



Abstractions of Euler Diagrams

Andrew Fish^{1,2} Jean Flower¹

*Visual Modelling Group
University of Brighton
Brighton BN2 4GJ, UK*

Abstract

Euler diagrams are a graphical means to represent information. Providing an abstraction captures the pertinent information precisely, ignoring irrelevant details (where relevancy is dependent upon the application domain). We present two new abstractions and show that these are equivalent to a standard existing (zone-based) representation. Examples illustrate the potential usefulness of different abstractions in various areas, such as: identifying properties like nestedness or drawability, expressing the semantics in a more readable manner, and in the layout of diagrams.

Keywords: Euler diagrams, Abstractions, Semantics, Layout.

1 Introduction

Euler diagrams have been used for many years and appear in applications in many areas, such as data storage and representation [1,2]. In order to formalize Euler diagrams it is useful to give them an *abstract syntax* which abstracts the essential diagram properties from non-essential *concrete syntax* details. The distinction between essential and non-essential depends upon the intended use of the diagram. A suitably chosen abstract syntax enables discussion of some diagram properties (such as nestedness, semantics, inference) without having to concern ourselves with irrelevant geometric and topological details (such as thickness of lines or the shape of contours).

¹ This research was partially supported by UK EPSRC grant GR/R63516

² Email: Andrew.Fish@brighton.ac.uk

A concrete Euler diagram is a certain type of drawing in the plane. Given such a drawing, one can capture its essential properties in various ways. One such abstract syntax which has been widely used encompasses information about which *zones* [8] (or *minimal regions* [7,13,15]) are present in the diagram. In this paper we suggest alternative abstract syntax and consider their advantages and drawbacks in the areas of diagrammatic properties (such as nestedness and drawability), semantic interpretation of diagrams, and diagram layout.

The first new abstract syntax involves a pair of regions for each contour, which describe its placing relative to the other contours in the diagram (called *containment conditions*). The second new abstraction uses a sequence of contours and pairs of regions to describe an incremental development: a *building sequence* for the final diagram.

Section 2 contains a description of concrete Euler diagrams and the basic algebraic framework required to discuss the abstractions. The abstractions are introduced in section 3.

In section 4 we present a collection of simple examples, deriving the abstractions from concrete diagrams, and indicate potential for identifying properties of diagrams (such as symmetries or nestedness) from these abstract descriptions alone.

Then, in section 5, we show that these abstractions are equivalent, in the sense that given any one of these abstract descriptions of a diagram one can construct any other abstract description. We do this by providing mappings between the different abstractions and showing that mapping from one abstraction to another and back again is the identity.

The semantics of Euler diagrams given by these differing abstractions are given in section 6. The different statements for a diagram (corresponding to the different abstractions) will be logically equivalent but will differ in appearance. They can be used to judge the readability of the semantics, using the different abstractions.

Given an abstract diagram one can also try to draw a concrete diagram which represents it. The difficulties associated with doing this will depend upon the abstract syntax chosen (the properties used to describe the relevant information) as well as on the well-formedness conditions imposed (conditions such as “no concurrent lines”, which such a drawing must satisfy). Furthermore, one may wish to draw a diagram which looks similar to another diagram. For example, in the area of logical reasoning with Euler diagrams, if one applies a reasoning rule such as “add contour” then the conclusion diagram should look like the premise, in order to facilitate understanding of the reasoning step. Such layout issues are briefly discussed in section 7. Different

abstractions are likely to be especially useful in this area.

Finally, in section 8, we present our conclusions, and indicate current and future work directions.

2 Concrete Euler diagrams and abstractions

A *concrete Euler diagram* consists of a collection of (labelled) simple closed curves in the plane, called *contours*. A *concrete region* in a concrete diagram is a set of points in the plane. It is defined inductively, starting with the *basic region* of the diagram (the set of points in the plane) and the basic region of each contour (the interior of each of the contours). Then the complement, union and intersection operations enable the generation of all of the regions of the diagram.

A *concrete zone* is a minimal region, in the sense that no other region is contained within it. The concrete diagram in fig. 1 has five zones.

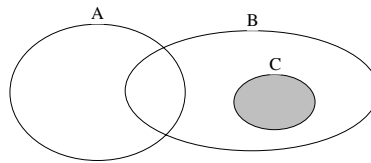


Fig. 1. A simple Euler diagram

The abstract description of a contour is given by its label. Thus, there are three abstract contours A , B and C in an abstraction of the diagram in fig. 1.

Concrete zones correspond to areas of the plane, which can be defined by the set P of contours which contain the area and the set Q of contours which exclude the area. Thus the pair (P, Q) where P and Q partition the set of contour labels is an *abstract zone*.

The abstract version of the shaded zone in fig. 1 is (BC, A) . Here we use a streamlined notation omitting set brackets and concatenating contour names for readability purposes - formally this zone is $(\{B, C\}, \{A\})$.

We use an “algebra of pairs” to construct abstract regions associated with a set of abstract contours. Any pair (R, S) , where R and S are disjoint sets of abstract contours (i.e. contour labels) represents an abstract region, and any union of such pairs also represents an abstract region. For example, the pair (\emptyset, A) is the abstract region which is an abstraction of the region outside A in a concrete diagram.

We use the symbol \emptyset to denote the union of no pairs. We allow algebraic

manipulation of abstract regions using the *split equivalence* rules

$$(X, Y) \equiv_{se} (X \cup \{c\}, Y) \cup (X, Y \cup \{c\}) \quad \text{and} \quad (X, Y) \cup (X, Y) \equiv_{se} (X, Y),$$

where c is a contour label which does not appear in X or Y . An abstract region is an equivalence class under these rules. Given a contour set, any region can be expressed uniquely as a union of zones (P, Q) , where P and Q partition the contour set. For example, $(A, B) \cup (B, \emptyset) \equiv_{se} (A, B) \cup (AB, \emptyset) \cup (B, A) \equiv_{se} (A, \emptyset) \cup (B, A)$ are different representations of the same abstract region. The representation as a union of zones can be obtained by “splitting” up any pair which does not use all of the contour labels, using the first split equivalence rule and then tidying up using the second.

Define a relation \subseteq_{se} on abstract regions as follows: Given two abstract regions r_1 and r_2 , say that $r_1 \subseteq_{se} r_2$ if the unique description of r_1 as zones (using split equivalence) is contained within the description of r_2 as zones.

However if we are in the context of a diagram (and so we have access to which zones are missing - those zones which are potentially present, using the same contours, but actually are not present) then we can define $\subseteq_{se\&z}$ which uses both split equivalence and missing zones. That is, given two abstract regions r_1 and r_2 , say that $r_1 \subseteq_{se\&z} r_2$ if the unique description of r_1 as zones, after removal of the missing zones, is contained within the description of r_2 as zones. For example, given the diagram in fig. 1, $(C, A) \subseteq_{se\&z} (B, A)$, because $(C, A) \equiv_{se} (BC, A) \cup (C, AB)$ and since (C, AB) is missing we get $(C, A) \equiv_{se\&z} (BC, A)$. Now $(B, A) \equiv_{se} (BC, A) \cup (B, AC)$ and so $(C, A) \subseteq_{se\&z} (B, A)$.

