D_n that are the same size as κ_n in D_n are removed before any other curves are removed,
2. if κ_n does not properly contain any abstract curve in D_n and is removed from D_n to give D_{n-1} then all other abstract curves in D_n that are the same size as κ_n in D_n are removed before any other curves are removed, and
3. given the first description, D_{n-i} , in $dec(D_n)$ that arose by the removal of an abstract curve not covered by the two cases above, the sequence (D_{n-i}, \dots, D_0) is selected by the implementation as a decomposition of D_{n-i} .

10.2. Adaptation of the piercing recomposition strategy

With regard to choosing a piercing recomposition, finding a minimal valid partition of a set $in(\kappa, D_i)$ is computationally expensive. A rather naïve approach would find all partitions, check whether they are valid, and then seek a minimal one from those that are valid.

Our approach, in the implementation, approximates this and does not guarantee to find *minimal* valid partitions. Whilst we cannot be certain that we find minimal partitions, [Theorem 9.2](#) still holds, since the proof follows from contour containment and disjointness being invariant under abstract curve splitting.

The approach uses the notion of *neighbouring* abstract basic regions:

Definition 10.2. Let β_1 and β_2 be abstract basic regions. They are *neighbours* if their symmetric difference contains exactly one element.

Thus, neighbouring abstract basic regions form a cluster. The implemented algorithm to find a valid partition of $in(\kappa, D)$ can be summarized as follows:

Strategy 5 (Find Valid Partition). Let $in(\kappa, D) = \{\beta_1, \dots, \beta_n\}$ be a set of abstract basic regions. Set $i=1$ and define $partition_0 = \emptyset$ and $remaining_0 = in(\kappa, D)$.

- (1) Search through $remaining_{i-1}$ for an available cluster, $\mathcal{C}\mathcal{L}_i$ containing exactly 4 abstract basic regions.
 - (a) If such a $\mathcal{C}\mathcal{L}_i$ exists, set

$$partition_i = partition_{i-1} \cup \{\mathcal{C}\mathcal{L}_i\},$$

$$remaining_i = remaining_{i-1} - \mathcal{C}\mathcal{L}_i,$$
 increment i by 1 and repeat step (1).
 - (b) Else proceed to step (2).
- (2) Search through $remaining_{i-1}$ for a cluster, $\mathcal{C}\mathcal{L}_i = \{\beta_1, \beta_2\}$ containing exactly 2 abstract basic regions where β_1 has fewest neighbours in $remaining_{i-1}$ and, out of all of β_1 's neighbours, β_2 has fewest neighbours in $remaining_{i-1}$.
 - (a) If such a $\mathcal{C}\mathcal{L}_i$ exists, set

$$partition_i = partition_{i-1} \cup \{\mathcal{C}\mathcal{L}_i\},$$

$$remaining_i = remaining_{i-1} - \mathcal{C}\mathcal{L}_i,$$
 increment i by 1 and repeat step (2).
 - (b) Else proceed to step (3).
- (3) At this point, there are no clusters containing exactly 4 or exactly 2 abstract basic regions. Set

$$validPartition = partition_{i-1} \cup \{\{\beta\} : \beta \in remaining_i\}.$$

In the worst case, $in(\kappa, D)$ has cardinality $2^{|C(D-\kappa)|}$, so this algorithm for computing valid partitions is computationally complex. The software that we have implemented includes with it the source code; those interested in knowing the full detail of this strategy for computing partitions can, therefore, inspect the code. We note that algorithms that are always capable of drawing an Euler diagram, given a description, are all computationally complex since the number of zones or abstract basic regions is exponential in the number of curves to be used.

10.3. Choosing circle centres and radii

The software uses the chosen piercing recomposition, $P = (P_1, \dots, P_q)$, and draws the diagram inductively, mirroring the abstract curve additions in P . A sketch of

the steps that the implementation follows for choosing circle centres and radii, given a piercing recomposition $P = (P_1, \dots, P_q)$, are given in [Algorithm 6](#). We have not presented [Algorithm 6](#) in much detail. Again, those interested in knowing the full detail of this algorithm can, therefore, inspect the code.

Algorithm 6. Produce a Drawing

Input: A piercing recomposition, $P = (P_1, \dots, P_q)$

Output: A drawing, d , of P_q .

