Equivalences in Euler-based diagram systems through normal forms

Andrew Fish and John Taylor

ABSTRACT

The form of information presented can influence its utility for the conveying of knowledge by affecting an interpreter's ability to reason with the information. There are distinct types of representational systems (for example, symbolic versus diagrammatic logics), various sub-systems (for example, propositional versus predicate logics), and even within a single representational system there may be different means of expressing the same piece of information content. Thus, to display information, choices must be made between its different representations, depending upon many factors such as: the context, the reasoning tasks to be considered, user preferences or desires (for example, for short symbolic sentences or minimal clutter within diagrammatic systems). The identification of all equivalent representations with the same information content is a sensible precursor to attempts to minimise a metric over this class. We posit that defining notions of semantic redundancy and identifying the syntactic properties that encapsulate redundancy can help in achieving the goal of completely identifying equivalences within a single notational system or across multiple systems, but that care must be taken when extending systems, since refinements of redundancy conditions may be necessary even for conservative system extensions. We demonstrate this theory within two diagrammatic systems, which are Euler-diagram-based notations. Such notations can be used to represent logical information and have applications including visualisation of database queries, social network visualisation, statistical data visualisation, and as the basis of more expressive diagrammatic logics such as constraint languages used in software specification and reasoning. The development of the new associated machinery and concepts required is important in its own right since it increases the growing body of knowledge on diagrammatic logics. In particular, we consider Euler diagrams with shading, and then we conservatively extend the system to include projections, which allow for a much greater degree of flexibility of representation. We give syntactic properties that encapsulate semantic equivalence in both systems, whilst observing that the same semantic concept of redundancy is significantly more difficult to realise as syntactic properties in the extended system with projections.

1. Introduction

We focus on particular diagrammatic systems, providing normal forms for two Euler diagram (ED) systems, Euler diagrams with shading (EDS) and Euler diagrams with projections (EDP). This provides a unique representative amongst the classes of semantically equivalent diagrams, and for each system we provide a procedure for transforming any given diagram into its normal form, thereby permitting an easy equivalence check. Along the way we develop general machinery which will be useful to the field, and observe that even conservative system extensions require a revisiting of all notions of syntactic and semantic redundancy due to potential interactions.

First, we describe works that impinge on the topic area, relating to cognition, diagrammatic theory and normal forms in symbolic logic. Then, we provide a summary of contributions,

definitions and results, for reference purposes whilst reading the paper, as well as a description of the paper structure.

Larkin and Simon [61] compare the ease of computation with sentential and diagrammatic representations with the same informational content, making use of terms *informational equivalence* (the information in either representation is inferable from the other) and *computational equivalence* (informational equivalence holds and the ease of inferences in either representation is comparable). Gurr [33] sketches a theory of diagrammatic communication, based upon studies of relevant syntactic, semantic and pragmatic components, exploring the question of 'what makes for an effective diagrammatic representation?'. In [67], Shijomina brings together important ideas, from many sources, relating to the functional traits of graphical representations: *free ride properties*, expressing a certain set of information in the system always results in the expression of another, consequential piece of information; *auto-consistency*, incapacity of the system to express a certain range of inconsistent sets of information; *specificity*, incapacity of the system to express certain sets of information without choosing to express another, non-consequential piece of information; *meaning derivation properties*, capacity to express semantic contents not defined in the basic semantic conventions, but only derivable from them. Gurr and Tourlas [34] claim to permit 'a principled approach to the identification of intuitive (and counter-intuitive) features of diagrammatic languages'. Hegarty [41] argues that 'the design of external visualizations should be informed by research on internal visualization skills, and that the development of technologies for external visualizations calls for more research on the nature of internal visualization abilities'.

Ruskey and Weston provided a comprehensive survey of Venn diagrams [66], describing historic and modern mathematics involved. EDs permit the relaxation of the Venn diagram [78] requirement that every set intersection of the set system is represented, enabling the presentation of the transitive subset relation in a manner conductive for free rides, for example. Whilst EDs are an important notation in their own right, their major significance is due to the fact that they form the basis of a number of other different notations. Harel [39, 40] developed higraphs, a diagrammatic system that extends EDs making use of additional graph-based features, suitable for applications to databases, knowledge representation, and the behavioral specification of complex concurrent systems using the higraph-based language of statecharts. EDs have been applied in a variety of areas including personal resource management [13, 17], library database query language [77], genetic set relation visualisation in bio-informatics [59, 60] and statistical data visualisation [79]. Visualisation techniques for EDs with additional graph based features have also been developed for highlighting clusters of nodes or for network visualisation [12, 65], for example.