Given a set of abstract contours, define the *complement* of an abstract region, r , to be the maximal region whose zone-wise description contains none of the zones present in r . Let U denote the complement of \emptyset .

See [10] for a similar algebraic framework relating to the equivalence of abstract regions.

3 Three abstractions of an Euler diagram

The following three subsections present alternative abstract syntax for Euler diagrams. These are alternative definitions of an (abstract) Euler diagram and, importantly, section 5 will show equivalence of the abstractions by presenting well-defined mappings to derive any abstraction from any other.

For each abstraction, we give an example of a concrete diagram and use the abstraction to describe the concrete diagram, and then we give an abstract definition and an example of a concrete diagram obtained from the abstract description.

The alternative abstractions we propose depend upon describing the (concrete) contours' placements instead of describing the (concrete) zones. A naive attempt at defining such an abstraction might be to specify, for each pair of contours c_1, c_2 which of the following conditions hold:

- c_1 is contained in c_2 (so the region (c_2, c_1) is missing)
- c_2 is contained in c_1 (so the region (c_1, c_2) is missing)
- c_1 disjoint from c_2 (so the region $(c_1 c_2, \emptyset)$ is missing).

There is a serious drawback to this attempt at an abstraction, however, in that it fails to distinguish between the two diagrams shown in fig. 2.

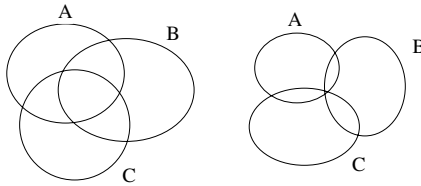


Fig. 2. Two different Euler diagrams

When proposing alternative abstractions, a benchmark test for adequacy of the abstraction is to ask whether the zone-wise abstraction can be derived from the alternative. In this naive case, no such derivation exists.

3.1 The zone-wise abstraction

For the concrete Euler diagram in fig. 1, the (abstract) zone set is

$$\{(A, BC), (AB, C), (B, AC), (BC, A), (\emptyset, ABC)\}.$$

Note that this includes the outside zone (\emptyset, ABC) .

Definition 3.1 An *abstract zone-wise Euler diagram* d is a set of contours $\mathcal{C}(d) = \{c_1, c_2, \dots, c_m\}$ and a set of zones $Z(d)$, which must contain the *outside zone* $(\emptyset, \mathcal{C}(d))$.

This zone-wise description of an Euler diagram is given explicitly in [8,10] and is a streamlined version of the abstraction used in [14,15]. Note that here we give a more general theory, not imposing conditions such as the existence of a zone inside each contour. The decision of imposing such conditions can then be deferred until the application domain is chosen.

An alternative way to view the zone-wise description is to order the set of contours and give a binary number (whose number of digits is the number of

contours) to represent a zone, taking 1 if the zone is “inside” the corresponding contour and 0 if it is “outside”. For example, if we order the contours in the diagram in fig. 1 lexicographically, then the shaded zone (BC, A) corresponds to 011. In [2], the zones are called *subfaces* and each of the points in a zone (in a concrete diagram) has an associated *bitmask* which is effectively this binary representation.

3.2 Containment conditions for contours

We give an abstraction which, for each contour c , specifies its placement relative to the other contours. This specification is done by giving a pair of regions $R_{small}(c)$ and $R_{big}(c)$.

An abstract region r is *minimal* subject to a given condition, X , if there is no alternative region s , with $s \subseteq r$, which also satisfies X . Similarly, r is *maximal* subject to a given condition, X , if there is no alternative region s , with $r \subseteq s$, which also satisfies X .

Given a concrete diagram d containing a contour c , $R_{small}(c)$ is the maximal region contained within c , describable without using c , and $R_{big}(c)$ is the minimal region containing c , describable without using c .

The diagram in fig. 1, has contour label set $\{A, B, C\}$ and a set of containment conditions is:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= (\emptyset, C), \\ R_{small}(B) &= (C, \emptyset) & R_{big}(B) &= U, \\ R_{small}(C) &= \emptyset & R_{big}(C) &= (B, A). \end{aligned}$$

The following definition allows us to work with containment condition abstractions even in the absence of a drawn diagram:

Definition 3.2 A *containment condition Euler diagram* is a set of contours $\mathcal{C}(d) = \{c_1, c_2, \dots, c_m\}$ and, for each contour, $c_i \in \mathcal{C}(d)$, a containment condition. A *containment condition* (for c_i) is a pair of regions $R_{small}(c_i)$ and $R_{big}(c_i)$, which are describable without using c_i (i.e. there is a representative only using c_j 's where $c_j \in \{c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_m\}$). In addition, $(\emptyset, \mathcal{C}(d)) \not\subseteq_{se} R_{small}(c)$, for any $c \in \mathcal{C}(d)$.

This last condition imposed on each $R_{small}(c)$ corresponds to asserting the presence of the outside zone in a diagram. This correspondence is made explicit when we map between zone-wise abstractions and contour containment abstractions in section 5.

Note that two containment condition abstract diagrams may use different representations of the regions. The following is an equivalent abstraction for the Euler diagram in fig. 1, using split-equivalence for $R_{big}(A)$:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= (\emptyset, BC) \cup (B, C) \\ R_{small}(B) &= (C, \emptyset) & R_{big}(B) &= U \\ R_{small}(C) &= \emptyset & R_{big}(C) &= (B, A). \end{aligned}$$

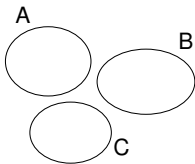


Fig. 3. Three disjoint contours

Abstract diagrams defined using the containment conditions may be *equivalent* under more than just the split equivalence rule (they may also use missing zones). Consider the concrete diagram in fig. 3, for example. Containment conditions for this diagram could include either $R_{big}(C) = (\emptyset, AB)$ or $R_{big}(C) = (\emptyset, AB) \cup (AB, \emptyset)$. These are not equivalent regions, using split equivalence only, but are equivalent regions (and hence give rise to equivalent abstract diagrams) using missing zones as well.

Containment condition Euler diagrams will be said to be equivalent if they map to the same zone-wise Euler diagram under the map defined in section 5.

3.3 Building sequences

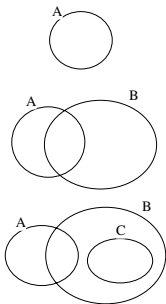


Fig. 4. A simple building sequence

Here, we give an abstraction which is based on the idea of building a diagram as an animation, by adding the contours one at a time. An example of such a diagram animation was explored in [17].

Given a concrete diagram and a total ordering of the contour set c_1, c_2, \dots, c_m , a *building sequence* specifies the placement of each contour c_i using a pair of regions $R_{small}^*(c_i)$, which will be maximal inside (c_i, \emptyset) , and $R_{big}^*(c_i)$, which will be minimal containing (c_i, \emptyset) , describable using only contours c_1, \dots, c_{i-1} .