1. For each required circle, choose a 'guide radius', determined by the number of basic regions inside the circle.
2. Make a list of 'build steps', one for each of the piercing recomposition steps.
3. For a sequence of build steps which add similar piercings with $|in| = 1$ (these piercings are to be drawn inside the same basic region), combine these steps together to apply as a single step.
4. For a sequence of build steps which add similar piercings with $|in| = 2$ (these piercings are to split the same two abstract basic regions), combine these steps together to apply as a single step.
5. Apply each build step in turn.
 - (a) For a collection of piercings with $|in| = 1$, choose a nice arrangement of disjoint circles inside the pre-existing basic region; for example, attempt to align the circles centers and choose similar sized radii.
 - (b) For a collection of piercings with $|in| = 2$, for each of the circles put the centre on the circle which separates the adjacent split basic regions, attempting to make each of the circles have a similar sized radius.
 - (c) For piercings with $|in| = 4$, put the centre of the new circle on an intersection point of the circles which the new circle is to pierce (take care because one of the intersection points may be unusable).

10.4. Using the software

To draw a diagram, users must enter the set intersections to be present in the form of a list. For example, to draw a diagram for the set intersection described by \emptyset , $\{P\}$ and $\{P, Q\}$, enter the list P PQ as shown in [Fig. 30](#); it is assumed that \emptyset is present and that all sets have single character names.

The user can choose between different decomposition and piercing recomposition approaches. In particular, to use the extended decomposition strategy, select *decompose using piercing curves first*. Another option allows decomposition in alphabetic order. When choosing a piercing recomposition, the options allow the use of the strategy presented here (called *recompose using double piercings*), the use of piercing curves that have only 1 or 2 abstract basic regions inside them (called *recompose using single pierings*), or the use of piercing curves that have only 1 abstract basic region inside them (called *recompose using zero piercing (nested)*). These different options allow the user to gain insight into the effect that different choices have on the layout of the diagram.

[Fig. 31](#) shows two different drawings of a diagram using a chosen decomposition (as defined above), with the lefthand diagram using 'recompose using zero piercing (nested)' and the righthand diagram using 'recompose using double piercings', i.e. a chosen piercing recomposition. Each of these diagrams represents the same set intersections as the diagrams in [Figs. 35 and 36](#), namely: a ac acd b ab abc abcd.

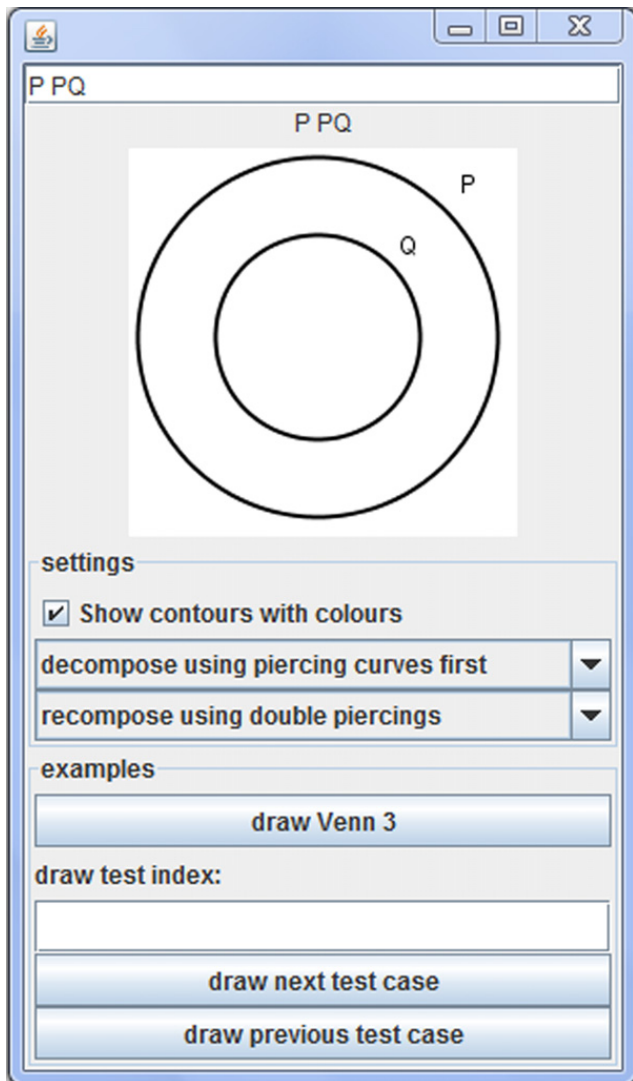


Fig. 30. Output from the software.

