

# Incorporating Negation into Visual Logics: A Case Study Using Euler Diagrams

Gem Stapleton  
Visual Modelling Group  
University of Brighton, Brighton, UK  
g.e.stapleton@brighton.ac.uk

Judith Masthoff  
University of Aberdeen, Aberdeen, UK  
jmasthoff@csd.abdn.ac.uk

## Abstract

Many diagrammatic logics based on Euler diagrams have been defined with the aim of making precise reasoning easier for people. In such logics, it is unusual to find the negation operator ( $\neg$ ) included. This is likely to impact the usability of the logics when users try to make statements that are naturally phrased using the  $\neg$  operator. Furthermore, if one wishes to use semantic tableaux methods for the purposes of establishing entailment then including  $\neg$  is essential. Thus, there are good reasons for extending existing diagrammatic logics to include  $\neg$  explicitly. In this paper, we take Euler diagrams and extend the notation to include the  $\neg$  operator, as well as  $\vee$  and  $\wedge$ . Various expressiveness results for the logic are established. We present a sound and complete set of reasoning rules for the logic, drawing parallels with existing completeness proof strategies and highlighting differences that arise due to including negation.

## 1 Introduction

Recently, a variety of diagrammatic logics have been developed with an aim of making reasoning tasks easier for people. Many such logics are based on either Euler or Venn diagrams; see, for example [2, 4, 5, 9, 11, 14]. Whilst these logics mostly include some logical connectives (usually  $\vee$  and occasionally  $\wedge$ ), none of the sound and complete logics developed have included both  $\vee$  and  $\wedge$  along with  $\neg$ . Indeed, none of the systems just referenced include  $\neg$ ; those in [4, 15] do not incorporate logical connectives at all. In some cases extending such logics to include  $\neg$  (or  $\wedge$ ) does not lead to an increase in expressive power. Even when expressiveness is not increased, it is likely that excluding  $\neg$  and  $\wedge$  will impact usability because to make a negated statement without using  $\neg$ , for example, one has to find an equivalent non-negated statement. Furthermore, if we wish to utilize semantic tableaux methods for establishing semantic entailment (as in [7]) then incorporating explicit

negation is essential.

We conjecture that this typical exclusion of explicit negation is to simplify the systems from a formal perspective; for example, fewer reasoning rules will need to be defined which makes the proof of soundness easier to derive. Moreover, the completeness strategies employed are also more straightforward when  $\neg$  is not included. Whilst  $\{\vee, \neg\}$  and  $\{\wedge, \neg\}$  are both minimal sets of operators, our own experience suggests that all three operators are useful when making logical statements. Thus it is important for us to have a full understanding of the effects that including all three operators have on expressiveness and on the completeness proof strategies available to us.

As a case study, we extend Euler diagrams to include  $\neg$ ,  $\vee$  and  $\wedge$  and highlight various effects that this extension brings with it. Euler diagrams are an ideal choice for such a case study, since all of the logics referenced above extend this language. The strategy we take to prove completeness is likely to adapt to other visual logics that are extended to include all of these connectives because the completeness proof strategies used when  $\neg$  is not included in these systems are generally similar.

## 2 Euler Diagrams: Syntax and Semantics

Here, we will briefly overview the syntax and semantics of Euler diagrams; for a formalization of the syntax see [12] and for the semantics see [13], adapting to the Euler diagram case.

There are two *unitary* Euler diagrams in figure 1. The diagram  $d_1$  contains three *contours*; these are the closed curves labelled  $A$ ,  $B$  and  $C$ . Contours represent sets and their spatial relationship is used to make statements about containment and disjointness of sets. So, in  $d_1$  the contours assert that the sets  $A$  and  $B$  are disjoint because the contours do not overlap in any way; similarly  $A \cap C = \emptyset$ . The placement of  $C$  inside  $B$  expresses  $C \subseteq B$ . Shading is used to assert the emptiness of sets. Thus,  $d_1$  also expresses  $B - C = \emptyset$ . The contours give rise to *regions* in the plane. A region is a set of *zones*, which are maximal sets of points

in the plane that can be described as being inside certain contours and outside the rest of the contours. The diagram  $d_1$  contains four zones of which one is shaded and can be described as inside  $B$  but outside both  $A$  and  $C$ .

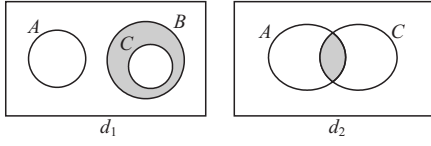


Figure 1. Two Euler diagrams.

Formally, we define a zone to be an ordered pair, (*inside*, *outside*), which forms a two-way partition of the contour label set; the shaded zone in  $d_1$  is  $(\{B\}, \{A, C\})$ . We can now talk about zones that can be described by such a pair but which do not correspond to regions present in a diagram. For example  $(\{A, B\}, \{C\})$  is a zone which is not present in  $d_1$  (no zone is inside both  $A$  and  $B$  but outside  $C$ ). Such zones are said to be *missing* from the diagram;  $d_1$ , therefore, has four missing zones whereas  $d_2$  has no missing zones. An Euler diagram with no missing zones is called a *Venn diagram*.