Another key application area of EDs is that of formal diagrammatic logics, which is a relatively new avenue of research, offering a potential alternative to the standard symbolic logics, attempting to make use of spatial relationships as an advantage within a logical setting. EDs are the modern version of Euler circles [23] which were introduced for the purpose of syllogistic reasoning. Shin, Hammer and Barwise [5, 36, 38, 68] provided seminal ideas about the production of such formal diagrammatic logics. Shin's Venn-II system [68] is expressively equivalent to monadic first-order logic (MFOL). Stapleton and Masthoff have shown [73] that EDs together with a negation operator (as well as conjunction or disjunction) is also expressively equivalent to MFOL. There are other related systems that extend EDs or provide variations in syntax and semantics, such as spider diagrams [47] and Euler/Venn diagrams [76]. Spider diagrams [47] extend EDs by permitting the expression of existential quantification and cardinality restrictions on sets. This Spider diagram system [47, 75] is expressively equivalent to monadic first-order logic with equality (MFOL[=]). The expressiveness of the Euler/Venn diagram system has not been formally established but it lies between MFOL and MFOL[=]. Delaney *et al.* defined spider diagrams of order [18], extending spider diagrams with an ordering operator, and subsequently showed the logic is expressively equivalent to monadic

first-order logic of order, (MFOL[<]), and examined the expressiveness of various fragments in [**19**, **20**].

In terms of applications of such logics, constraint diagrams [**58**] were developed to specify and reason with constraints in an object-oriented setting. They extend spider diagrams via the additional explicit representation of universal quantification and binary relations. Since there can be flexibility in the reading order of diagrammatic notations (versus a linear order in symbolic logics), one can augment a diagram with extra information to provide the necessary uniqueness of semantics assigned to any given diagram [**25**]. In [**70**], constraint diagrams were shown to be no more expressive than dyadic first-order logic and a system of generalised constraint diagrams which is at least as expressive as dyadic first-order logic was proposed. An example of the use of constraint diagrams for modelling a software system was presented in [**45**], together with a parallel construction in the symbolic language Z. The spider diagram notation has also been extended for practical usage by explicitly linking to types and instances in an object oriented setting as well as extending to incorporate a temporal component, along the lines of Allen's interval calculus [**2**], in the policy specification domain [**8**]. A variation of constraint diagrams, called concept diagrams, was proposed for ontology modelling purposes [**48**]; these diagrams permit quantification over sets and are expressively equivalent to dyadic second-order logic [**71**]. Chapman *et al.* also defined second-order spider diagrams, generalising spider diagrams of order, including existential quantification over subsets, indicating that it is at least as expressive as monadic second-order logic [**9**]. Furthermore, Conceptual Spider Diagrams were introduced in [**16**] as a hybrid notion combining the advantages of spider diagrams, regarding the representation of set-based hierarchies, with the power of Sowa's conceptual graphs [**51**] regarding the expression of relational predicates between objects; the latter have history tracing to Charles Sanders Peirce's existential graphs.

When considering EDS (used to place an upper bounds on the cardinality of sets, thereby indicating sets are empty in this system), there are different ways of representing relationships between sets. For instance, disjoint sets may be represented either by interior-disjoint curves or by overlapping curves with shaded region of overlap; this is analogous between a choice of presentation of set relationships of $A \subseteq B$ versus $A \cap B \neq \varnothing$ in propositional logic. The decision between different representations within a diagrammatic notation (or between notations) may involve many factors such as user preferences, drawability criteria (called well-formedness conditions in [**29**]) and 'clutter' metrics [**55**].