For example, a building sequence abstraction for the diagram in fig. 4, is given by the contour sequence $\langle A, B, C \rangle$ and

$$\begin{aligned} R_{small}^*(A) &= \emptyset & R_{big}^*(A) &= U \\ R_{small}^*(B) &= \emptyset & R_{big}^*(B) &= U \\ R_{small}^*(C) &= \emptyset & R_{big}^*(C) &= (B, A). \end{aligned}$$

Definition 3.3 A *building sequence Euler diagram* is a sequence of distinct contours $\langle c_1, c_2, \dots, c_m \rangle$ and, for each contour $c_i \in \mathcal{C}(d) = \{c_1, c_2, \dots, c_m\}$, a pair of regions $R_{small}^*(c_i)$ and $R_{big}^*(c_i)$, which are describable without using c_i, \dots, c_m . In addition we impose the condition that $(\emptyset, \mathcal{C}(d)) \not\subseteq_{se} R_{small}^*(c_i)$, for any c_i .

4 Examples

In this section we compare the abstractions of some concrete diagrams in specific examples. We remark on the potential use of the differing abstractions for identifying certain properties of a diagram.

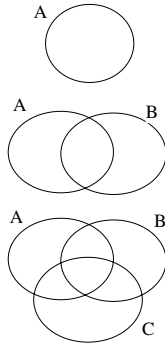


Fig. 5. Building a Venn diagram

The zone-wise description of the Venn diagram on three contours, shown at the bottom of fig. 5, has contour set $\{A, B, C\}$ and is given by:

$$\{(ABC, \emptyset), (AB, C), (AC, B), (BC, A), (A, BC), (B, AC), (C, AB), (\emptyset, ABC)\}.$$

A containment condition abstraction, has contour set $\{A, B, C\}$, and conditions:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= U, \\ R_{small}(B) &= \emptyset & R_{big}(B) &= U, \\ R_{small}(C) &= \emptyset & R_{big}(C) &= U. \end{aligned}$$

A building sequence abstraction, with contour sequence $\langle A, B, C \rangle$, is given by:

$$\begin{aligned} R_{small}^*(A) &= \emptyset & R_{big}^*(A) &= U, \\ R_{small}^*(B) &= \emptyset & R_{big}^*(B) &= U, \\ R_{small}^*(C) &= \emptyset & R_{big}^*(C) &= U. \end{aligned}$$

Notice that in both the containment condition and the building sequence cases, the regions' descriptions involve no contours. Thus it is easy to identify if a diagram is actually a Venn diagram (it requires counting the number of zones for the zone-wise abstraction). This abstraction may also allow us to recognize if contours have “Venn-like” properties (meets every zone). This can be thought of as a kind of contour independence which can be generalized (similar to spider independence in [4,5]). Diagrams with contour independence may convey information that could equally be expressed using a conjunction of simpler diagrams. This has potential uses in either drawing or storing diagram information.

The zone-wise description of the diagram with three disjoint contours, as shown at the bottom of fig. 6, has contour set $\{A, B, C\}$, and is given by:

$$\{(A, BC), (B, AC), (C, AB), (\emptyset, ABC)\}.$$

A containment condition abstraction has contour set $\{A, B, C\}$, and conditions:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= (\emptyset, BC), \\ R_{small}(B) &= \emptyset & R_{big}(B) &= (\emptyset, AC), \\ R_{small}(C) &= \emptyset & R_{big}(C) &= (\emptyset, AB). \end{aligned}$$

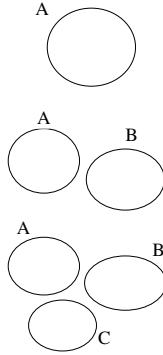


Fig. 6. Building a diagram of disjoint contours

A building sequence abstraction, with contour sequence $\langle A, B, C \rangle$, is given by:

$$\begin{aligned}
 R_{small}^*(A) &= \emptyset & R_{big}^*(A) &= U, \\
 R_{small}^*(B) &= \emptyset & R_{big}^*(B) &= (\emptyset, A), \\
 R_{small}^*(C) &= \emptyset & R_{big}^*(C) &= (\emptyset, AB).
 \end{aligned}$$

Symmetries in the containment conditions and the obvious pattern in the building sequence potentially gives ways to identify symmetries and disjointness of contours.

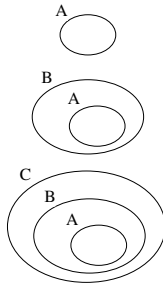


Fig. 7. Adding outer concentric contours

The diagram in fig. 7 has been built by adding outer, containing contours in turn. The zone-wise description of the resulting diagram has contour set $\{A, B, C\}$, and is given by:

$$\{(ABC, \emptyset), (BC, A), (C, AB), (\emptyset, ABC)\}.$$

A containment condition abstraction, has contour set $\{A, B, C\}$, and condi-

tions:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= (B, \emptyset), \\ R_{small}(B) &= (A, \emptyset) & R_{big}(B) &= (C, \emptyset), \\ R_{small}(C) &= (B, \emptyset) & R_{big}(C) &= U. \end{aligned}$$

A building sequence abstraction, with contour sequence $\langle A, B, C \rangle$, is given by:

$$\begin{aligned} R_{small}^*(A) &= \emptyset & R_{big}^*(A) &= U, \\ R_{small}^*(B) &= (A, \emptyset) & R_{big}^*(B) &= U, \\ R_{small}^*(C) &= (B, \emptyset) & R_{big}^*(C) &= U. \end{aligned}$$

One may be able to generalise these and other examples in order to identify nestedness (where nestedness can be thought of as being able to express one diagram as the image of an embedding of one diagram into a zone of another diagram).

The information conveyed by the diagram in fig. 7 can be separated and depicted as the conjunction of two diagrams, as shown in fig. 8. It appears that the building sequence expression is more likely to highlight opportunities to separate information into smaller pieces.

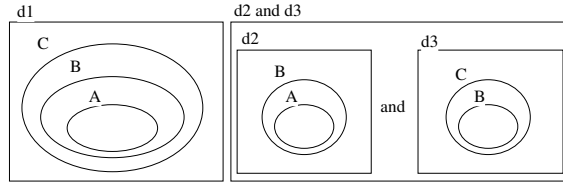


Fig. 8. A diagram equivalent to a conjunction: with nesting

The diagram d_1 in fig. 9 can be separated into a conjunction of two diagrams, even though there is no nesting in this example. Building sequence conditions, with contour sequence $\langle A, B, C, D \rangle$ can be given by:

$$\begin{aligned} R_{small}^*(A) &= \emptyset & R_{big}^*(A) &= U, \\ R_{small}^*(B) &= \emptyset & R_{big}^*(B) &= (\emptyset, A), \\ R_{small}^*(C) &= \emptyset & R_{big}^*(C) &= U, \\ R_{small}^*(D) &= \emptyset & R_{big}^*(D) &= (\emptyset, C). \end{aligned}$$

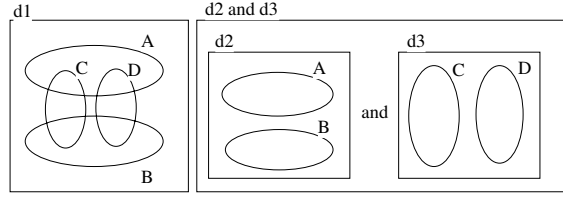


Fig. 9. A diagram equivalent to a conjunction: without nesting

5 Equivalence of abstractions

We define mappings between the three abstractions described, define equivalence classes of containment abstractions and building sequences, and also show that the mappings give an equivalence between the three abstractions.