Informally, a *zonal region* can be thought of as a region that becomes a zone when contours are removed. In figure 1, the two zones inside  $A$  in  $d_2$  become a zone when  $C$  is removed; in total  $d_2$  has nine zonal regions. We define zonal regions in a similar way to zones: a zonal region in a diagram  $d$  is a set of zones that are described by a containing label set  $P$  and an excluding label set  $Q$ ; more precisely, given any such disjoint sets  $P$  and  $Q$ , the set of zones  $\{(inside, outside) : P \subseteq inside \wedge Q \subseteq outside\}$  is a zonal region. Whilst the two sets  $P$  and  $Q$  are disjoint, they need not form a partition of  $d$ 's label set (unlike the sets used to describe a zone).

Unitary diagrams form the building blocks of *compound* diagrams. If  $d_1$  and  $d_2$  are Euler diagrams then so are  $\neg d_1$ ,  $(d_1 \vee d_2)$  and  $(d_1 \wedge d_2)$ . For simplicity, we adopt the usual convention of omitting brackets where no ambiguity arises. At the drawn diagram level, we denote  $\neg d_1$  by  $\overline{d_1}$ ,  $(d_1 \vee d_2)$  by  $\boxed{d_1 - d_2}$  and  $(d_1 \wedge d_2)$  by  $\boxed{d_1 d_2}$  (i.e. by juxtaposition), where the rectangles act as brackets.

At the semantic level, an *interpretation* is a universal set,  $U$ , together with an assignment of a subset of  $U$  to each contour (strictly, to contour labels) which is extended to interpret zones and regions. A zone,  $(a, b)$ , represents the set  $\bigcap_{l \in a} set(l) \cap \bigcap_{l \in b} (U - set(l))$  where  $set(l)$  is the set assigned to label  $l$ . A region in a unitary Euler diagram,  $d$ , represents the set which is the union of the sets represented by the region's constituent zones. Briefly, we say that an interpretation is a *model* for  $d$  if all of the zones which are shaded in  $d$  or missing from  $d$  represent the empty set. The semantics

extend to compound diagrams in the obvious way. A diagram,  $d_2$ , is a *logical consequence* of diagram  $d_1$ , denoted  $d_1 \models d_2$ , if all of the models for  $d_1$  are also models for  $d_2$ .

### 3 Expressiveness

In this section, we will provide a series of results concerning the expressiveness of the Euler diagram system. Our first observation is that including  $\neg$  in the Euler diagram case impacts expressiveness.

**Theorem 3.1.** *Removing  $\neg$  from the Euler diagram logic reduces expressiveness.*

**Proof (Sketch)** Every satisfiable non-negated diagram is satisfied by the empty model (i.e.  $U = \emptyset$ ). There are negated diagrams that have only non-empty models.  $\square$

Our second observation is that including  $\wedge$  does not impact expressiveness, regardless of whether  $\neg$  is included.

**Theorem 3.2.** *Removing  $\neg$  and  $\wedge$  from the Euler diagram logic is equivalent in terms of expressiveness to removing only  $\neg$  from the logic.*

To prove the above theorem, we can assume, without loss of generality, that our diagrams are in disjunctive normal form. We illustrate the proof strategy by the example in figure 2. The diagram  $d_1 \wedge d_2$  can be replaced by the unitary diagram  $d_5$ . First, we make the contour label sets of  $d_1$  and  $d_2$  identical, giving  $d_3$  and  $d_4$  respectively. It is then easy to see how  $d_3 \wedge d_4$  gives rise to  $d_5$ : take the union of the shaded zones in  $d_3$  and  $d_4$  to give the shaded zones in  $d_5$ ; in general, the shaded zones in  $d_5$  will be the union of the shaded and missing zones. This process can be used to turn the conjunction of any pair of unitary diagrams into a single unitary diagram.

We cannot, however, necessarily replace the disjunction of two unitary diagrams by a single unitary diagram. For example, taking  $d_3$  and  $d_4$ , figure 2, in disjunction (as opposed to the conjunction displayed in the figure) provides a compound diagram that has no unitary counterpart, justified as follows. The diagram  $d_3 \vee d_4$  has only models where  $A - B = \emptyset$  or  $C - A = \emptyset$ . Thus, a model for  $d_3 \vee d_4$  can allow  $A - B$  to be non-empty, provided  $C - A$  is empty. Therefore, any unitary diagram,  $d$ , semantically equivalent to  $d_3 \vee d_4$  must have a model where  $A - B$  is non-empty. Similarly,  $d$  must also allow  $C - A$  to be non-empty. It can be shown that any such  $d$  also has a model where both  $A - B$  and  $C - A$  are non-empty, which is necessarily not a model for  $d_3 \vee d_4$ , showing that no unitary diagram is semantically equivalent to  $d_3 \vee d_4$ .

Next, we observe that a negated unitary diagram is never equivalent to a unitary diagram.

**Theorem 3.3.** *Let  $d_1$  be a unitary diagram. Then  $\neg d_1$  is not semantically equivalent to any unitary diagram.*

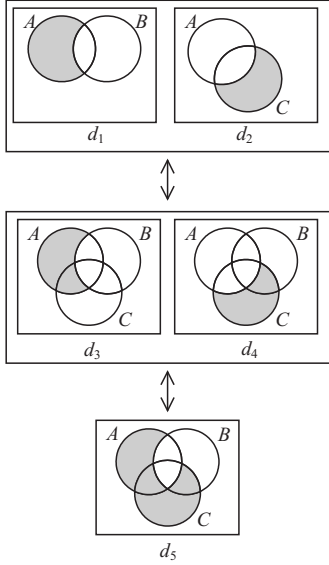


Figure 2. Removing conjunction.