Fig. 32 shows all diagrams, up to isomorphism, for at least one and up to three sets drawn using chosen piercing recompositions. Fig. 33 shows a diagram representing 13 sets and Fig. 34 shows a diagram representing 24 sets. Here we can see that it is possible that the circles become small. The potential difficulty with reading diagrams with small circles could be eased by including scaling functionality. We note that the use of small circles does not arise because we are using circles: if one curve is to properly contain another curve then the latter necessarily has a smaller interior area, regardless of whether the curves are circles. Thus, this problem arises in Euler diagrams generally and is not a feature specific to our method.

11. Comparison with related work

This section complements the introduction by illustrating the range of drawing methods currently available, including some automatically generated images. These images should allow the reader to gain some insight into the existing state-of-the-art in terms of what it is currently possible to draw and the kinds of layouts achieved. The

drawing methods that currently exist for Euler diagrams can be broadly classified into three types: dual graph methods, inductive methods, and methods using particular geometric shapes. The method we present in this paper uses circles and is inductive.

11.1. Dual graph methods

With this class of methods, a *dual graph* of the required Euler diagram is identified from the zone descriptions and embedded in the plane [30]. Typically, but not always, the Euler diagram is then drawn by producing a dual of the dual graph, often called the Euler graph [28]. Methods in this class include the first Euler diagram drawing technique, attributable to Flower and Howse [16]. Others who have developed this class of drawing method include Verroust and Viaud [19], Chow [28], and Simonetto et al. [18]. A diagram drawn using the approach of [18] can be seen in Fig. 35, with the labels manually added post-drawing.² Rodgers et al. developed a general dual graph based method that is capable of drawing a diagram given any abstract description [29]; see Fig. 36 for a diagram drawn using this method which represents the same information as the diagram in Fig. 35. The examples in Figs. 35 and 36 typify the aesthetics of the diagrams produced by automated dual graph methods; it should be apparent that there is a need for improvement to the visual appearance. To allow the reader to draw comparison, a diagram with the same description drawn using our software can be seen in Fig. 37.

11.2. Inductive methods

With inductive methods, one curve of the required Euler diagram is drawn at a time, building up the diagram as one proceeds. The approach starts by breaking down the description, D , into a sequence of descriptions, $(D_0, D_1, \dots, D_n = D)$ say, where D_i describes an Euler diagram containing i curves. Given such a sequence, one identifies how to add a curve to the drawing of D_i in order to obtain D_{i+1} and, thus, eventually a drawing of D . This is a recently devised technique, attributable to Stapleton et al. [31], and builds on similar work for Venn diagrams [32,23]. An example of an automatically drawn diagram using the inductive method of [31] can be seen in Fig. 38, and a diagram with the same description, drawn using our method is in Fig. 39. As with the dual graph based methods, the diagram is lacking aesthetic quality. However, like some of the dual graph methods, Stapleton et al.'s inductive method is capable of drawing an appropriate diagram (i.e one that necessarily has the intended the semantics) given any description of the required zones. Moreover, it has advantages over the dual graph based methods in that it readily incorporates user preference for the well-formedness conditions that the to-be-drawn diagram is to satisfy.

² We thank Paolo Simonetto for supplying this image.



Fig. 31. Different piercing recompositions.