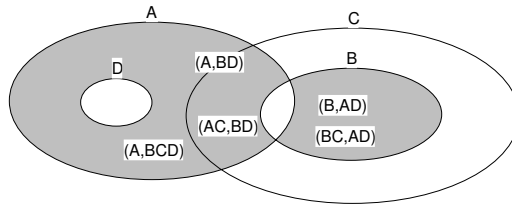
Let \mathcal{D}_z be the class of zone-wise Euler diagrams, \mathcal{D}_{cc} be the class of containment condition Euler diagrams, and let \mathcal{D}_{bs} be the class of building sequence Euler diagrams.

First of all we will define maps from sets of zones to containment conditions and vice versa. Then we'll investigate properties of these maps.

5.1 Zone-wise to containment conditions mapping

Definition 5.1 Let $d_z \in \mathcal{D}_z$ with contour set $\mathcal{C}(d_z)$ and zone set $Z = Z(d_z)$. Define the map $\phi : \mathcal{D}_z \rightarrow \mathcal{D}_{cc}$ by $\phi(d_z) = d_{cc}$, where d_{cc} has: contour set $\mathcal{C}(d_{cc}) = \mathcal{C}(d_z)$ and for each contour $c \in \mathcal{C}(d_{cc})$,

- (i) $R_{small}(c) = \bigcup \{(P, Q) : c \not\subseteq P \cup Q \wedge (P \cup \{c\}, Q) \in Z \wedge (P, Q \cup \{c\}) \notin Z\}$,
- (ii) $R_{big}(c) = \bigcup \{(P, Q) : c \not\subseteq P \cup Q \wedge (P \cup \{c\}, Q) \in Z\}$.

Fig. 10. Understanding $R_{small}(c)$ and $R_{big}(c)$

Example 5.2 The diagram in fig. 10 highlights the region (B, AD) which is in both $R_{small}(C)$ and $R_{big}(C)$ because (BC, AD) is present and (B, ACD) is absent from the zone set. Notice that $(B, AD) \equiv_{semz} (BC, AD)$ in this case. Also highlighted is the region (A, BD) which is in $R_{big}(C)$ because (AC, BD)

is present, but is not in $R_{small}(C)$ because (A, BCD) is also present. In this case $(A, BD) \equiv_{se} (AC, BD) \cup (A, BCD)$.

Example 5.3 For the example in fig. 1, the zone set is

$$\{(A, BC), (AB, C), (B, AC), (BC, A), (\emptyset, ABC)\}.$$

We build $R_{small}(A)$ and $R_{big}(A)$ by considering each of the two zones which have A in their first component in turn:

- for (A, BC) , note that (\emptyset, ABC) is also present,
- for (AB, C) , note that (B, AC) is also present.

Therefore, neither (\emptyset, BC) nor (B, C) contribute to $R_{small}(A) = \emptyset$, but both contribute to $R_{big}(A) = (\emptyset, BC) \cup (B, C) \equiv_{se} (\emptyset, C)$. On the other hand, consider the contour B , which contains zones $(B, AC), (AB, C), (BC, A)$.

- For (B, AC) , note that (\emptyset, ABC) is also present.
- For (AB, C) , note that (A, BC) is also present.
- For (BC, A) , note that (C, AB) is absent.

The third case provides a contribution to $R_{small}(B) = (C, A)$, and all three cases contribute to $R_{big}(B) = (\emptyset, AC) \cup (A, C) \cup (C, A)$.

The following theorem shows that ϕ is a well-defined map and also describes properties of the containment conditions defined under this map.

Theorem 5.4 *Let $d \in \mathcal{D}_z$ be a zone-wise Euler diagram. Then, under ϕ , we have that, for each contour $c \in \mathcal{C}(d)$,*

- (i) $R_{small}(c)$ and $R_{big}(c)$ are unions of regions which do not reference c .
- (ii) $(\emptyset, \mathcal{C}(d)) \not\subseteq_{se} R_{small}(c)$.
- (iii) $R_{small}(c) \subseteq_{se} (\{c\}, \emptyset) \subseteq_{se} R_{big}(c)$.
- (iv) $R_{small}(c)$ is maximal and $R_{big}(c)$ is minimal subject to the above conditions.

Proof.

1. The construction ensures that $R_{small}(c)$ and $R_{big}(c)$ are unions of regions (P, Q) which do not reference c .

2. $R_{small}(c)$ contains $(\emptyset, \mathcal{C}(d) - \{c\})$ only if

- the zone $(c, \mathcal{C}(d) - \{c\})$ is present in d and
- the zone $(\emptyset, \mathcal{C}(d))$ is missing from d .

The second condition, along with presence of the outside zone, ensures that $(\emptyset, \mathcal{C}(d))$ is not in $R_{small}(c)$.

3. To see that $R_{small}(c)$ is contained in (c, \emptyset) , we have

$$\begin{aligned} & \bigcup \{(P, Q) : (P \cup \{c\}, Q) \in Z \wedge (P, Q \cup \{c\}) \notin Z\} \\ \equiv_{se} & \bigcup \{(P \cup \{c\}, Q) \cup (P, Q \cup \{c\}) : (P \cup \{c\}, Q) \in Z \wedge (P, Q \cup \{c\}) \notin Z\} \\ \equiv_{semsz} & \bigcup \{(P \cup \{c\}, Q) : (P \cup \{c\}, Q) \in Z \wedge (P, Q \cup \{c\}) \notin Z\} \end{aligned}$$

and each contribution $(P \cup \{c\}, Q)$ is in the region $(\{c\}, \emptyset)$.

Similarly, to see that $R_{big}(c)$ contains (c, \emptyset) , take each zone, $(P \cup \{c\}, Q)$ in (c, \emptyset) in turn. For each of these zones,

$$(P \cup \{c\}, Q) \subseteq_{semsz} (P, Q) \subseteq \bigcup_{(R \cup \{c\}, S) \in Z} (R, S) = R_{big}(c).$$

4. To show that $R_{small}(c)$ is maximal, consider a region R with

$$R = \bigcup \{(K_i, L_i)\} \subseteq_{semsz} (\{c\}, \emptyset)$$

where there is no occurrence of c in any K_i or L_i . Without loss of generality (since we can always split them up using the split equivalence), we can assume that the regions (K_i, L_i) use all labels except c , so $K_i \cup L_i = \mathcal{C}(d) - \{c\}$. Since $(K_i, L_i) \equiv_{se} (K_i \cup \{c\}, L_i) \cup (K_i, L_i \cup \{c\})$, and each component $(K_i, L_i) \subseteq_{semsz} (\{c\}, \emptyset)$, we have $(K_i, L_i \cup \{c\}) \subseteq_{semsz} (\{c\}, \emptyset)$ which is impossible unless $(K_i, L_i \cup \{c\})$ is a missing zone. The fact that $(K_i, L_i \cup \{c\})$ is a missing zone ensures that either $(K_i \cup \{c\}, L_i)$ is missing or (K_i, L_i) contributes to $R_{small}(c)$, so $R \subseteq_{semsz} R_{small}(c)$.