**Proof (Sketch)** Every unitary diagram is satisfied by the empty model (i.e  $U = \emptyset$ ). The negation of  $d_1$ , therefore, is not satisfied by the empty model. Thus  $\neg d_1$  is not semantically equivalent to any unitary diagram.  $\square$

This result can be extended to show that  $\neg d_1$  is not equivalent to any compound diagram that does not include negation. In fact, given any diagram,  $D$ , that involves negation, there only exists a semantically equivalent diagram that does not involve negation when the standard propositional logic equivalences (such as involution and De Morgan’s Laws) can be used to remove the negation from  $D$ .

We can establish the expressiveness of the Euler diagram logic by comparing with other logics. First, we observe that the diagram  $d_2$  in figure 1 is equivalent to the Monadic First Order Logic (MFOL) sentence  $\forall x \neg(A(x) \wedge C(x))$ . In MFOL all of the predicate symbols are unary (they correspond to contour labels). It is easy to translate Euler diagrams into MFOL sentences. It is also possible, but not so easy, to convert MFOL sentences into Euler diagrams. As a straightforward example, the MFOL sentence  $\exists x A(x) \wedge \forall x B(x)$  is equivalent to the diagram in figure 3.

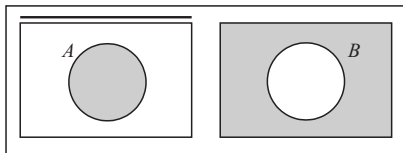


Figure 3. Comparing with MFOL.

**Theorem 3.4.** Euler diagrams are equivalent in expressive

power to MFOL.

It follows that Euler diagrams are less expressive than spider diagrams, which are equivalent to MFOL with equality [13].

In seminal work, Shin introduced a variant of Venn-Peirce diagrams [8] called Venn-II [9] which extends Venn diagrams by including  $\otimes$ -sequences to express the non-emptiness of a set. For example, figure 4 shows a Venn-II diagram that expresses either  $A - B \neq \emptyset$  and  $A \cap B = \emptyset$  or  $C \neq \emptyset$ .

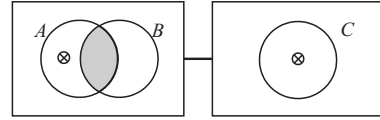


Figure 4. A Venn-II diagram.

In Venn-II, the only explicit logical connective is disjunction; there are no explicit negation or conjunction operators. Shin proved that Venn-II is also equivalent in expressive power to MFOL. As a consequence, if we extend Venn-II to include the negation and conjunction operators then we do not increase expressiveness.

Whilst Euler diagrams and Venn-II are equivalent in expressive power, there are advantages of developing a sound and complete logic for Euler diagrams over Venn-II extended to include  $\neg$  and  $\wedge$ : many diagrammatic logics extend Euler diagrams but not Venn-II. For example, constraint diagrams [6] extend Euler diagrams and include all the connectives considered here. Furthermore, unlike Venn-II,  $\neg$  and  $\wedge$  are not both syntactic sugar in the constraint diagram case (see [3] for a formalization).

## 4 Reasoning Rules

In [12], it was found that rules which make ‘medium level’ changes to diagram syntax are generally good choices if we want to automatically find proofs efficiently. Moreover, it is sensible, from an automated theorem proving perspective, to define rules that change few different types of pieces of syntax because this helps us to define good search strategies; for example, a rule that applies to a compound diagram, changing its unitary parts and its tree structure changes many pieces of syntax. The rules presented here have been defined following both of these philosophies.

### 4.1 Information Preserving Rules

Our first rule allows us to change the contours in a diagram. For example, in figure 5, we can add a contour to  $d_1$  giving  $d_2$ . The two diagrams are semantically equivalent, with neither providing any information about the set

$D$ . We can also remove  $D$  from  $d_2$  to yield  $d_1$ . We will write  $d_1 \rightarrow d_2$  to mean that  $d_1$  can be replaced by  $d_2$  by the application of a reasoning rule.

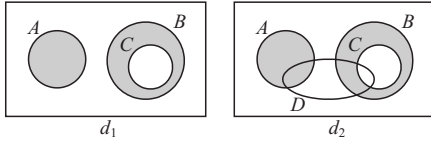


Figure 5. Changing contours.

**Rule 1.** Let  $d_1$  and  $d_2$  be unitary diagrams such that all of the contours in  $d_1$  are in  $d_2$  and  $d_2$  has exactly one more contour,  $l$ . If we can add  $l$  to  $d_1$  in such a way that  $l$  splits all of the zones in  $d_1$  into two zones and the shading does not change to give  $d_2$  then we say  $d_1 \rightarrow d_2$  by applying the **add a contour** rule.

**Rule 2.** Let  $d_1$  and  $d_2$  be unitary diagrams such that  $d_1 \rightarrow d_2$  by applying the add a contour rule. Then  $d_2 \rightarrow d_1$  by applying the **remove a contour** rule.

We know that shaded regions represent the empty set, as do missing regions. The diagram  $d_1$  in figure 5 asserts that  $A \cap B = \emptyset$  because the two zones inside both  $A$  and  $B$  are missing. We can replace  $d_1$  by  $d_3$  in figure 6, using the *add a shaded zonal region* rule. We can also replace  $d_1$  by  $d_4$  using a *remove a shaded zonal region* rule.

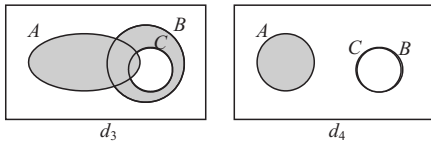


Figure 6. Changing zonal regions.