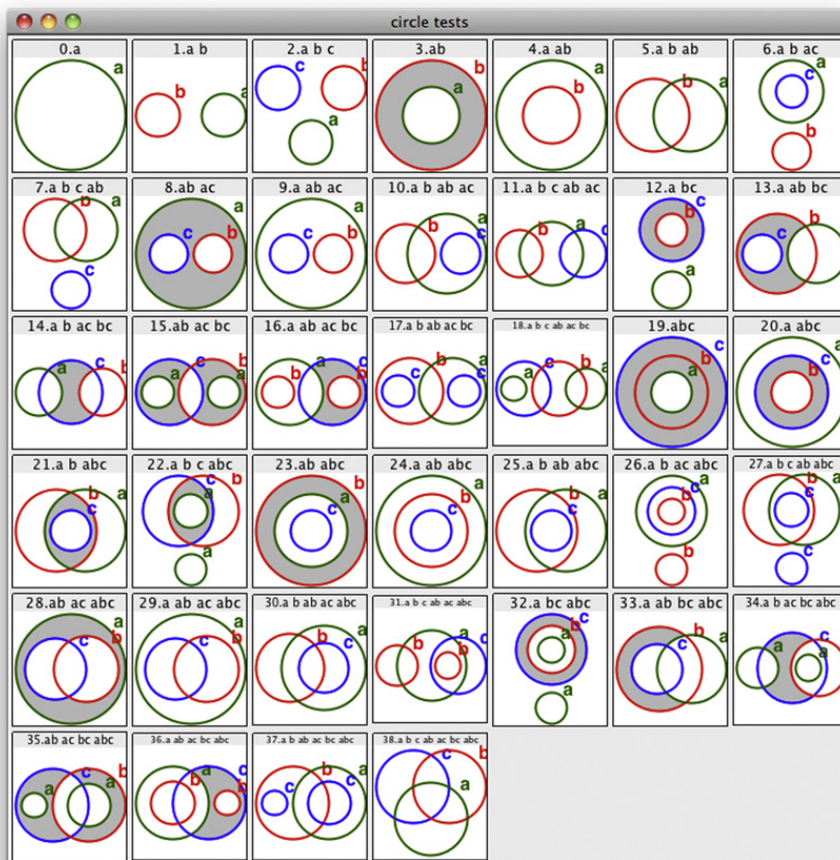


Fig. 32. Diagrams representing up to three sets.

11.3. 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. Many of these methods also draw so-called area-proportional Euler diagrams, where the areas of the zones must be particular values. Chow considers drawing diagrams with

exactly two circles [28], which is extended to three circles by Chow and Rodgers [33]. The Google Charts API includes facilities to draw Euler diagrams with up to three circles

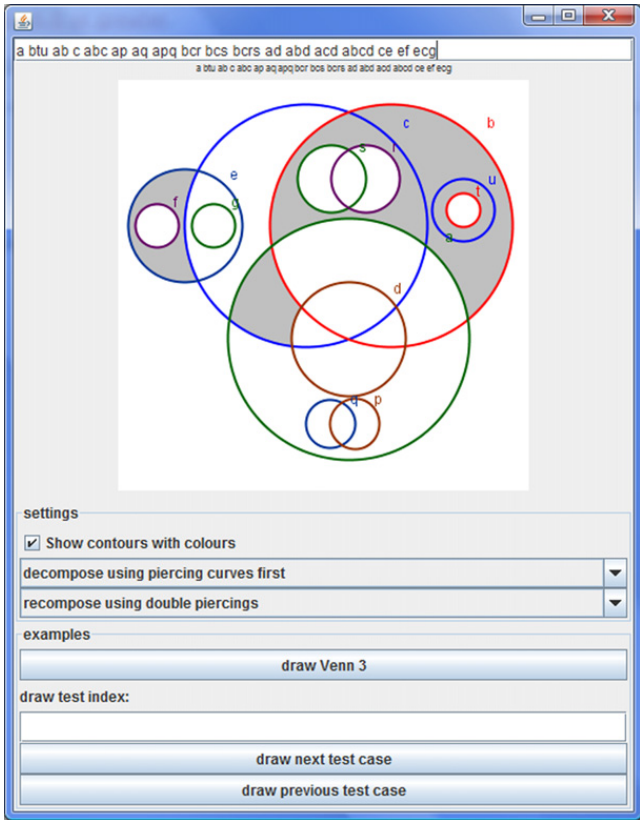


Fig. 33. A diagram representing 13 sets.

