Electronic Notes in Theoretical Computer Science 72 No. 3 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume72.html 10 pages

Nesting in Euler Diagrams

J. Flower [2,1] J.Howse [3,1] J.Taylor [4,1]

School of Computing and Mathematical Sciences,
University of Brighton, Brighton, U.K.

Abstract

This paper outlines the notion of *nesting* in Euler diagrams, and how nesting affects the interpretation and construction of such diagrams. After setting up the necessary definitions for Euler diagrams at concrete syntax and abstract levels, the notion of nestedness is introduced at the concrete level, then an equivalent notion is given at the abstract level. The natural progression to the diagram semantics is explored. In the final sections, we describe how this work supports tool-building for diagrams, and how effective we might expect this support to be in terms of the proportion of nested diagrams.

1 Introduction

The distinction between concrete diagrams (drawn in the plane) and abstract diagrams (having just formal structure) was highlighted in [5]. The problem of converting an abstract Euler diagram into a concrete representative was addressed in [2]. This article extends work on Euler diagrams by incorporating the notion of a *nested* diagram (other choices of name could have been *disconnected* or *separated*). Section 2 begins with the necessary background notation and definitions for the rest of the paper.

The concept of nesting is most obvious, visually, for concrete diagram representatives. In section 3 we define the notion of nesting in a concrete Euler diagram and present an equivalent notion of nesting in an abstract diagram. The two notions of nesting are shown to be equivalent under the morphism from concrete to abstract diagrams (theorems 3.6,3.7).

Nesting in diagrams gives rise to different ways of writing down diagram semantics, and one example in section 4 points the way towards a "nested normal form" for diagram semantics.

[1] Partially supported by UK EPSRC grant GR/R63516
[2] Email:j.a.flower@brighton.ac.uk
[3] Email:john.howse@brighton.ac.uk
[4] Email:john.taylor@brighton.ac.uk

One application of this work is in diagram generation algorithms which are used to drive software tools. This application of the nested concept is discussed in section 5.

Finally, in section 6, some statistics are presented at the end of the paper to show how much leverage can be gained from making use of nesting in abstract diagrams.

2 The context: Euler diagrams

Work in this section is largely based upon work from [2]. Euler diagrams form the foundation of many diagrammatic notations such as Harel's hi-graphs, some UML notations [6] and constraint diagrams [4].

An *abstract Euler diagram* comprises a set whose elements are called *contours* and a set of *zones* which are subsets of the contour set.

Definition 2.1 An *abstract (Euler) diagram* is a pair: $d = \langle \mathcal{C}(d), \mathcal{Z}(d) \rangle$ where

(i) $\mathcal{C}(d)$ is a finite set whose members are called *contours*

(ii) $\emptyset \in \mathcal{Z}(d) \subseteq \mathcal{P}\mathcal{C}(d)$ is the set of *zones* of d, so $z \in \mathcal{Z}(d)$ is $z \subseteq \mathcal{C}(d)$

(iii) $\bigcup_{z \in \mathcal{Z}(d)} z = \mathcal{C}(d)$

The set of abstract diagrams is denoted $\mathcal{D}$.

Example 2.2 [Abstract diagram] This abstract diagram has three contours and five zones: $\langle \{a, b, c\}, \{\{\}, \{a\}, \{a, b\}, \{b\}, \{c\}\} \rangle \in \mathcal{D}$.

A *concrete Euler diagram* is a set of labelled *contours* (simple closed curves) in the plane, each with a unique label. A *zone* is a connected component of the complement of the contour set. The zone corresponding to the empty set is the component outside all contours of the diagram.

Definition 2.3 A *concrete (Euler) diagram* is a triple $\hat{d} = \langle \widehat{\mathcal{L}}(\hat{d}), \widehat{\mathcal{C}}(\hat{d}), \widehat{\mathcal{Z}}(\hat{d}) \rangle$ whose components are defined as follows:

(i) $\widehat{\mathcal{C}}(\hat{d})$ is a finite set of simple closed curves, *contours*, in the plane $\mathbb{R}^2$. Each contour has a label from the set $\widehat{\mathcal{L}}(\hat{d})$, so that the labelling mapping $\widehat{\mathcal{C}}(\hat{d}) \to \widehat{\mathcal{L}}(\hat{d})$ is a bijection.