Similarly, to show that $R_{big}(c)$ is minimal, consider a region R with

$$(\{c\}, \emptyset) \subseteq_{semsz} \bigcup \{(K_i, L_i)\} = R$$

where there is no occurrence of c in any K_i or L_i . Without loss of generality, we can assume that the regions (K_i, L_i) use all contour labels except c , so $K_i \cup L_i = \mathcal{C}(d) - \{c\}$. Each zone $(P \cup \{c\}, Q)$ is in some (K_i, L_i) , so each (P, Q) in $R_{big}(c)$ is in some (K_i, L_i) of R , and $R_{big}(c) \subseteq_{semsz} R$. \square

5.2 Containment conditions to zone-wise mapping

Definition 5.5 Let $d_{cc} \in \mathcal{D}_{cc}$ with contour set $\mathcal{C}(d_{cc})$ and containment conditions $R_{small}(c)$ and $R_{big}(c)$ for each $c \in \mathcal{C}(d_{cc})$. Define the map $\psi : \mathcal{D}_{cc} \rightarrow \mathcal{D}_z$

by $\psi(d_{cc}) = d_z$, where d_z has contour set $\mathcal{C}(d_z) = \mathcal{C}(d_{cc})$ and zone set $Z(d_z)$ defined by: zone (P, Q) is present if

- $(P, Q) \subseteq_{se} R_{big}(c) \ \forall c \in P$, and
- $(P, Q) \not\subseteq_{se} R_{small}(c) \ \forall c \in Q$.

Example 5.6 In our previous example, based on fig. 1, we found the set of containment conditions:

$$\begin{aligned} R_{small}(A) &= \emptyset & R_{big}(A) &= (\emptyset, C), \\ R_{small}(B) &= (C, \emptyset) & R_{big}(B) &= U, \\ R_{small}(C) &= \emptyset & R_{big}(C) &= (B, A). \end{aligned}$$

Write these as a union of zones using split equivalence:

$$\begin{aligned} R_{small}(A) &= \emptyset, \\ R_{big}(A) &= (AB, C) \cup (A, BC) \cup (B, AC) \cup (B, AC), \\ R_{small}(B) &= (ABC, \emptyset) \cup (AC, B) \cup (BC, A) \cup (C, AB), \\ R_{big}(B) &= U, \\ R_{small}(C) &= \emptyset \\ R_{big}(C) &= (BC, A) \cup (B, AC). \end{aligned}$$

To test if (AB, C) is a zone under the map ψ , for example, note that

- (AB, C) is in $R_{big}(A) = (AB, C) \cup (A, BC) \cup (B, AC) \cup (B, AC)$,
- (AB, C) is in $R_{big}(B) = U$,
- (AB, C) is missing from $R_{small}(C) = \emptyset$.

Therefore, (AB, C) is included in the zone-wise description.

To see that (AC, B) is omitted from the zone-set, we notice that (AC, B) is missing from $R_{big}(A)$.

Theorem 5.7 *This map ψ is well-defined.*

Proof. Let d be a containment condition Euler diagram, with contour set $\mathcal{C}(d)$. We have to show that the outside zone is present in $\psi(d)$. The outside zone $(\emptyset, \mathcal{C}(d))$ is included when mapping from containment conditions (under ψ), if

$$(\emptyset, \mathcal{C}(d)) \subseteq_{se} R_{small}(c) \quad \forall c \in \mathcal{C}(d)$$

This is exactly the condition for validity of the containment conditions. \square

Theorem 5.8 *Let d_z denote the zone-wise Euler diagram which is the image of a containment condition Euler diagram d_{cc} under ψ . Then, in d_z , the regions $R_{small}(c)$ and $R_{big}(c)$ satisfy*

$$R_{small}(c) \subseteq_{se} (c, \emptyset) \subseteq_{se} R_{big}(c).$$

Proof. The first property of the definition of zones using ψ says that any present zone $z = (P, Q)$ in d_z with $c \in P$ is in $R_{big}(c)$. Since every zone in (c, \emptyset) , using split equivalence, satisfies this condition, we have that $(c, \emptyset) \subseteq_{se} R_{big}(c)$. Similarly the second property says that any present zone $z = (P, Q)$ in d_z is not in $R_{small}(c)$ for any $c \in Q$. Therefore, $R_{small}(c)$ contains only present zones (P, Q) which have $c \notin Q$. Thus, since c must appear in any zone description (P, Q) and it is not in Q , we have $c \in P$ and so $R_{small}(c) \subseteq_{se} (c, \emptyset)$. □

This mapping allows us to define an equivalence of containment condition abstractions. Two containment condition Euler diagrams are *equivalent* if they map to the same zone-wise Euler diagram.

We check that the composite mapping from a zone-wise Euler diagram to a set of contour containments, and then back to a zone-wise Euler diagram is just the identity map. That is, mapping from a zone-wise abstraction to generate a set of containment conditions, and then using those containment conditions to generate a set of zones retrieves the same set of zones back again.

Theorem 5.9 *Let d_z be a zone-wise Euler diagram, with a non-empty set of contours $\mathcal{C}(d_z)$ and zone set $Z(d_z)$. Then $\psi\phi(d_z) = d_z$.*

Proof. We must show that

- (i) $z \in Z(d_z)$ implies that $z \in Z(\psi\phi(d_z))$, and
 - (ii) $z \notin Z(d_z)$ implies that $z \notin Z(\psi\phi(d_z))$.
- (i) Suppose that $z = (P, Q) \in Z(d_z)$. Applying ϕ to d_z gives us that for each contour c ,

$$(a) \quad R_{small}(c) = \bigcup \{ (K, L) : c \notin K \cup L \wedge (K \cup \{c\}, L) \in Z(d_z) \wedge (K, L \cup \{c\}) \notin Z(d_z) \} \text{ and}$$

$$(b) \quad R_{big}(c) = \bigcup \{ (K, L) : c \notin K \cup L \wedge (K \cup \{c\}, L) \in Z(d_z) \}.$$

Under ψ , we have that zone (P, Q) is present in $\psi\phi(d_z)$ if

- $(P, Q) \subseteq_{se} R_{big}(p) \quad \forall p \in P$, and
- $(P, Q) \not\subseteq_{se} R_{small}(q) \quad \forall q \in Q$.

For each $p \in P$, $(P - \{p\}, Q) \subseteq_{se} R_{big}(p)$, using condition (b), because $p \notin (P - \{p\}) \cup Q$ and $z = (P, Q)$ is a zone of d_z . Therefore $(P, Q) \subseteq_{se} R_{big}(p)$.

Also, for each $q \in Q$, $(P, Q - \{q\}) \not\subseteq_{se} R_{small}(q)$, using condition (a), because if it was then $(P, Q) = (P, Q - \{q\} \cup \{q\}) \notin Z(d_z)$, and we know that $(P, Q) \in Z(d_z)$. Since $R_{small}(q)$ is expressible without reference to q , if $(P, Q) \subseteq_{se} R_{small}(q)$ then we must have that $(P, Q - \{q\}) \subseteq_{se} R_{small}(q)$ which is false. Therefore, $(P, Q) \not\subseteq_{se} R_{small}(q) \forall q \in Q$. Hence z is also a zone in $Z(\psi\phi(d_z))$.