In symbolic logic, there are various normal forms for formulae. For example, in propositional logic a formula is in *disjunctive normal form* if it is a disjunction of a conjunction of literals or *conjunctive normal form* if it is a conjunction of a disjunction of literals. In predicate logic, a formula is in *prenex normal form* if all of the quantifiers 'come first'. There are algorithms that take an 'input' formula and produce a logically equivalent formula in the required normal form. Normal forms have a wide variety of uses in logical systems. For example, simplification of propositional logic formulae, with applications in circuit design, using Karnaugh maps [**57**] or the Quine–McCluskey algorithm, assume an input formula in disjunctive normal form. In predicate logic, Gödel's proof of the completeness of first-order logic supposes that all formulae are expressed in prenex normal form. Automated theorem provers make extensive use of normal forms in the manipulation, or rewriting, of formulae.

Paper contributions and structure. In this paper, we provide normal forms for unitary EDS as well as for unitary EDP; one can extend *unitary* systems to *compound* systems which permit logical connectives between diagrams. Projections give a means to provide 'local' information within a unitary diagram, providing flexibility in the presentation of information, whilst overcoming certain specificity issues. These normal forms concern the 'internal' structure of unitary diagrams, rather than the 'global' structure of compound diagrams (how the compound
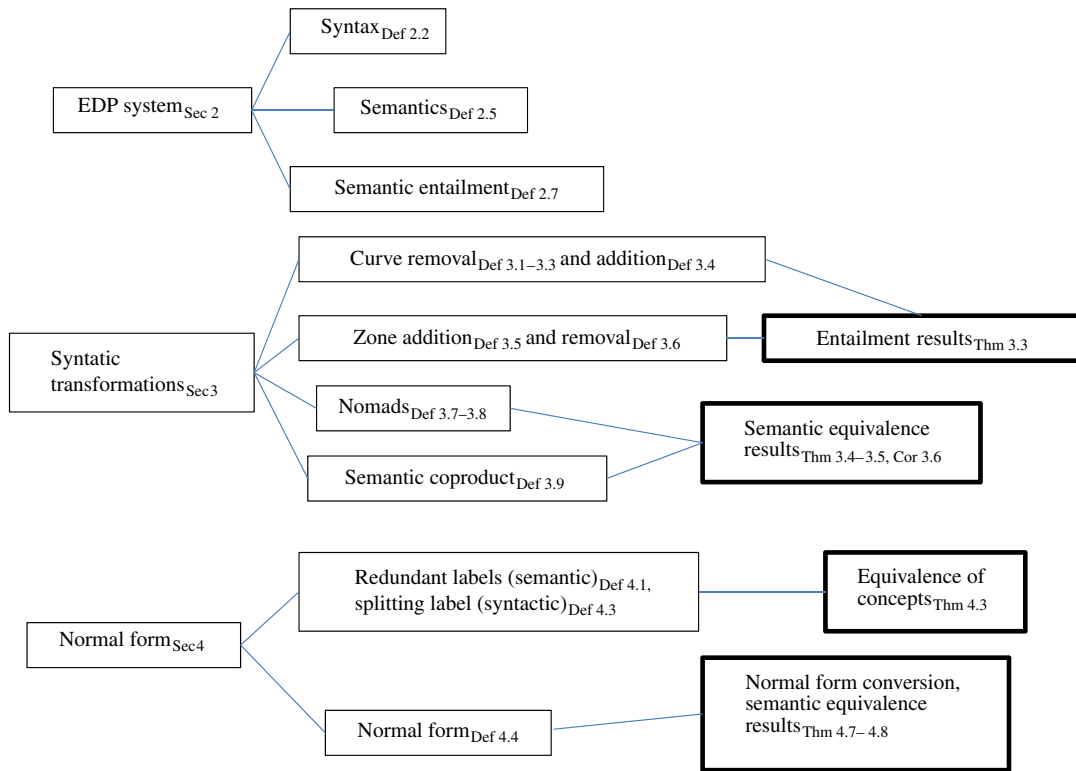
FIGURE 1. *The landscape of definitions and theorems from* §§ 2–4.

diagram is built from unitary diagram components). We provide algorithms that produce, for any diagram, a semantically equivalent diagram in its normal form. Thus, one can compare the semantic equivalence class of any two diagrams (within either system) via conversion to their normal forms. Along the way, the identification of purely syntactic conditions that completely encapsulate semantic equivalence are discovered, permitting the treatment of diagrams syntactically, without the need for recourse to model theory. The normal forms have the additional property that they are the minimally cluttered of all semantically equivalent diagrams (using any of the clutter metrics introduced in [**55**]).