**Rule 3.** Let  $d_1$  and  $d_2$  be unitary diagrams such that the only difference between them is that  $d_2$  contains an entirely shaded zonal region that is completely missing from  $d_1$ . Then  $d_1 \rightarrow d_2$  by applying the **add a shaded zonal region** rule.

**Rule 4.** Let  $d_1$  and  $d_2$  be unitary diagrams such that  $d_1 \rightarrow d_2$  by applying the add a shaded zonal region rule. Then  $d_2 \rightarrow d_1$  by applying the **remove a shaded zonal region** rule.

In addition to the unitary rules introduced above, there are rules that operate on compound diagrams. In figure 7,  $d_1$  asserts that  $C - A = \emptyset$ . Therefore, we can add shading to  $d_2$  in the region corresponding to  $C - A$  to give  $d_3$ ;  $d_1 \wedge d_2$  is semantically equivalent to  $d_1 \wedge d_3$ .

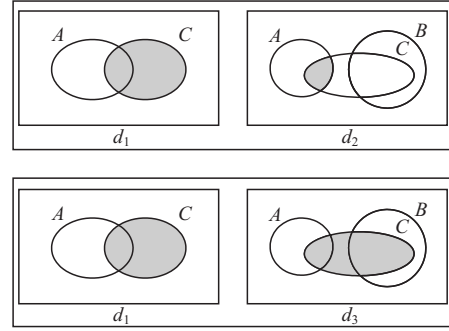


Figure 7. Changing shading.

**Rule 5.** Let  $d_1$  and  $d_2$  be unitary diagrams such that there exists a zonal region,  $zr$ , with the properties that

1. none of the zones in  $zr$  are missing from either  $d_1$  or  $d_2$  and
2.  $zr$  is completely shaded in  $d_1$  and
3.  $zr$  is completely non-shaded in  $d_2$

Let  $d_3$  be identical to  $d_2$  except that all zones in  $zr$  are shaded in  $d_3$ . Then  $d_1 \wedge d_2 \rightarrow d_1 \wedge d_3$  by applying the **add shading to a zonal region** rule.

**Rule 6.** Let  $d_3$  and  $d_4$  be diagrams such that  $d_3 \rightarrow d_4$  by applying the add shading to a zonal region rule. Then  $d_3 \rightarrow d_4$  by applying the **remove shading from a zonal region** rule.

The remaining rules in this section operate on the structure of compound diagrams and they have analogies in propositional logic. Firstly, we observe that our explicit representation of true is a unitary diagram,  $d$ , that contains no contours and no shading; in other words  $d = \square$ . Therefore, the diagram  $\neg\square$  is a contradiction. Throughout the remainder of this section, let  $d_1$ ,  $d_2$  and  $d_3$  be diagrams. We write  $d_1 \leftrightarrow d_2$  to denote  $d_1 \rightarrow d_2$  and  $d_2 \rightarrow d_1$ .

**Rule 7. Identity Law**  $d_1 \wedge \square \leftrightarrow d_1$ .

**Rule 8. Complement Laws**  $d_1 \vee \neg d_1 \leftrightarrow \square$ , and  $d_1 \wedge \neg d_1 \leftrightarrow \neg\square$

**Rule 9. De Morgan's Laws**  $\neg(d_1 \vee d_2) \leftrightarrow \neg d_1 \wedge \neg d_2$ , and  $\neg(d_1 \wedge d_2) \leftrightarrow \neg d_1 \vee \neg d_2$ .

**Rule 10. Involution**  $\neg\neg d_1 \leftrightarrow d_1$ .

**Rule 11. Distributivity**  $d_1 \wedge (d_2 \vee d_3) \leftrightarrow (d_1 \wedge d_2) \vee (d_1 \wedge d_3)$  and  $d_1 \vee (d_2 \wedge d_3) \leftrightarrow (d_1 \vee d_2) \wedge (d_1 \vee d_3)$ .

**Rule 12. Idempotency**  $d_1 \leftrightarrow d_1 \wedge d_1$  and  $d_1 \leftrightarrow d_1 \vee d_1$ .

For simplicity, we assume commutativity and associativity. Furthermore, we assume that we can apply the above rules to any part of a compound diagram; for example, if  $d_1$  can be replaced by  $d_2$  by applying some rule then  $(d_1 \wedge d_3) \vee \neg d_1 \vee d_4$  can be replaced by  $(d_2 \wedge d_3) \vee \neg d_2 \vee d_4$ .

## 4.2 Information weakening rules

Next, we introduce some rules that weaken the informational content of a diagram. These information weakening rules can only be applied inside an even number of negation signs. For example, if  $d_1$  can be replaced by  $d_2$  by applying one of the weakening rules then only the second occurrence of  $d_1$  can be replaced by  $d_2$  in the compound diagram  $\neg d_1 \vee \neg(d_3 \wedge \neg d_1) \vee \neg(d_1 \vee d_4)$ .

**Rule 13. Inconsistency**  $\neg \square \rightarrow d_1$ .

**Rule 14. Connecting a diagram**  $d_1 \rightarrow d_1 \vee d_2$ .

**Rule 15. Removing a diagram**  $d_1 \wedge d_2 \rightarrow d_1$ .

Information weakening rules cannot be applied inside an odd number of negation signs because such inferences are not necessarily sound. As an illustration, in figure 8, the diagram  $d_1 \vee d_2 \vee d_3$  can be obtained from  $d_1 \vee d_2$  by applying the connecting a diagram rule. An interpretation in which  $A = D = \emptyset$ ,  $B = \{1\}$  and  $C = \{2\}$  is a model for  $\neg(d_1 \vee d_2)$  but not a model for  $\neg(d_1 \vee d_2 \vee d_3)$ .