- (ii) Take a missing zone $z = (P, Q) \notin Z(d_z)$. The outside zone is present in $Z(d_z)$ so $P \neq \emptyset$. The zone (P, Q) is also missing from $Z(\psi(\phi(d_z)))$ if following condition holds:

- (a) for some $p \in P$, $(P, Q) \not\subseteq_{se} R_{big}(p)$, or
 (b) for some $q \in Q$, $(P, Q) \subseteq_{se} R_{small}(q)$.

Substituting, using the definitions of $R_{big}(p)$ and $R_{small}(q)$, gives the condition:

- (a) for some $p \in P$, $(P - p, Q) \not\subseteq_{se} R_{big}(p)$, or
 (b) for some $q \in Q$, $(P, Q - q) \subseteq_{se} R_{small}(q)$.

Since P is non-empty and $(P, Q) \notin Z(d_z)$, we have that $(P - p, Q) \not\subseteq_{se} R_{big}(p)$ as required. □

5.3 Zone-wise to building sequence mapping

We define a map $\phi^* : \mathcal{D}_z \rightarrow \mathcal{D}_{bs}$. Note that since ϕ^* involves a choice of ordering of contours we really have a family of these maps, one for each ordering, but we will not clutter the notation any more by adding subscripts to ϕ^* for example.

Definition 5.10 Let $d_z \in \mathcal{D}_z$ with contour set $\mathcal{C}(d_z) = \{c_1, \dots, c_m\}$ and zone set $Z(d_z)$. Let $(d_z; c_1, \dots, c_i)$ denote the diagram d_z if we only had contours c_1, \dots, c_i present (the zones in $(d_z; c_1, \dots, c_i)$ are zonal regions in the diagram d_z). Define the map $\phi^* : \mathcal{D}_z \rightarrow \mathcal{D}_{bs}$ by $\phi^*(d_z) = d_{bs}$, where d_{bs} has: contour sequence $\langle c_1, \dots, c_m \rangle$, and for each contour c_i ,

- (i) $R_{small}^*(c_i) = \bigcup \{(P, Q) : c_i \notin P \cup Q \wedge (P \cup \{c_i\}, Q) \in Z(d_z; c_1, \dots, c_i) \wedge (P, Q \cup \{c_i\}) \notin Z(d_z; c_1, \dots, c_i)\}$ and
 (ii) $R_{big}^*(c_i) = \bigcup \{(P, Q) : c_i \notin P \cup Q \wedge (P \cup \{c_i\}, Q) \in Z(d_z; c_1, \dots, c_i)\}$.

Theorem 5.11 Let $d_z \in \mathcal{D}_z$ be a zone-wise Euler diagram. Then, under ϕ^* , we have that, for each contour $c_i \in \mathcal{C}(d_z)$,

- (i) $R_{small}^*(c_i)$ and $R_{big}^*(c_i)$ are unions of regions which don't reference c_i, \dots, c_n .
 (ii) $(\emptyset, \mathcal{C}(d)) \not\subseteq \phi^*(R_{small}^*(c_i))$.

- (iii) $R_{small}^*(c_i) \subseteq_{sems} (\{c_i\}, \emptyset) \subseteq_{sems} R_{big}^*(c_i)$ – using the missing zones in $(d_z; c_1, \dots, c_i)$. Note that this is true in d_z and not just in $(d_z; c_1, \dots, c_i)$.
- (iv) $R_{small}^*(c_i)$ is maximal and $R_{big}^*(c_i)$ is minimal subject to the above conditions.

Proof. Follows from the containment condition case (see theorem 5.4) applied to the diagram $(d_z; c_1, \dots, c_i)$. \square

Example 5.12 For the diagram d in fig. 1, the zone set is:

$$Z(d) = \{(A, BC), (AB, C), (B, AC), (BC, A), (\emptyset, ABC)\}.$$

Give the contours an ordering $\langle A, B, C \rangle$. The diagram $(d; A)$ consists of just the contour A and we have $R_{small}^*(A) = \emptyset$ and $R_{big}^*(A) = U$. The set of zones in the diagram containing contours A and B (which can be obtained by deleting C from the descriptions in $Z(d)$ above) is:

$$Z(d; A, B) = \{(A, B), (AB, \emptyset), (B, A), (\emptyset, AB)\}.$$

Then $R_{big}^*(B)$ is the union of regions which are obtained by deleting B from those descriptions in $Z(d; A, B)$ with B in the first coordinate. So $R_{big}^*(B) = (A, \emptyset) \cup (\emptyset, A) = U$. Since (A, B) and (\emptyset, AB) are also both present in $Z(d; A, B)$, $R_{small}^*(B) = \emptyset$. Similarly $R_{big}^*(C)$ is the union of regions which can be obtained by deleting C from the descriptions of the zones in $Z(d)$ with C in the first coordinate. Thus $R_{big}^*(C) = (B, A)$. Since (B, AC) is also present in $Z(d)$, the region (B, A) is not included in $R_{small}^*(C)$ and $R_{small}^*(C) = \emptyset$.

5.4 Building sequence to zone-wise mapping

Definition 5.13 Let $d_{bs} \in \mathcal{D}_{bs}$ with contour sequence $\langle c_1, \dots, c_m \rangle$ and regions $R_{small}^*(c_i)$ and $R_{big}^*(c_i)$ for each contour c_i . Let $rem_{j, \dots, m}(P, Q)$, where $P \cup Q = \{c_1, \dots, c_m\}$, denote the pair $(P - \{c_j, \dots, c_m\}, Q - \{c_j, \dots, c_m\})$. If (P, Q) is a zone in diagram d , then $rem_{j, \dots, m}(P, Q)$ is a zone in diagram $(d; c_1, \dots, c_{j-1})$.

Define the map $\psi^* : \mathcal{D}_{bs} \rightarrow \mathcal{D}_z$ by $\psi^*(d_{bs}) = d_z$, where d_z has contour set $\{c_1, \dots, c_m\}$ and zone set $\mathcal{Z}(d_z)$ defined by: zone (P, Q) is present if

- $rem_{j, \dots, m}(P, Q) \subseteq_{se} R_{big}^*(c_j) \forall c_j \in P$, and
- $rem_{j, \dots, m}(P, Q) \not\subseteq_{se} R_{small}^*(c_j) \forall c_j \in Q$.

Theorem 5.14 This map ψ^* is well-defined.

Proof. The proof is similar to that given for theorem 5.7. \square

The following properties of building sequences and the maps ϕ^* and ψ^* are similar to the containment conditions case. The proofs are similar and so they are omitted.