(ii) contours meet transversely.

(iii) each component $\hat{z} \in \mathbb{R}^2 - \bigcup_{\hat{c} \in \widehat{\mathcal{C}}(\hat{d})} \hat{c}$ is a *zone*.

(iv) each zone is uniquely identified by a set of contours $\widehat{\mathcal{Z}}(\hat{d}) \subset \widehat{\mathcal{C}}(\hat{d})$ with $\hat{z} = \bigcap_{\hat{c} \in \widehat{\mathcal{Z}}(\hat{d})} \mathit{interior}\,(\hat{c}) \cap \bigcap_{\hat{c} \in \widehat{\mathcal{C}}(\hat{d}) - \widehat{\mathcal{Z}}(\hat{d})} \mathit{exterior}\,(\hat{c})$.

The set of concrete diagrams is denoted $\widehat{\mathcal{D}}$.

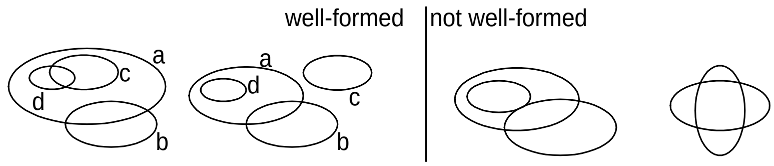

Fig. 1. Well-formed and not well-formed concrete diagrams

Example 2.4 [A concrete diagram] Let $\hat{d}$ be the first concrete diagram given in figure 1. $\widehat{\mathcal{C}}(\hat{d})$ has four elements (the four contours shown) $\widehat{\mathcal{L}}(\hat{d}) = \{a, b, c, d\}$ and $\widehat{\mathcal{Z}}(\hat{d})$ has seven elements, uniquely determined by the label sets $\{\}$, $\{a\}$, $\{b\}$, $\{a, b\}$, $\{a, c\}$, $\{a, d\}$ and $\{a, c, d\}$.

The rules about transverse crossings and connectedness of zones are the chosen well-formedness rules for this paper. Figure 1 shows two well-formed concrete diagrams and two which are not well-formed. In future work, we plan to accommodate different definitions of "well-formed" concrete diagrams.

Definition 2.5 The mapping $ab : \widehat{\mathcal{D}} \to \mathcal{D}$ ("ab" for "abstractify") forgets positioning of the contours. It is defined by

$$ab : \langle \widehat{\mathcal{L}}(\hat{d}), \widehat{\mathcal{C}}(\hat{d}), \widehat{\mathcal{Z}}(\hat{d}) \rangle \mapsto \langle \widehat{\mathcal{L}}(\hat{d}), \{\widehat{\mathcal{L}}(\hat{z}) : \hat{z} \in \widehat{\mathcal{Z}}(\hat{d})\} \rangle$$

Example 2.6 Let $\hat{d}$ be the concrete diagram given in figure 1. Then its abstract diagram has:

$$\mathcal{C}(ab(\hat{d})) = \{a, b, c, d\}$$
$$\mathcal{Z}(ab(\hat{d})) = \{\{\}, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{a, d\}, \{a, c, d\}\}$$

Definition 2.7 A concrete diagram $\hat{d}$ *represents* or *complies with* an abstract diagram d if and only if $d = ab(\hat{d})$. An abstract diagram which has a compliant concrete representation is *drawable*.

Definition 2.8 An *abstract labelled graph* is a triple $\langle \mathcal{L}(G), \mathcal{V}(G), \mathcal{E}(G) \rangle$ where the components are defined as follows.

(i) $\mathcal{L}(G)$ is a set of labels
(ii) $\mathcal{V}(G)$ is a set of vertices. Each vertex $\hat{v}$ is labelled with $\mathcal{L}(v) \subseteq \mathcal{L}(G)$
(iii) $\mathcal{E}(G)$ is a set of edges. Each edge $\hat{e}$ is a pair of vertices in $\mathcal{V}(G)$, where the vertex labels must have a singleton symmetric difference (one set exceeds the other by a single additional element). The label which distinguishes the end vertices can be used to label the edge

The set of abstract labelled graphs is denoted $\mathcal{LG}$.