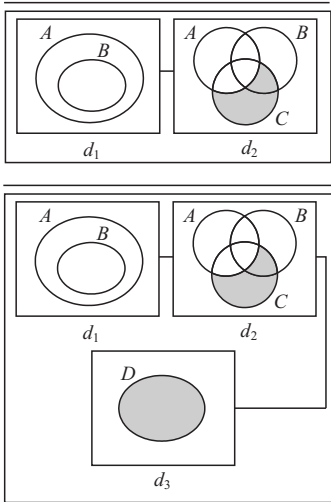


Figure 8. Incorrectly applying rules.

To conclude this section, we define obtainability.

**Definition 4.1.** We say  $d_2$  is *obtainable* from  $d_1$ , denoted  $d_1 \vdash d_2$ , if there exists a finite set of diagrams,  $\{d_3, \dots, d_n\}$ , such that  $d_1 \rightarrow d_3 \rightarrow \dots \rightarrow d_n \rightarrow d_2$ . If  $d_1 \vdash d_2$  and  $d_2 \vdash d_1$  then  $d_1$  and  $d_2$  are *syntactically equivalent*, denoted  $d_1 \equiv d_2$ .

## 5 Soundness and Completeness

To prove that the system is sound, an induction argument is used. The base case is trivial. For the inductive step it is sufficient to prove each rule is sound (that is, applying the rule maintains or enlarges the model set).

**Theorem 5.1. Soundness** If  $d_1 \vdash d_2$  then  $d_1 \models d_2$ .

The system is also complete. We give a reasonably thorough account of the completeness proof, omitting the details of the stages which have analogies in the *spider diagram* logic [5]. For spider diagrams, given  $d_1 \models d_2$ , the high level strategy is to convert both  $d_1$  and  $d_2$  into a disjunction of unitary diagrams each of which has some given set of labels,  $L$ , and set of zones,  $Z$ , without changing their semantics, and then reason about these disjunctions. For this Euler diagram system it is not possible to transform some diagrams into such disjunctions because of the presence of negation. Thus we need to modify the completeness proof strategy and it is likely that our new strategy will extend to other systems that include negation.

Our approach is also to begin by introducing contours to each unitary part of  $d_1$  and  $d_2$  until all unitary parts have the same label set,  $L$  say, giving  $d_1^L$  and  $d_2^L$  respectively. Next, we convert each unitary part of both  $d_1^L$  and  $d_2^L$  into Venn diagrams (so there are no missing zones) using the add a shaded zonal region rule, giving  $d_1^Z$  and  $d_2^Z$  respectively. So, each unitary part of both  $d_1^Z$  and  $d_2^Z$  has zone set  $Z = \{a, L - a\}$ . We now deviate from the spider diagram strategy.

Our aim is to convert  $d_1^Z$  into disjunctive normal form and  $d_2^Z$  into conjunctive normal form where, in both cases, each unitary part contains at most one shaded zone. From now on, unless stated otherwise we will assume, without loss of generality, that all unitary diagrams have zone set  $Z$  and, therefore, have no missing zones.

**Definition 5.1.** Let  $d$  be a unitary diagram with no missing zones and at most one shaded zone. Then  $d$  and  $\neg d$  are called *literals*.

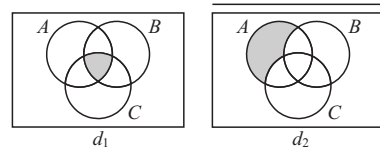


Figure 9. Two literals.

A literal,  $d$ , with a shaded zone  $z$ , gives us a single piece of information: in the *positive* case  $d$  tells us that  $z$  represents the empty set and in the *negative* case  $d$  tells us that  $z$  does not represent the empty set. For example, the positive

literal  $d_1$  in figure 9 asserts that  $A \cap B \cap C = \emptyset$  whereas the negative literal  $d_2$  expresses that  $A - (B \cup C) \neq \emptyset$ . A literal with no shaded zones is either universally valid (the positive case) or a contradiction (the negative case).

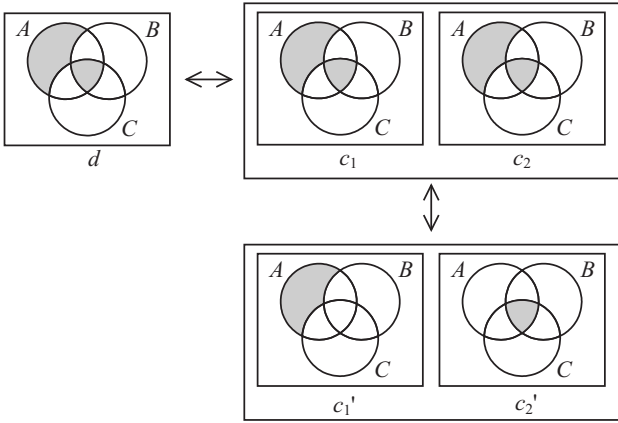
**Definition 5.2.** A diagram of the form

$$\bigwedge_{1 \leq x \leq p} \bigvee_{1 \leq y \leq q} d_{x,y},$$

where each  $d_{x,y}$  is a literal, is in **literal conjunctive normal form (LCNF)**. A diagram of the form

$$\bigvee_{1 \leq i \leq m} \bigwedge_{1 \leq j \leq n} c_{i,j},$$

where each  $c_{i,j}$  is a literal, is in **literal disjunctive normal form (LDNF)**.