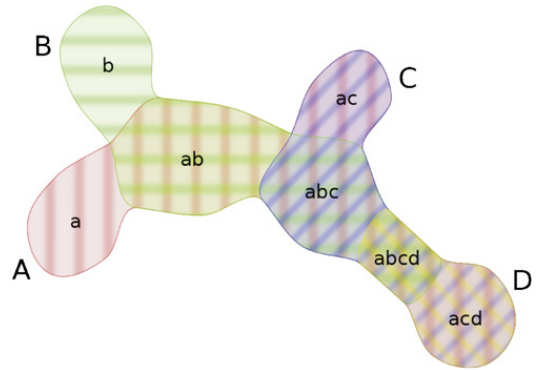


Fig. 35. An automatically drawn Euler diagram by Simonetto et al.'s method.



Fig. 36. Another automatically drawn Euler diagram by Rodgers et al.'s method.

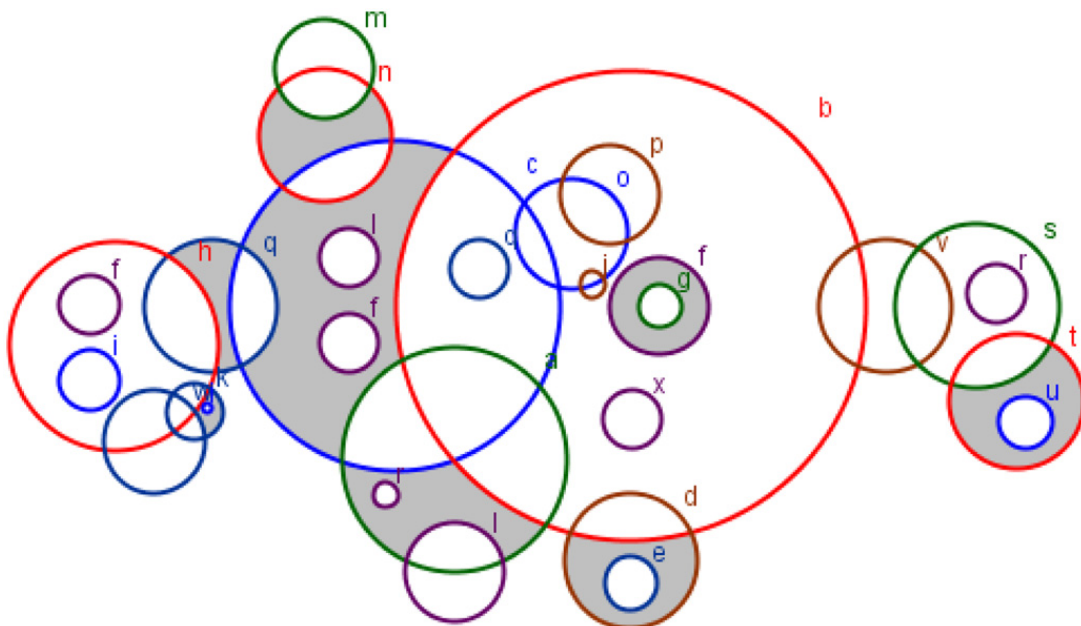


Fig. 34. A diagram representing 24 sets.

[34]. Kestler et al. devised a method, implemented in a tool called Vennmaster, that draws Euler diagrams with regular polygons but it does not guarantee that the diagrams have the required zones [35]; an example

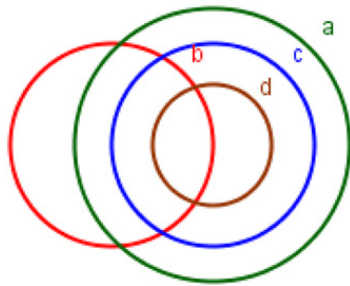


Fig. 37. The same information represented using our drawing method.

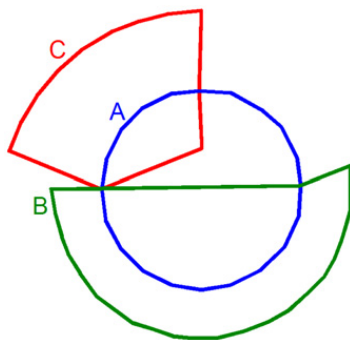


Fig. 38. An inductively drawn diagram using Stapleton et al.'s method.