Definition 2.9 The map $dual : \mathcal{D} \to \mathcal{LG}$ is defined by

$$\langle \mathcal{C}(d), \mathcal{Z}(d) \rangle \mapsto \langle \mathcal{C}(d), \mathcal{Z}(d), \mathcal{E}(G) \rangle$$

where the edges include all possible $e = (v_1, v_2)$ where v_1 and v_2 have singleton symmetric difference.

This definition of the dual graph of an abstract diagram extends to a definition of the dual of a concrete diagram.

Definition 2.10 The map $dual : \widehat{\mathcal{D}} \to \mathcal{LG}$ is defined by

$$\langle \widehat{\mathcal{L}}(\hat{d}), \widehat{\mathcal{C}}(\hat{d}), \widehat{\mathcal{Z}}(\hat{d}) \rangle \mapsto dual(ab(\langle \widehat{\mathcal{L}}(\hat{d}), \widehat{\mathcal{C}}(\hat{d}), \widehat{\mathcal{Z}}(\hat{d}) \rangle)).$$

Note that this dual graph is not a topological construction. It is possible for two zones which are not topologically adjacent in $\hat{d}$ to correspond to adjacent vertices in the dual. However, if two zones are adjacent in $\hat{d}$ then the vertices are bound to be adjacent in the dual.

Definition 2.11 [The connectivity conditions] An abstract labelled graph $\langle \mathcal{L}(G), \mathcal{V}(G), \mathcal{E}(G) \rangle$ satisfies the *connectivity conditions* if it is connected and, for all labels $l \in \mathcal{L}(G)$, the subgraphs $G^{+}(l)$ generated by vertices whose labels include l, and $G^{-}(l)$ generated by vertices whose labels exclude l are connected.

Theorem 2.12 (The connectivity theorem) *Let $\hat{d}$ be a concrete diagram. Then $dual(\hat{d})$ satisfies the connectivity conditions. Hence, if abstract diagram d is drawable then $dual(d)$ satisfies the connectivity conditions.*

3 Defining atomic and nested diagrams

We want to be able to identify nesting within a given diagram.

3.1 Nesting in concrete diagrams

Definition 3.1 $\hat{d}$ is a *nested* concrete diagram if it can be split into at least two sub-diagrams $\hat{d}_1, ..., \hat{d}_n$ where a contour in $\widehat{\mathcal{C}}(\hat{d}_i)$ never crosses any contour in $\widehat{\mathcal{C}}(\hat{d}_j)$ in $\hat{d}$ (i and j distinct). A diagram which is not nested is called *atomic*.

Proposition 3.2 *A concrete Euler diagram $\hat{d}$ is* nested *if there exists a simple closed curve γ which doesn't meet any of the contours of $\hat{d}$, and splits the plane into two parts, both including at least one contour of $\hat{d}$.*

Proposition 3.3 *A concrete Euler diagram $\hat{d}$ is atomic if the union of its contours is a connected subset of the plane.*

Proposition 3.4 *A concrete Euler diagram $\hat{d}$ is nested if it splits into sub-diagrams $\hat{d}_1$ and $\hat{d}_2$ and there is a zone $\hat{z} \in \widehat{\mathcal{Z}}(\hat{d}_1)$ such that all contours in $\widehat{\mathcal{C}}(\hat{d}_2)$ are contained within $\hat{z}$.*

The following figure illustrates three equivalent approaches to nested concrete diagrams.

3.2 Nesting in abstract diagrams

The notions of crossing contours, topological connectedness or topological containment are unavailable to us when we define the notion nesting in the abstract case.