**Figure 10. Obtaining LCNF.**

**Lemma 5.1.** Every unitary diagram,  $d$ , is syntactically equivalent to a diagram in LCNF.

**Proof** If  $d$  has no shaded zones or exactly one shaded zone then  $d$  is in LCNF. Otherwise, enumerate the shaded zones in  $d$ , giving  $\{z_1, \dots, z_n\}$ , and apply the idempotency rule to  $d$  in order to obtain  $\bigwedge_{1 \leq i \leq n} c_i$  where each  $c_i$  equals  $d$  (see figure 10). For each  $c_i$  apply the remove shading from a zonal region rule, removing shading from all of the shaded zones except  $z_i$ , giving  $c'_i$ . Clearly  $d \equiv \bigwedge_{1 \leq i \leq n} c'_i$ .  $\square$

**Theorem 5.2.** Every diagram,  $d$ , is syntactically equivalent to a diagram in LCNF.

**Proof (Sketch)** Replace each unitary part of  $d$  by a conjunction of literals (this can be done by lemma 5.1). Then use the information preserving rules that have analogies in propositional logic to convert to LCNF.  $\square$

**Theorem 5.3.** Every diagram,  $d$ , is syntactically equivalent to a diagram in LDNF.

The proof is similar to the LCNF case.

For the next step in our completeness argument, we replace  $d_1^Z$  by a syntactically equivalent diagram in LDNF, say

$$d_1^{LD} = \bigvee_{1 \leq i \leq m} \bigwedge_{1 \leq j \leq n} c_{i,j}.$$

The diagram  $d_2^Z$  is replaced by a syntactically equivalent diagram in LCNF, say

$$d_2^{LC} = \bigwedge_{1 \leq x \leq p} \bigvee_{1 \leq y \leq q} d_{x,y}.$$

We observe that  $d_1 \equiv d_1^{LD}$  and  $d_2 \equiv d_2^{LC}$ . Therefore, if  $d_1^{LD} \vdash d_2^{LC}$  then  $d_1 \vdash d_2$ . Hence, if we can show  $d_1^{LD} \vdash d_2^{LC}$  then we have proved that the system is complete. In order to show  $d_1^{LD} \vdash d_2^{LC}$ , we further observe that it is sufficient to prove, for each  $i$  ( $1 \leq i \leq m$ )

$$\bigwedge_{1 \leq j \leq n} c_{i,j} \vdash d_2^{LC}.$$

Since, when showing completeness, we start by assuming  $d_1 \models d_2$ , we know

$$\bigwedge_{1 \leq j \leq n} c_{i,j} \models d_2^{LC},$$

so we can deduce that, for each  $x$  ( $1 \leq x \leq p$ ),

$$\bigwedge_{1 \leq j \leq n} c_{i,j} \models \bigvee_{1 \leq y \leq q} d_{x,y}.$$

We prove

$$\bigwedge_{1 \leq j \leq n} c_{i,j} \vdash \bigvee_{1 \leq y \leq q} d_{x,y}$$

from which it follows that the system is complete.

We will first establish some syntactic relationships between  $\bigwedge_{1 \leq j \leq n} c_{i,j}$  and  $\bigvee_{1 \leq y \leq q} d_{x,y}$  when  $\bigwedge_{1 \leq j \leq n} c_{i,j}$  is satisfiable. Recall that  $\square$  is a unitary diagram containing no contours or shading. The notation  $\square^L$  will be used to denote a positive literal with label set  $L$ , zone set  $Z$  and no shading. We extend this notation, and denote a positive literal with label set  $L$ , zone set  $Z$  and shaded zone  $z$  by  $\square_z^L$ .

**Theorem 5.4.** Let  $C = \bigwedge_{1 \leq j \leq n} c_j$  be a satisfiable conjunction of literals and let  $D = \bigvee_{1 \leq y \leq q} d_y$  be a disjunction of literals. If  $C \models D$  then either

- (1) there exists  $j$  and  $y$  such that  $c_j = d_y$  or
- (2) there exists  $x$  and  $y$  such that  $d_x = \neg d_y$  or

(3) the diagram  $\square^L$  occurs in  $D$ .

**Proof** We proceed by contradiction and assume that none of the above three conditions hold. Reduce  $D$  to a disjunction of literals in which  $\neg\square^L$  never occurs, using remove a contour (removing all of the contours in  $L$  from  $\neg\square^L$  to give  $\neg\square$ ) and inconsistency (replacing  $\neg\square$  by any literal in  $D$  that contains a shaded zone), to give  $D'$ ; this can be done because (3) is false and, because  $C$  is satisfiable,  $D$  is satisfiable. For clarity, the diagram  $D'$  is a disjunction of literals, each with a shaded zone. Moreover,  $D \models D'$  so  $C \models D'$ . Thus, showing that  $C \not\models D'$  will give us a contradiction.

We will construct a model,  $m$ , for  $C$  that is not a model for  $D'$ . We take the universal set to be

$$U = \{z \in Z : \square_z^L \text{ is in } D' \text{ or } \neg\square_z^L \text{ is in } C\}$$

and each zone represents the empty set if  $z$  is not in  $U$ , otherwise  $z$  represents itself.