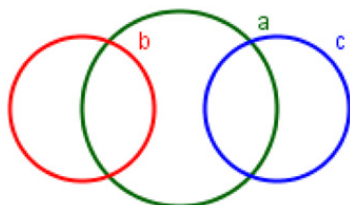


Fig. 39. The same information represented using our drawing method.

produced by Vennmaster can be seen in Fig. 40. The method devised by Wilkinson, which uses only circles, suffers from the same problem [20]. In previous work, we have devised a method for drawing a very restricted class of diagrams with circles, which does ensure the correct description is achieved [24]. None of these methods is capable of drawing an appropriate Euler diagram for some abstract descriptions. In part, this is because many abstract descriptions are not drawable with circles or regular polygons, given the constraints imposed by the various authors of this previous work on the properties that the diagrams are to possess (such as using exactly one curve to represent each set).

12. Conclusion

The primary contribution of this paper is that it is now possible to automatically draw an Euler diagram using only circles to represent any data set where the data are classified into categories. To achieve this, we have identified an equivalence between a class of Euler diagrams drawn with circles and a class of descriptions drawable with circles, i.e. the inductively pierced classes. Many descriptions are not inductively pierced and, indeed, not drawable with circles. Thus, we devised a method that can take any description and transform it into a description that is drawable with circles. During this transformation process many choices can be made which can have a significant impact on the qualities of the drawn diagram. We devised strategies that ensure the transformation process preserves contour containment and disjointness. The method has been implemented in a freely available software tool, available from www.eulerdiagrams.com/inductivecircles.html.

There are many directions for future work. First, the strategies we presented do not ensure that (a) a minimal number of abstract basic regions are added – for instance, any two chosen decompositions can produce chosen region-adding recompositions with different numbers of abstract basic regions, and (b) do not ensure that we have a minimal number of duplicated curve labels – for instance, choosing a non-minimal valid partition at one

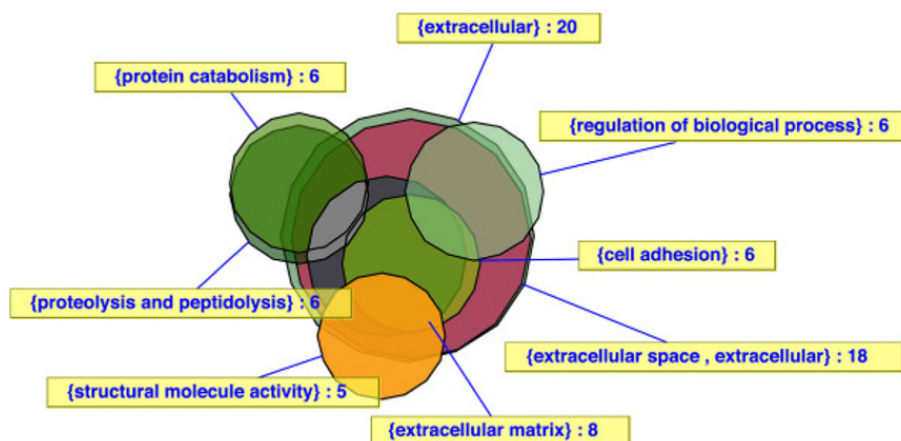


Fig. 40. Drawing with regular polygons.

step might enable a smaller minimal valid partition to be found at a later step. It could certainly lead to better diagram layouts, in some cases, if the strategies were extended or adapted with these deficiencies in mind. However, any extensions should also be mindful of the need to draw diagrams in real-time. There is often tension between finding an ‘optimal’ layout and computational efficiency.

Second, when considering well-matchedness, we have only considered pairwise comparisons: our strategy ensured that if a contour is contained by, or disjoint from, another contour then this is preserved under our transformations. With regard to disjointness, creating minimal region-adding recompositions ensured that disjointness properties were maintained. An extension is to ensure that if contour P is contained by the ‘union’ of contours Q_1 and Q_2 , as in Fig. 41, then this too is preserved by our transformation process. In general, we would want to ensure that if P is contained by the union of Q_1, \dots, Q_n then this is preserved by our transformation process.