Theorem 5.15 *Let d_z denote the zone-wise Euler diagram which is the image of a building sequence Euler diagram d_{bs} under ψ^* . Then, in d_z , which has contour set $\{c_1, \dots, c_m\}$, the regions $R_{small}^*(c)$ and $R_{big}^*(c)$ have*

$$R_{small}^*(c_i) \subseteq_{sems} (c_i, \emptyset) \subseteq_{sems} R_{big}^*(c_i).$$

Theorem 5.16 *Let d_z be a zone-wise Euler diagram, with a non-empty set of contours $\mathcal{C}(d_z)$ and with zone set $Z(d_z)$. Then $\psi^*\phi^*(d_z) = d_z$.*

6 Semantics

We compare abstractions in the contexts of Euler diagram semantics and reasoning. For notational simplicity, we will identify contour labels (A, B, \dots) with the sets (A, B, \dots) that they represent.

For each abstraction, we can define the semantics of an Euler diagram. The zone-wise abstraction has been interpreted in the past [8] using a *plane tiling condition*. The plane tiling condition is

$$\bigcup_{(P,Q) \in Z(d)} \left(\bigcap_{c \in P} c \cap \bigcap_{c \in Q} \bar{c} \right) = U,$$

where \bar{c} denotes the complement of c .

Effectively this asserts that the union of the zones present in a diagram represents the universal set. Any zone which is absent from a diagram represents the empty set.

The semantic interpretation arising from a containment condition Euler diagram is given by:

$$\bigwedge_{c \in C} (R_{small}(c) \subseteq c \subseteq R_{big}(c)).$$

The semantic interpretation arising from a building sequence is given by:

$$\bigwedge_{c \in C} (R_{small}^*(c) \subseteq c \subseteq R_{big}^*(c)).$$

Both of these abstractions' statements correspond to specifying where each contour c is in relation to other contours.

Example 6.1 For the example in fig. 1, the plane tiling condition is

$$(A \cap B \cap \overline{C}) \cup (\overline{A} \cap B \cap C) \cup (A \cap \overline{B} \cap \overline{C}) \cup (\overline{A} \cap B \cap \overline{C}) \cup (\overline{A} \cap \overline{B} \cap \overline{C}) = U.$$

On the other hand, the containment conditions are easily interpreted as set containments. The second abstraction gives

$$(\emptyset \subseteq A \subseteq \overline{C}) \wedge (C \subseteq B \subseteq U) \wedge (\emptyset \subseteq C \subseteq B \cap \overline{A}).$$

The building sequence abstraction gives

$$(\emptyset \subseteq A \subseteq U) \wedge (\emptyset \subseteq B \subseteq U) \wedge (\emptyset \subseteq C \subseteq B \cap \overline{A}).$$

Example 6.2 In the case of the Venn diagram, shown in fig. 5, the plane tiling condition gives

$$(A \cap B \cap C) \cup (A \cap B \cap \overline{C}) \cup (A \cap \overline{B} \cap C) \cup (\overline{A} \cap B \cap C) \cup (A \cap \overline{B} \cap \overline{C}) \\ \cup (\overline{A} \cap B \cap \overline{C}) \cup (\overline{A} \cap \overline{B} \cap C) \cup (\overline{A} \cap \overline{B} \cap \overline{C}) = U.$$

On the other hand, the containment conditions give

$$(\emptyset \subseteq A \subseteq U) \wedge (\emptyset \subseteq B \subseteq U) \wedge (\emptyset \subseteq C \subseteq U),$$

and the building sequence gives

$$(\emptyset \subseteq A \subseteq U) \wedge (\emptyset \subseteq B \subseteq U) \wedge (\emptyset \subseteq C \subseteq U).$$

In this example, all of these semantic statements are tautologies because the Venn diagram with no annotation conveys no information about the sets.

In general, the containment condition and building sequence abstractions produces more compact semantic statements and arguably appear more readable than the plane tiling condition statement.

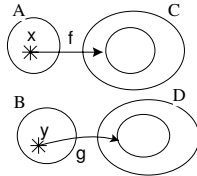


Fig. 11. A constraint diagram

When interpreting extensions of Euler diagrams which include syntax for explicitly expressing universal and existential quantification, we may be unable

to always use a zone-wise description of contours. Constraint diagrams (see [4,5] for example) are such an extension which can be used to diagrammatically represent logical statements. They are especially appropriate in the context of object-oriented modelling [9]. The constraint diagram shown in fig. 11 can be interpreted as the sets A, B, C and D are pairwise disjoint and $\forall x \in A(x.f \subseteq C) \wedge \forall y \in B(y.g \subseteq D)$, where the dot notation is the usual object oriented navigational expression which denotes relational image. Note that these unlabelled (derived) contours have corresponding semantics $x.f$ and $y.g$ and represent collections of sets rather than single sets like the labelled (given) contours. The information of the containment conditions, $\emptyset \subseteq x.f \subseteq C$ and $\emptyset \subseteq y.g \subseteq D$, for these unlabelled contours could not be re-expressed as part of a plane tiling condition because at no point in the sentence are both x and y in scope.

Assigning semantics to diagrams enables reasoning (see [3,11,14]). Given diagram $d1$, we can apply diagram transformations to create (or entail) $d2$, where $d2$ has equivalent or weaker semantics. Typical kinds of transformations for Euler diagrams are *add/remove zone* and *add/remove contour*. Examples of the transformations of adding a contour and of adding a zone are shown in fig. 12.

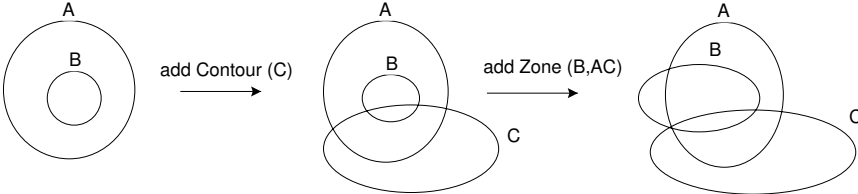


Fig. 12. Transformations of Euler diagrams

It will be interesting to investigate how the provision of different diagram abstractions guides us towards different kinds of transformation rules. Certain rules will be easier to express, or analyse, using one abstraction rather than another. For example, the transformations *add/remove contour* are easy to describe given a suitable building sequence, whereas the *add/remove zone* transformations are likely to be more suited to the zone-wise abstraction.

For example, the zone-wise descriptions of the three diagrams in fig. 12 are given by:

$$(i) \{(AB, \emptyset), (A, B), (\emptyset, AB)\}.$$

$$(ii) \{(ABC, \emptyset), (AB, C), (AC, B), (A, BC), (C, AB), (\emptyset, ABC)\}.$$

(iii) $\{(ABC, \emptyset), (AB, C), (AC, B), (A, BC), (C, AB), (\emptyset, ABC), (B, AC)\}$.

Compare with a building sequence abstraction of the same diagrams, with contour sequence $\langle A, B, C \rangle$:

$$(i) \quad \begin{array}{ll} R_{small}^*(A) = \emptyset, & R_{big}^*(A) = U, \\ R_{small}^*(B) = \emptyset, & R_{big}^*(B) = (A, \emptyset). \end{array}$$

$$(ii) \quad \begin{array}{ll} R_{small}^*(A) = \emptyset, & R_{big}^*(A) = U, \\ R_{small}^*(B) = \emptyset, & R_{big}^*(B) = (A, \emptyset), \\ R_{small}^*(C) = \emptyset, & R_{big}^*(C) = U. \end{array}$$

$$(iii) \quad \begin{array}{ll} R_{small}^*(A) = \emptyset, & R_{big}^*(A) = U, \\ R_{small}^*(B) = \emptyset, & R_{big}^*(B) = U, \\ R_{small}^*(C) = \emptyset, & R_{big}^*(C) = (AB, \emptyset) \cup (A, B) \cup (\emptyset, AB). \end{array}$$

7 Layout

Using the zone-wise abstraction, algorithms for drawing concrete diagrams given abstract ones have been investigated [6,12,16]. The difficulties in drawing will depend upon the well-formedness conditions imposed (not allowing concurrent lines or triple points for example). Using different abstractions may allow us to identify when a certain condition is essential in order to draw a diagram.