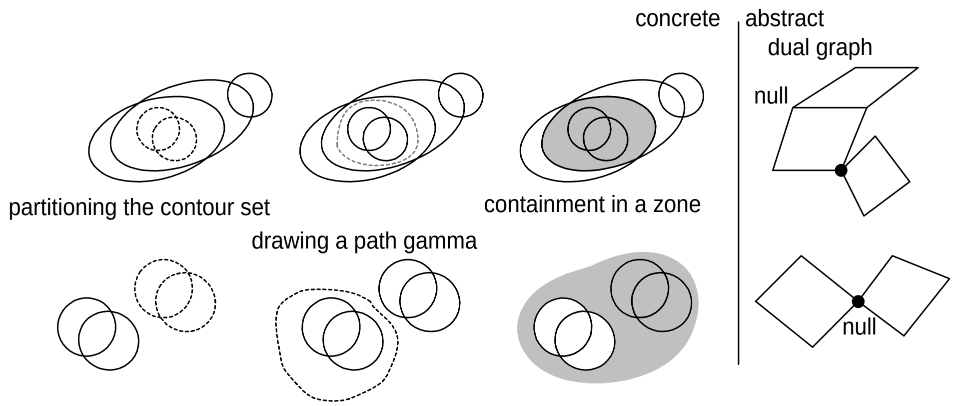

Fig. 2. Criteria for nesting in a Euler diagrams

Definition 3.5 An abstract Euler diagram d is *nested* if there exists a cut vertex of $dual(d)$. A diagram which is not nested is called *atomic.*

3.3 Consistency between abstract and concrete nesting

Figure 2 illustrates the relationship between nested concrete diagrams and the presence of a (highlighted) cut vertex in the dual graph of the abstract diagram.

We will show two results: if a concrete diagram is nested, then its abstract diagram is also nested, and if an abstract diagram is nested, then all concrete representations of it will be nested.

Theorem 3.6 *Given an abstract diagram d which is nested, let $\hat{d}$ be any well-formed concrete representation. Then $\hat{d}$ must be nested.*

Proof. Let cv be a cutvertex of $dual(d)$. The dual has $n > 1$ subgraphs $S_1, ..., S_n$ obtained by removing the cutvertex and replacing it back into each component in turn. By construction of the S_i we have, for distinct i and j, $\mathcal{V}(S_i) \cap \mathcal{V}(S_j) = \{cv\}$.

For each $1 \leq i \leq n$, let $C_i \subseteq \mathcal{C}(d)$ be the set of contour labels appearing as edge labels of S_i. Every contour in $\mathcal{C}(d)$ appears in one of these sets. We will show that the sets $C_1, ..., C_n$ are disjoint, by contradiction.

Let $c \in C_i \cap C_j$. There are edges of e_i in S_i, $e_j$in S_j which are both labelled c. Let e_i have ends v_i, w_i and e_j have ends v_j, w_j where $c \in v_i, v_j$ and $c \notin w_i, w_j$.

Assume first that $c \notin cv$ so that $v_i, v_j \in S_i - cv$, $S_j - cv$. Any path from v_i to v_j must pass through the cutvertex cv, but the drawability of d tells us that the dual satisfies the connectivity conditions, including the fact that the subgraph restricted to those vertices which contain c is connected. This is a contradiction.

If, on the other hand, $c \in cv$ then follow a similar line of argument using w_i and w_j, and the contradiction comes from the connectivity condition that the subgraph of S built from vertices which exclude c is connected.

Thus $\hat{C}_1, ..., \hat{C}_n$ partition the contour set. It will be enough to show that a contour $\hat{c}_i \in \hat{C}_i$ can never cross a contour $\hat{c}_j \in \hat{C}_j$. If $\hat{c}_i$ meets $\hat{c}_j$ and the diagram has transverse crossings, then there must be zones z, $z \cup \{c_i\}$,$z \cup \{c_j\}$ and $z \cup \{c_i, c_j\}$ in the abstract diagram d. The dual edges between z and $z \cup \{c_i\}$ and between $z \cup \{c_j\}$ and $z \cup \{c_i, c_j\}$ lie in subgraph S_i, and the dual edges between z and $z \cup \{c_j\}$ and between $z \cup \{c_i\}$ and $z \cup \{c_i, c_j\}$ lie in subgraph S_j. But the subgraphs S_i and S_j share only one vertex, the cutvertex cv. This is a contradiction, so the partitioning of the contour set shows that the concrete diagram $\hat{d}$ is nested. $\square$

The second of these results only holds in the presence of the well-formedness rules. (For example: $A \subseteq B$ can be drawn non-nested if we allow tangential contours.)