Third, the layout choices we have made consider the alignment of circles with respect to those already drawn. This can mean that circles added later have a compromised layout. Seemingly sensible layout choices may, later in the drawing process, require modification to ensure a better final layout is found. At present, our method incorporates no post-processing step: once a circle is drawn, its center and radius are fixed. Layout improvement techniques, such as force directed approaches, may well be worth developing for these diagrams. In addition, other alterations could well lead to better diagrams. In particular it may be that adding more regions, when producing a region-adding recomposition, allows us to reduce the number of curves of which a contour consists. Our framework readily allows an extension of this kind, although the details of determining which regions to add in order to reduce the number of curves are likely to be challenging.

Finally, we have restricted curves to being circles because they are often used in manually drawn examples. This restriction was helpful in that we could characterize the inductively pierced diagrams at the abstract description level. One (potentially difficult) extension is to remove the restriction to inductively pierced diagrams, so considering all diagrams drawn with circles, and derive an abstract characterization of this more general class. A further obvious extension is to other geometric shapes, such as ellipses or rectangles, with which effective layouts can also be found. However, the task of finding a suitable characterization at the abstract level would be considerably more challenging. For example, when using ellipses there are more choices to be made about their layout: the size of the major axis, the size of the minor axis, the

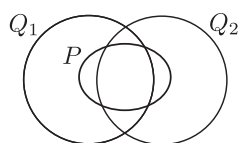


Fig. 41. Non-pairwise contour containment.

centre point, and the angle of rotation. In addition, in a well-formed diagram, any pair of ellipses can intersect at 0, 2, or 4 points. By contrast, circles intersect at 0 or 2 points, and we only have to determine their centres and radii. It is unclear whether the problem of classifying diagrams that are ‘inductively pierced with ellipses’ is readily solvable, although the work in this paper certainly provides a basis for this generalization.

Acknowledgements

This work is supported by the UK EPSRC Grants EP/H012311/1 and EP/H048480/1. Thanks to Peter Chapman for helpful comments on this paper.

References

- [1] J. Soriano, K. Davis, B. Coleman, G. Visick, D. Mannino, N. Pride, The proportional Venn diagram of obstructive lung disease, *Chest* 124.
- [2] G. Farrell, W. Sousa, Repeat victimization and hot spots: the overlap and its implication for crime control and problem-oriented policing, *Crime Prevention Studies* 12 (2001) 221–240.
- [3] R. DeChiara, U. Erra, V. Scarano, VennFS: A Venn diagram file manager, in: *Proceedings of Information Visualisation*, IEEE Computer Society, 2003, pp. 120–126.
- [4] J. Thiève, M. Viaud, A. Verroust-Blondet, Using Euler diagrams in traditional library environments, in: *Euler Diagrams 2004*, ENTCS, vol. 134, 2005, pp. 189–202.
- [5] E. Ip, Visualizing multiple regression, *Journal of Statistics Education* 9 (1).
- [6] H. Kestler, A. Muller, H. Liu, D. Kane, B. Zeeberg, J. Weinstein, Euler diagrams for visualizing annotated gene expression data, in: *Proceedings of Euler Diagrams 2005*, Paris, 2005.
- [7] N. Swoboda, G. Allwein, Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference, *Journal on Software and System Modeling* 3 (2) (2004) 136–149.
- [8] S.-J. Shin, *The Logical Status of Diagrams*, Cambridge University Press, 1994.
- [9] J. Howse, G. Stapleton, J. Taylor, Spider diagrams, *LMS Journal of Computation and Mathematics* 8 (2005) 145–194.
- [10] A. Delaney, G. Stapleton, Spider diagrams of order, in: *Visual Languages and Logic*, CEUR, vol. 274, 2007, pp. 27–39.
- [11] I. Oliver, J. Howse, G. Stapleton, E. Nuutila, S. Torma, Visualising and specifying ontologies using diagrammatic logics, in: *5th Australasian Ontologies Workshop*, CRPIT, vol. 112, 2009, pp. 87–104.
- [12] S. Kent, Constraint diagrams: visualizing invariants in object oriented modelling, in: *Proceedings of OOPSLA97*, ACM Press, 1997, pp. 327–341.
- [13] G. Stapleton, A. Delaney, Evaluating and generalizing constraint diagrams, *Journal of Visual Languages and Computing* 19 (4) (2008) 499–521.
- [14] G. Stapleton, J. Masthoff, J. Flower, A. Fish, J. Southern, Automated theorem proving in Euler diagrams systems, *Journal of Automated Reasoning* 39 (2007) 431–470.
- [15] S. Chow, F. Ruskey, Drawing area-proportional Venn and Euler diagrams, in: *Proceedings of Graph Drawing 2003*, Perugia, Italy, *Lecture Notes in Computer Science*, vol. 2912, Springer-Verlag, 2003, pp. 466–477.
- [16] J. Flower, J. Howse, Generating Euler diagrams, in: *2nd International Conference on the Theory and Application of Diagrams*, Springer, 2002, pp. 61–75.
- [17] N. Riche, T. Dwyer, Untangling Euler diagrams, *IEEE Transactions on Visualization and Computer Graphics* 16 (6) (2010) 1090–1099.
- [18] P. Simonetto, D. Auber, D. Archambault, Fully automatic visualisation of overlapping sets, *Computer Graphics Forum* 28 (3) (2009) 967–974.
- [19] A. Verroust, M.-L. Viaud, Ensuring the drawability of Euler diagrams for up to eight sets, in: *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, *Lecture Notes in Artificial Intelligence*, vol. 2980, Springer, 2004, pp. 128–141.
- [20] L. Wilkinson, Exact and approximate area-proportional circular Venn and Euler diagrams, *IEEE Transactions on Visualization and Computer Graphics* 18 (2) (2012) 321–331.