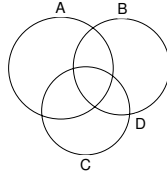


Fig. 13. Contour D requires concurrency

For example, the diagram in fig.13 is a concrete drawing of the following containment condition abstract Euler diagram with contour set $\{A, B, C, D\}$ and conditions:

$$\begin{array}{ll} R_{small}(A) = \emptyset & R_{big}(A) = U, \\ R_{small}(B) = \emptyset & R_{big}(B) = U, \\ R_{small}(C) = \emptyset & R_{big}(C) = U, \\ R_{small}(D) = (A, \emptyset) \cup (B, \emptyset) \cup (C, \emptyset) & R_{big}(D) = (A, \emptyset) \cup (B, \emptyset) \cup (C, \emptyset). \end{array}$$

It requires D to be drawn completely concurrently with other contours (D is drawn concurrently with the outside edges of Venn(3) in the diagram). In the semantics we would be asserting that the set D is equal to $A \cup B \cup C$ by the conditions for D .

Furthermore, sometimes an application requires us to draw a diagram which looks similar to another diagram. For example, after an application of a reasoning rule (which are applied at the abstract level and can change the diagram) one would wish to present the user with a conclusion diagram which looks similar to the premise diagram. This is a difficult task for rules such as add/remove contour using the zone-wise abstraction, but may be much more tractable using the building sequence abstraction. See fig. 12 for example.

Work on identifying commonality between drawn diagrams, in a zone-wise setting, has been done in [12]. It is worth considering if we can use the building sequence abstraction to construct an iterative drawing algorithm. It is not immediately apparent how to do this, but since this abstraction specifies where to place a contour in relation to other contours it is a natural area to investigate.

8 Conclusion

We have presented two new abstractions for Euler diagrams, and shown that they are equivalent to an existing abstraction. We expect that using different abstractions will be useful in different contexts. Many examples and suggestions for further investigation of such uses in areas such as identifying diagram properties, semantics and layout have been given throughout the paper. For instance, the containment condition abstraction often gives more readable semantics than the zone-wise abstraction, and the building sequence descriptions are necessary for the semantics of certain extensions of Euler diagrams. Each of the abstractions is likely to have advantages in describing properties of a diagram; for instance, the building sequence abstraction appears to easily indicate a case when a contour must be drawn completely concurrently with other contours in the diagram.

Potential application areas for these new abstractions are numerous. Some of these future work suggestions, include:

- Can we identify diagram properties like *nestedness*, *symmetry* or *drawability* (the existence of a drawing subject to some well-formedness conditions)?
- Can we use the building sequence abstraction to construct an iterative drawing algorithm?
- When using a building sequence abstraction for the purpose of a nice se-

mantic interpretation, can we choose an optimal ordering of contours?

- Can we view the zone-wise abstraction and the building sequence abstractions as opposite extremes of a spectrum, with intermediate abstractions merging diagrams made up of multiple contours (overlying one diagram with another)?

It is also possible to use these abstractions in conjunction. For instance, one could store information about which zones are present in a diagram (that is, use the zone-wise abstraction), and then describe diagrams created from this by giving the information of how another contour is added (using the building sequence description). This may enable existing tools to simply have extra features added which allows easy generation of diagrams which look similar to other diagrams.

These new abstractions present new ways of looking at and describing familiar diagrams. They have the potential to aid in many open questions about Euler diagrams.

References

- [1] R. De Chiara, U. Erra and V. Scarano. *VennFS: a Venn Diagram file manager*. In Proc. of Information Visualisation, pages 120–126, IEEE Computer Society, 2003.
- [2] R. De Chiara, U. Erra and V. Scarano. *A system for Virtual Directories using Euler Diagrams*. To appear in Proc. of Euler Diagrams, ENTCS, 2005.
- [3] A. Fish and J. Flower. *Investigating Reasoning with Constraint Diagrams*. In Proc. of Visual Languages and Formal Methods 2004, pages 53–67, IEEE, 2004.
- [4] A. Fish, J. Flower and J. Howse. *A reading algorithm for constraint diagrams*. In Proc. of Human Centric Computing, Languages and Environments 2003, pages 161–168, IEEE, 2003.
- [5] A. Fish, J. Flower and J. Howse. *The semantics of augmented constraint diagrams*. To appear in Journal of Visual Languages & Computing, 2005.
- [6] J. Flower and J. Howse. *Generating Euler diagrams*. In Proc. of Diagrams 2002, pages 61–75, LNAI 2317, Springer-Verlag, 2002.
- [7] E. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.
- [8] J. Howse, F. Molina, S-J. Shin and J. Taylor. *On diagram tokens and types*. In Proc. of Diagrams 2002, pages 76–90, LNAI 2317, Springer-Verlag, 2002.
- [9] J. Howse and S. Schumann. *Precise Visual Modelling*. To appear in Software & Systems Modelling, 2005.
- [10] J. Howse, G. Stapleton, J. Flower and J. Taylor. *Corresponding regions in Euler diagrams*. In Proc. of Diagrams 2002, pages 146–160, Springer-Verlag, 2002.
- [11] J. Flower and G. Stapleton. *Automated Theorem Proving with Spider Diagrams*. In CATS '04, pages 116–132, ENTCS.
- [12] P. Rodgers, P. Mutton and J. Flower. *Dynamic drawing of Euler diagrams*. In IEEE Symp. on Visual languages and human-centric computing, pages 147–156, IEEE, 2004.

- [13] S.-J. Shin. *The logical status of diagrams*. CUP, 1994.
- [14] N. Swoboda and G. Allwein. *Using DAG Transformations to Verify Euler/Venn Homogeneous and Euler/Venn FOL Heterogeneous Rules of Inference*. *Journal of Software & System Modeling*, pages 136–149, number 2, volume 3, 2004.
- [15] N. Swoboda. *Implementing Euler/Venn Reasoning Systems*. *Diagrammatic Representation and Reasoning*, eds. M. Anderson, B. Meyer and P. Olivier, pages 371–386, Springer-Verlag 2002.
- [16] A. Verroust and M-L. Viaud. *Ensuring the drawability of extended Euler diagrams for up to 8 sets*. In *Proc. of Diagrams 2004*, pages 128–141, Springer-Verlag, 2004.
- [17] D. Winterstein, A. Bundy, C. Gurr and M. Jamnik. *Using animation in diagrammatic theorem proving*. In *Proc. of Diagrams 2002*, pages 46–60, LNAI 2317, Springer-Verlag, 2002.