Theorem 3.7 *Given a concrete diagram $\hat{d}$ which is nested, then its abstract diagram $ab(\hat{d})$ is nested.*

Recall that topological adjacency implies dual adjacency but the converse does not hold.

Proof. Let $\hat{d}$ be nested, and let $\hat{\mathcal{C}}_2$ be the contours in an innermost connected component of the union of contours of $\hat{d}$ (see 3.3). Let $\hat{\mathcal{C}}_1$ be $\widehat{\mathcal{C}}(\hat{d}) - \hat{\mathcal{C}}_2$. Think of contours in $\hat{\mathcal{C}}_2$ as being "inside" some simple closed curve, γ and contours in $\hat{\mathcal{C}}_1$ being outside γ (see 3.2).

This enables us to partition $\widehat{\mathcal{Z}}(\hat{d}) = \hat{\mathcal{Z}}_{in} \sqcup \{\hat{z_\gamma}\} \sqcup \hat{\mathcal{Z}}_{out}$ where the zones in $\hat{\mathcal{Z}}_{in}$ have boundaries made up of contours from $\hat{C}_2$, the zones in $\hat{\mathcal{Z}}_{out}$ have boundaries made up of contours from $\hat{C}_1$ and the zone which has a boundary meeting both contours from $\hat{C}_1$ and $\hat{C}_2$ is called $\hat{z}_\gamma$.

$$\begin{array}{rcl} \hat{z} \in \hat{\mathcal{Z}}_{in} \wedge \hat{c} \in \hat{\mathcal{C}}_1 & \Rightarrow & \partial\hat{z} \cap \hat{c} = \emptyset \\ \hat{z} \in \hat{\mathcal{Z}}_{out} \wedge \hat{c} \in \hat{\mathcal{C}}_2 & \Rightarrow & \partial\hat{z} \cap \hat{c} = \emptyset \\ \exists \hat{c}_1 \in \hat{\mathcal{C}}_1 \wedge \hat{c}_2 \in \hat{\mathcal{C}}_2 & \text{such that} & \partial\hat{z_\gamma} \cap \hat{c}_1 \neq \emptyset \wedge \partial\hat{z_\gamma} \cap \hat{c}_2 \neq \emptyset \end{array}$$

Given any zone $\hat{z} \in \hat{\mathcal{Z}}_{in}$, there is a path α inside γ from a point in $\hat{z}$ to a point in $\hat{z}_\gamma$. The symmetric difference between the abstract zones $z \in \mathcal{Z}_{in}$ and z_γ consists of contours in C_2. The partitioning of concrete zones induces a partitioning of abstract zones with symmetric difference properties (use $\triangle$ to denote set symmetric difference).

$$\begin{array}{c} \mathcal{Z}(d) = \mathcal{Z}_{in} \sqcup \{z_\gamma\} \sqcup \mathcal{Z}_{out} \\ z \in \mathcal{Z}_{in} \quad \Rightarrow \quad z \triangle z_\gamma \subseteq \mathcal{C}_1 \wedge z \triangle z_\gamma \neq \emptyset \\ z \in \mathcal{Z}_{out} \quad \Rightarrow \quad z \triangle z_\gamma \subseteq \mathcal{C}_2 \wedge z \triangle z_\gamma \neq \emptyset \\ z_1 \in \mathcal{Z}_{in} \wedge z_2 \in \mathcal{Z}_{out} \quad \Rightarrow \quad z_1 \triangle z_2 = z_1 \triangle z_\gamma \sqcup z_2 \triangle z_\gamma \end{array}$$

The symmetric difference of abstract zones in sets $\mathcal{Z}_{in}$ and $\mathcal{Z}_{out}$ contains at least two elements, so no two are adjacent. The zone z_γ acts as a pathway in the dual graph from $\mathcal{Z}_{in}$ to $\mathcal{Z}_{out}$, and is a cut vertex of the dual graph. $\square$

4 The semantics of nested diagrams

A model for Euler diagrams assigns sets to contours. Given a diagram, some models will be valid and others invalid, according to the indicated set containment and disjointness rules.