We introduce the syntax and semantics of EDS in § 2. Section 3 introduces various concepts and syntactic manipulations needed to define the normal form, including *nomads* (capturing a particular syntactic form of the representation of emptiness) and the *semantic coproduct* (for capturing common information). In § 4 we give a syntactic characterisation (using *splitting labels*) of the semantic property of redundancy of contours and we define the normal form for the EDS system. Section 5 follows the same strategy employed for the EDS system for the more complicated conservative extension to the EDP system. Further related work and discussions on variations of choices made occur in § 6. Finally, conclusions and directions of future work are discussed in § 7. For reference throughout the paper, see Figures 1 and 2.

2. *Euler diagrams with shading*

2.1. *Syntax of Euler diagrams*

The simplest type of diagrams we consider are EDS and this section provides a standard description of EDS; similar descriptions of related systems can be found in [**28**, **42**, **47**, **55**]
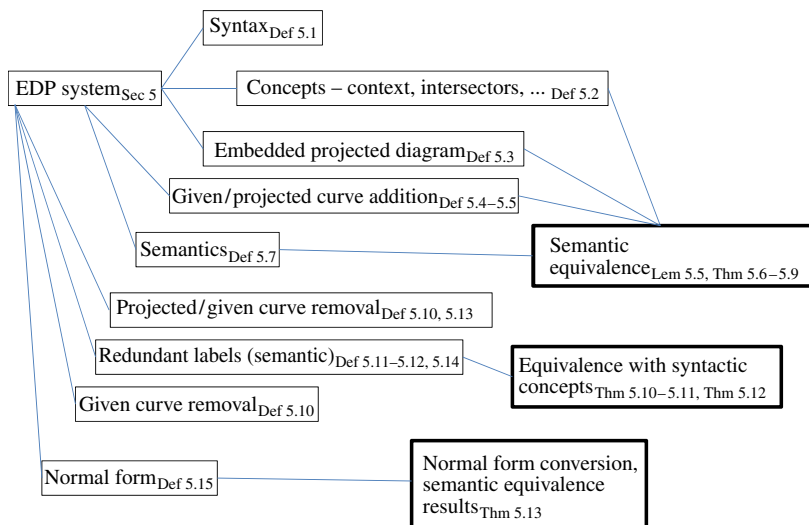
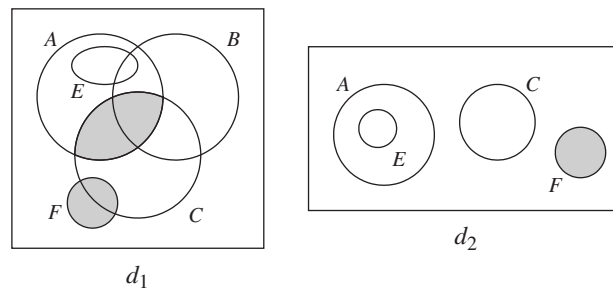FIGURE 2. *The landscape of definitions and theorems from* §5.

FIGURE 3. *Two Euler diagrams representing the same information.*

for example. EDs are built using contours (that is, simple closed curves) to represent sets. The topological relationships of 'containment' and 'separation' between contours represent the relations of subset and disjointness between sets respectively. We augment EDS. A shaded region of an ED represents the empty set.

EXAMPLE 2.1. Figure 3 shows two EDS. The diagram d_1 expresses the following properties of the sets represented in the diagram:
  (1) $A \cap C = \varnothing$ by the shading in the 'overlap' region between A and C;
  (2) $E \subseteq A \cap \overline{C} \cap \overline{F}$ by the positioning of the contour E inside contour A but outside contours C and F;
  (3) $F = \varnothing$ by shading the whole region inside F.
Although the contour B appears in the diagram, d_1 makes no assertion about the corresponding set B. In this sense the contour B is 'redundant'.