First, we show that  $m$  satisfies  $C$ . Let  $c$  be a literal in  $C$  and suppose  $c$  is positive. For  $m$  to satisfy  $c$  it must be that any zone which is shaded in  $c$  (of which there is at most one) represents the empty set. If  $c$  has no shaded zones then  $m$  trivially satisfies  $c$ . Otherwise,  $c = \square_z^L$ , for some  $z \in Z$ , and  $z$  does not represent the empty set if and only if  $\square_z^L$  is in  $D'$  or  $\neg\square_z^L$  is in  $C$  (by the definition of  $m$ ). In the former case  $c$  occurs in both  $C$  and  $D'$  which cannot happen (condition (1) above is false). The latter case also cannot happen since  $C$  is satisfiable. Hence  $m$  satisfies  $c$ .

Alternatively,  $c$  is negative. If  $c$  has no shaded zones then  $c = \neg\square^L$ , which is unsatisfiable, contradicting the assumption that  $C$  is satisfiable. Therefore  $c$  has exactly one shaded zone,  $z$  say, and  $c = \neg\square_z^L$ . Then the only way  $m$  can fail to be a model for  $c$  is if  $z$  represents the empty set. But, by the definition of  $m$ ,  $z \in U$  so  $z$  represents itself in  $m$  and not the empty set. Hence  $m$  satisfies  $c$ . In any case,  $m$  is a model for each  $c$  in  $C$ , so  $m$  is a model for  $C$ .

Now, we show that  $m$  is not a model for  $D'$ . To do so, we show that  $m$  is not a model for any  $d'$  in  $D'$ . Let  $d'$  be a literal in  $D'$  and suppose that  $d'$  is positive. Then  $d' = \square_z^L$  for some  $z$  and we want to show that  $z \in U$  (i.e.  $z$  does not represent the empty set); this follows trivially from the definition of  $U$ , so  $m$  is not a model for  $d'$ . Alternatively,  $d'$  is negative and of the form  $\neg\square_z^L$  for some  $z$ ;  $\neg\square_z^L$  asserts the non-emptiness of  $z$  so, for  $m$  not to satisfy  $\neg\square_z^L$  we must show that  $z \notin U$ . Since (1) is false,  $\neg\square_z^L$  is not in  $C$ . Furthermore, we know that, for each unitary diagram,  $d$ , if  $d$  appears in  $D'$  then  $\neg d$  does not appear in  $D'$  because (2) is false. Therefore,  $z \notin U$ , so  $z$  represents the empty set and  $m$  does not satisfy  $d'$ . Hence  $m$  is not a model for any literal in  $D'$ , so  $m$  is not a model for  $D'$ .

Thus, we have reached a contradiction, since  $C \models D \models D'$  implies  $C \models D'$  but we have shown that  $C \not\models D'$ . Therefore at least one of the three conditions stated in the theorem holds.  $\square$

Using theorem 5.4, we are able to show that  $C \vdash D$ .

**Theorem 5.5.** Let  $C = \bigwedge_{1 \leq j \leq n} c_j$  be a satisfiable conjunction of literals and let  $D = \bigvee_{1 \leq y \leq q} d_y$  be a disjunction of literals. If  $C \models D$  then  $C \vdash D$ .

**Proof** We break the proof down into three cases, corresponding to those in the statement of theorem 5.4.

- (1) *There exists  $j$  and  $y$  such that  $c_j = d_y$ .* Choose such a  $c_j$  and  $d_y$ . Remove diagrams from  $C$  until only  $c_j = d_y$  remains, using the removing a diagram rule. Next, connect a diagram to  $c_j = d_y$  yielding  $D$ , using the connecting a diagram rule. Hence  $C \vdash D$ .
- (2) *There exists  $x$  and  $y$  such that  $d_x = \neg d_y$ .* In this case, take  $C$  and use the identity law to obtain  $C \wedge \square$ . Using the removing a diagram rule, obtain  $\square$ , to which we apply the complement law to give  $d_x \vee d_y$ . From this, use the connecting a diagram rule (if necessary) to obtain  $D$ .
- (3) *The diagram  $\square^L$  occurs in  $D$ .* Apply the identity law and removing a diagram rule to obtain  $\square$  from  $C$ . Next, add contours to  $\square$  to give  $\square^L$  (which is in  $D$ ). Finally, use the connecting a diagram rule (if necessary) to obtain  $D$ .

In each case, we have shown that  $C \vdash D$ .  $\square$

To conclude our completeness proof, we now show that, if  $C$  is unsatisfiable, then  $C \vdash D$ . To do so, we prove that  $C \vdash \neg\square$  after identifying some syntactic conditions on  $C$ .

**Lemma 5.2.** Let  $C = \bigwedge_{1 \leq j \leq n} c_j$  be an unsatisfiable conjunction of literals. Then either  $C$  contains  $\neg\square^L$  or  $C$  contains both  $\neg\square_z^L$ , and  $\square_z^L$  for some zone  $z$ .

**Proof** Suppose that  $C$  does not contain  $\neg\square^L$ . Define an interpretation,  $I$ , where

$$U = Z - \{z' : \square_{z'}^L \text{ is in } C\}$$

and every zone in  $U$  maps to itself in  $I$  and all of the other zones map to the empty set. Obviously,  $I$  satisfies all of the positive literals in  $C$  so, since  $C$  is unsatisfiable, there exists a negative literal in  $C$ , say  $c$ , with shaded zone  $z$ , which is not satisfied by  $I$ ;  $c = \neg\square_z^L$ . Since  $I$  does not satisfy  $\neg\square_z^L$ ,  $z$  represents the empty set and, therefore, must be shaded in some positive literal, namely  $\square_z^L$ , in  $C$  as required.  $\square$

**Lemma 5.3.** Let  $C = \bigwedge_{1 \leq j \leq n} c_j$  be an unsatisfiable conjunction of literals. Then  $C \vdash \neg\square$ .