Definition 4.1 A *model* for diagram d is an assignment $\psi : \mathcal{C}(d) \longrightarrow \mathcal{P}(\mathcal{U})$, where U is some *universal set*. Such a mapping ψ extends to a set assignment to zones:

$$\psi : \mathcal{Z}(d) \longrightarrow Set; z \mapsto \bigcap_{c \in z} \psi(c) \cap \bigcap_{c \notin z} \overline{\psi(c)}$$

The overline used here means set complement in the context of the universal set U. The extension of ψ to zones ensures that two different zones correspond to disjoint sets.

Definition 4.2 A mapping ψ is *valid* for diagram d if the extension of ψ to zones satisfies the *plane tiling condition*: that the sets represented by all zones union to make up the whole of the universal set.

$$\bigcup_{z \in \mathcal{Z}(d)} \psi(z) = U$$

Example 4.3 [A valid model] Consider the first diagram given in Figure 1. Define a mapping from contours to sets and extend it to a mapping from zones to sets. Take a universal set $U = \{1, 2, 3, 4, 5\}$

$$\begin{gathered}\psi : \{a, b, c, d\} \rightarrow Set; \mathcal{Z}(d) \rightarrow Set \\ a \mapsto \{1, 2, 3, 4\}; b \mapsto \{4, 5\}; c \mapsto \{1, 2\}; d \mapsto \{2, 3\} \\ \{\} \mapsto \{\}; \{a\} \mapsto \{\}; \{b\} \mapsto \{5\}; \{a, b\} \mapsto \{4\} \\ \{a, c\} \mapsto \{1\}; \{a, d\} \mapsto \{3\}; \{a, c, d\} \mapsto \{2\};\end{gathered}$$

Example 4.4 [An invalid model] Consider the first diagram given in Figure 1. Define a mapping from contours to sets and extend it to a mapping from zones to sets. Take a universal set $U = \{1, 2, 3, 4, 5\}$

$$\begin{gathered}\psi : \{a, b, c, d\} \rightarrow Set; \mathcal{Z}(d) \rightarrow Set \\ a \mapsto \{1, 2, 3, 4\}; b \mapsto \{3, 4, 5\}; c \mapsto \{1, 2\}; d \mapsto \{2, 3\} \\ \{\} \mapsto \{\}; \{a\} \mapsto \{\}; \{b\} \mapsto \{\}; \{a, b\} \mapsto \{4\} \\ \{a, c\} \mapsto \{1\}; \{a, d\} \mapsto \{\}; \{a, c, d\} \mapsto \{2\};\end{gathered}$$

The zones only combine to make $\{1, 2, 4\} \neq U$ and the plane tiling condition is not satisfied. The model is not valid.

The semantics of an abstract Euler diagram are encapsulated in the plane tiling condition. This could be taken as a *normal form* for the semantics.

Example 4.5 Consider the first diagram given in Figure 1. The normal form of its semantics is

$$(\overline{A} \cap \overline{B} \cap \overline{C} \cap \overline{D}) \cup (A \cap \overline{B} \cap \overline{C} \cap \overline{D}) \cup (\overline{A} \cap B \cap \overline{C} \cap \overline{D}) \cup$$

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

If a diagram is nested, then this normal form for the diagram semantics can be simplified to a *nested normal form*:

Example 4.6 Consider the first diagram given in Figure 1. The nested normal form of its semantics is as follows. The contours partition into $\{a, b\} \sqcup \{c, d\}$. This partition gives a nested normal form:

$$\Big((\overline{A} \cap \overline{B}) \cup (A \cap \overline{B}) \cup (\overline{A} \cap B) \cup (A \cap B) = U\Big) \wedge \Big(C, D \subseteq A \cap \overline{B}\Big)$$
$$\wedge\Big((\overline{C} \cap \overline{D} \cap A \cap \overline{B}) \cup (C \cap \overline{D}) \cup (\overline{C} \cap D) \cup (C \cap D) = A \cap \overline{B}\Big)$$

The first part of this expression is the semantics for the containing subdiagram. The second part expresses a containment relationship about the contours inserted into the containing diagram. The third part is similar to the semantics for the inner subdiagram, with the universal set taken to be the set of the zone into which the inner diagram is inserted, and the "outside" zone replaced with the intermediate zone between the inserted diagram and the containing diagram.