- [21] L. Euler, Lettres a une princesse d'Allemagne sur divers sujets de physique et de philosophie, *Letters 2* (1775) 102–108. berne, Socit Typographique.
- [22] F. Ruskey, A survey of Venn diagrams, *Electronic Journal of Combinatorics* <www.combinatorics.org/Surveys/ds5/VennEJC.html>.
- [23] J. Venn, On the diagrammatic and mechanical representation of propositions and reasonings, *Philosophical Magazine*.
- [24] G. Stapleton, L. Zhang, J. Howse, P. Rodgers, Drawing Euler diagrams with circles: the theory of piercings, *IEEE Transactions on Visualization and Computer Graphics*.
- [25] G. Stapleton, L. Zhang, J. Howse, P. Rodgers, Drawing Euler diagrams with circles, in: *6th International Conference on the Theory and Application of Diagrams*, Springer, 2010, pp. 23–38.
- [26] C. Gurr, K. Turlas, Towards the principled design of software engineering diagrams, in: *Proceedings of 22nd International Conference on Software Engineering*, ACM Press, 2000, pp. 509–518.
- [27] G. Stapleton, P. Rodgers, J. Howse, J. Taylor, Properties of Euler diagrams, in: *Proceedings of Layout of Software Engineering Diagrams*, EASST, 2007, pp. 2–16.
- [28] S. Chow, Generating and drawing area-proportional Euler and Venn diagrams, Ph.D. Thesis, University of Victoria, 2007.
- [29] P. Rodgers, L. Zhang, A. Fish, General Euler diagram generation, in: *5th International Conference on the Theory and Application of Diagrams*, Springer, 2008, pp. 13–27.
- [30] G. Stapleton, J. Howse, P. Rodgers, A graph theoretic approach to general Euler diagram drawing, *Theoretical Computer Science* 411 (1) (2010) 91–112.
- [31] G. Stapleton, P. Rodgers, J. Howse, L. Zhang, Inductively generating Euler diagrams, *IEEE Transactions of Visualization and Computer Graphics* 17 (1) (2011) 88–100.
- [32] A.W. Edwards, Venn diagrams for many sets, *New Scientist* 7 (1989) 51–56.
- [33] S. Chow, P. Rodgers, Constructing area-proportional Venn and Euler diagrams with three circles, in: *Proceedings of Euler Diagrams 2005*, 2005.
- [34] Google charts API, <<http://code.google.com/apis/chart/>> (accessed August 2009).
- [35] H. Kestler, A. Muller, J. Kraus, M. Buchholz, T. Gress, H. L. abd D. Kane, B. Zeeberg, J. Weinstein, Vennmaster: Area-proportional Euler diagrams for functional go analysis of microarrays, *BMC Bioinformatics* 9 (67).