**Proof** We consider two cases, corresponding to those in the statement of lemma 5.2.

1.  $C$  contains  $\neg\Box^L$ . Remove all of the contours from  $\neg\Box^L$ , to give  $\neg\Box$  using the remove contour rule. Replace  $\neg\Box$  by a conjunction of literals so that, in the result, each unitary part appears both positively and negatively using the inconsistency rule. Replace each pair of literals,  $d \wedge \neg d$ , by  $\neg\Box$  using the complement laws (this may also require idempotency). Finally, use idempotency to obtain  $\neg\Box$ .
2.  $C$  contains  $\neg\Box_z^L$  and  $\Box_z^L$  for some zone  $z$ . Replace  $\Box_z^L \wedge \neg\Box_z^L$  by  $\neg\Box$ , using the complement law and then proceed as the first case.

Therefore  $C \vdash \neg\Box$ . □

**Theorem 5.6.** Let  $C = \bigwedge_{1 \leq j \leq n} c_j$  be an unsatisfiable conjunction of literals and let  $D = \bigvee_{1 \leq y \leq q} d_y$  be a disjunction of literals. Then  $C \vdash D$ .

**Proof** By lemma 5.3, replace  $C$  with  $\neg\Box$  then use inconsistency to obtain  $D$ . □  
Hence the system is complete.

**Theorem 5.7. Completeness** If  $d_1 \models d_2$  then  $d_1 \vdash d_2$ .

## 6 Conclusion

In this paper, we have presented a sound and complete Euler diagram logic that includes operators  $\neg$ ,  $\wedge$  and  $\vee$ . Typically, logics that extend Euler diagrams include  $\vee$  and, sometimes,  $\wedge$  but not  $\neg$ . Including  $\neg$  impacts the strategy used to prove completeness. Typically, the Euler diagram based logics that do not include  $\neg$  rely on being able to reduce  $d_1$  and  $d_2$ , where  $d_1 \models d_2$ , to a disjunction of unitary diagrams as part of the completeness proof. Reducing to disjunctions of unitary diagrams, say  $\bigvee c_i$  and  $\bigvee d_j$  allows one then to prove completeness by simply showing that each  $c_i$  implies one of the  $d_j$ 's, for example. See [5, 9, 10, 11] for logics whose completeness proofs use this type of strategy.

When negation is included, obtaining such disjunctions is not necessarily possible, as in this Euler diagram case. It is likely that our strategy, to reduce  $d_1$  to literal disjunctive normal form and  $d_2$  to literal conjunctive normal form, will extend to other Euler diagram based logics where constructive completeness proofs exist (i.e. in decidable languages). At the very least, we anticipate that the strategy will extend to the systems presented in [1, 4, 5, 9, 10, 11, 15].

**Acknowledgement** Gem Stapleton is supported by a Leverhulme Trust Early Career Fellowship and by UK EPSRC grant EP/E011160/1 for the Visualization with Euler Diagrams project.

## References

- [1] L. Choudhury and M. K. Chakraborty. On extending Venn diagrams by augmenting names of individuals. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 142–146. Springer-Verlag, March 2004.
- [2] A. Fish and J. Flower. Investigating reasoning with constraint diagrams. In *Visual Language and Formal Methods 2004*, volume 127 of *ENTCS*, pages 53–69, Rome, Italy, 2005. Elsevier.
- [3] A. Fish, J. Flower, and J. Howse. The semantics of augmented constraint diagrams. *Journal of Visual Languages and Computing*, 16:541–573, 2005.
- [4] E. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.
- [5] J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS Journal of Computation and Mathematics*, 8:145–194, 2005.
- [6] S. Kent. Constraint diagrams: Visualizing invariants in object oriented modelling. In *Proceedings of OOPSLA97*, pages 327–341. ACM Press, October 1997.
- [7] O. Patrascoiu, S. Thompson, and P. Rodgers. Tableaux for diagrammatic reasoning. In *Proceedings of the Eleventh International Conference on Distributed Multimedia Systems, International Workshop on Visual Languages and Computing*, pages 279–286. Knowledge Systems Institute, September 2005.
- [8] C. Peirce. *Collected Papers*, volume 4. Harvard University Press, 1933.
- [9] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.
- [10] G. Stapleton, J. Howse, and J. Taylor. A constraint diagram reasoning system. In *Proceedings of International Conference on Visual Languages and Computing*, pages 263–270. Knowledge Systems Institute, 2003.
- [11] G. Stapleton, J. Howse, and J. Taylor. A decidable constraint diagram reasoning system. *Journal of Logic and Computation*, 15(6):975–1008, December 2005.
- [12] G. Stapleton, J. Masthoff, J. Flower, A. Fish, and J. Southern. Automated theorem proving in Euler diagrams systems. *Accepted for Journal of Automated Reasoning*, to appear 2007.
- [13] G. Stapleton, S. Thompson, J. Howse, and J. Taylor. The expressiveness of spider diagrams. *Journal of Logic and Computation*, 14(6):857–880, December 2004.
- [14] N. Swoboda and 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):136–149, 2004.
- [15] N. Swoboda and G. Allwein. Heterogeneous reasoning with Euler/Venn diagrams containing named constants and FOL. In *Proceedings of Euler Diagrams 2004*, volume 134 of *ENTCS*. Elsevier Science, 2005.