This nested normal form simplifies further to give

$$A, B, C, D \subseteq U \wedge C \subseteq A \cap \overline{B} \wedge D \subseteq A \cap \overline{B}$$

This final simplification is evident from the nested normal form, by noticing that, for example, $(\overline{A} \cap \overline{B}) \cup (A \cap \overline{B}) \cup (\overline{A} \cap B) \cup (A \cap B) = U$ reduces to $A \subseteq U$ and $B \subseteq U$. The more concise expression was disguised in the first normal form.

There remains a question: is there such a concept as a "nested proposition", which, expressed as as Euler diagram, would yield a nested diagram? Is there a "nested normal form" for propositions?

5 Constructing atomic and nested diagrams

An algorithm has been devised and implemented to create drawings of drawable Euler diagrams [2]. To enhance the efficiency of the algorithm and the readability of its output, we describe here an approach to make use of nesting in the abstract Euler diagrams.

Given an abstract Euler diagram d whose dual has a cut vertex, there are sub-graphs $S_1, ..., S_n$ of the $dual(d)$ obtainable by removing the cut vertex and replacing it, in turn, to each component. Without loss of generality, S_1 contains a vertex labelled by the empty set. (Possibly other subgraphs do too, if the cutvertex is the null vertex).

Draw a concrete representation for the diagram whose dual is S_1, and add to it places to insert $n-1$ other diagrams inside the zone corresponding to the cut-vertex. Think of the diagram as a *template*, as in [3].

The other $n-1$ subgraphs of the dual have vertex labels which are all supersets of the cut vertex. Replace each abstract zone z with $z - cv$. Then each of subgraph can be represented by a concrete diagram. These concrete diagrams are inserted into the template, to make up $\hat{d}$.

Figure 3 shows an example where the dual has two cut vertices $\{\}$ and $\{a\}$. The containing diagram is used as template with insertion into two different zones. The subgraph in zone $\{a\}$ has vertices are $\{a\}, \{a, c\}$, which are all reduced to $\{\}, \{c\}$ before constructing the inner concrete diagram. The

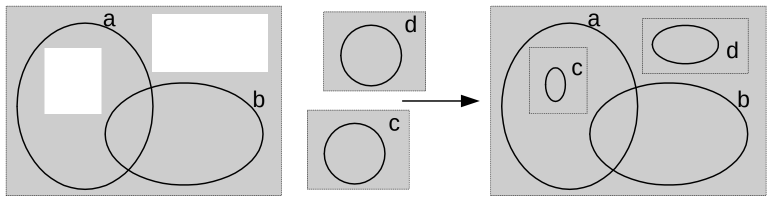

Fig. 3. Constructing concrete nested diagrams using templates

sub-diagrams constructed for insertion into the template correspond to part of the semantic expression in nested normal form. The nested normal form essentially combines information from the template diagram, information from the sub-diagrams and information about the inclusion of the subdiagram into the template.

6 Counting atomic and nested diagrams

To see the leverage gained by using the nesting concept in semantics or drawing problems, consider how the numbers of abstract diagrams grow with the number of contours. The following table shows how many well-formed diagrams there are with a given number of contours (by row) and a given number of zones (by column). The number of diagrams is seen to grow quickly, but the number of atomic diagrams, shown in brackets, grows much less quickly. Drawing nested diagrams using templates as described in the previous section can handle the vast majority of diagrams, leaving just a few atomic examples to be drawn without using a template.

	3z	4z	5z	6z	7z	8z
2c	2	1(1)				
3c		4	4	3(3)	3(3)	1(1)
4c			9	15	20	30(14)
5c				20	50	101

The figure shows some of these diagrams in concrete form. The first column shows the atomic examples and the later columns show nested examples. Studying these diagrams may give insight into methods for counting the diagrams - use tree-counting (eg in [1]) or group symmetries, for example.

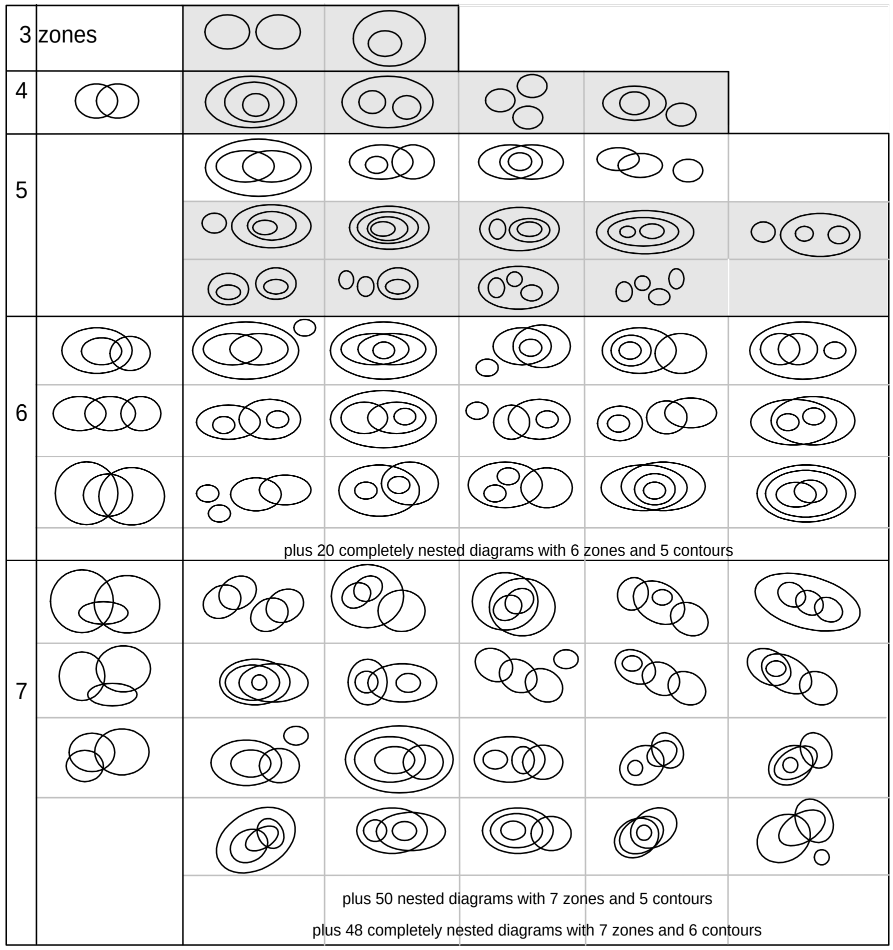

Fig. 4. Examples of small Euler diagrams

References

[1] Cayley, A. "On the analytical forms called trees", American Journal of Mathematics 4, 1881, p.266-268.

[2] Flower, J. and Howse, J., "Generating Euler Diagrams", Diagrammatic Representation and Inference, Proc. Diagrams 2002, LNAI 2317, Springer Verlag, 2002, p.61-75.

[3] Flower J., Howse, J. and Taylor, J.,"A framework for modelling with heterogeneous notations", Proc IEEE Symp on Human-Centric computing, 2002 (to appear).

[4] Gil, J., Howse, J., Kent, S. and Gil, J.,"Towards a formalisation of constraint diagrams", Proc IEEE Symp on Human-Centric computing, 2001, p.72-79.

[5] Howse, J., Molina, F, Shin, S.J. and Taylor, J., "On Diagram tokens and types", Diagrammatic Representation and Inference, Proc. Diagrams 2002, LNAI 2317, Springer Verlag, 2002, p.146-160.

[6] Object Management Group. "UML Specification, Version 1.3:" www.omg.org.