The diagram d_2 represents the same information as d_1 about these sets but it is visually simpler. The properties of the sets are more readily observed from d_2 than d_1.

Informally, an EDS in the plane $\mathbb{R}^2$ comprises a collection of simple closed curves called *contours*. We draw the contours of our EDs inside a bounding rectangle although formally the rectangle is not part of the diagram but serves to indicate 'where the diagram ends'. The contours divide the part of the plane inside the bounding box into connected *regions*.

A minimal region is called a *zone*. The diagram d_1 in Figure 3 has 12 zones. A zone can be described neatly as being inside some of the contours of the diagram and outside the remaining contours. For example, the four shaded zones in d_1 in Figure 3 can be described as follows:

inside A, C;	outside B, E, F;
inside A, B, C;	outside E, F;
inside C, F;	outside A, B, E;
inside F;	outside A, B, C, E.

To specify an EDS d, it is sufficient to specify the following.

(1) The set of contours in d.

(2) Which zones are present. A zone can be described as a pair (in, out) where in and out are sets of contours that partition† the set of contours of d.

(3) Which zones are shaded.

For example, the diagram d_2 in Figure 3 is specified as follows.

(1) The set of contours is $\{A, C, E, F\}$.

(2) The five zones are: $(\{A\}, \{C, E, F\}), (\{A, E\}, \{C, F\}), (\{C\}, \{A, E, F\}), (\{F\}, \{A, C, E\})$ and $(\varnothing, \{A, C, E, F\})$.

(3) There is one shaded zone: $(\{F\}, \{A, C, E\})$.

In fact, the set of contours may be deduced from the description of any one of the zones.

We call this specification of d_1 in Figure 3 an *abstract diagram*. The benefits of distinguishing clearly between '*concrete diagrams*': that is, diagrams that are drawn on paper or realised on some other medium such as a computer screen: and their abstract description is well-documented; see [**42**], for example.

It is convenient for the contour labels of the diagrams in the system to be drawn from a fixed, countably infinite set $\mathcal{L}$. Given $\mathcal{L}$, we can define an *abstract zone* to be an ordered pair (a, b) where a and b are finite subsets of $\mathcal{L}$ such that $a \cap b = \varnothing$. We may think of the zones of an abstract diagram d as being drawn from the set of abstract zones (a, b) where $a \cup b$ is the set of contour labels in the diagram. Thus, for example, the abstract zone $(\{A, B\}, \{C, D\})$ could appear in the description of an ED with contour label set $\{A, B, C, D\}$. We formalise these notions in the following definition.

DEFINITION 2.1. The set $\mathcal{L}$ is a countably infinite set of *contour labels* from which all contour labels of diagrams will be drawn.

An *abstract zone* on $\mathcal{L}$ is an ordered pair $z = (in(z), out(z))$ where $in(z)$ and $out(z)$ are disjoint, finite subsets of $\mathcal{L}$. The set $in(z)$ is called the set of contour labels that *contain z* and the set $out(z)$ is the called the set of contour labels that *exclude z*. The set of abstract zones is

$$\mathcal{Z} = \{(a, b) \in \mathbb{F}(\mathcal{L}) \times \mathbb{F}(\mathcal{L}) : a \cap b = \varnothing\}$$

where $\mathbb{F}(\mathcal{L})$ denotes the set of all finite subsets of $\mathcal{L}$.

An *abstract region* on $\mathcal{L}$ is a set of abstract zones. The set of regions on $\mathcal{L}$ is $\mathcal{R} = \mathbb{P}(\mathcal{Z})$, the power set of $\mathcal{Z}$.

We are now in a position to define an abstract unitary EDS.

DEFINITION 2.2. An *abstract EDS* d with labels in $\mathcal{L}$ is an ordered pair $\langle Z, Z^\bullet \rangle$ whose components are defined as follows.

(i) The set $Z = Z(d) \subseteq \mathcal{Z}$ is a finite set of zones such that, for some finite set of contour labels $L \in \mathbb{F}(\mathcal{L})$, the following conditions are satisfied:

(a) for all $z \in Z$, $in(z) \cup out(z) = L$; the set $L = L(d)$ is called the set of *contour labels* in d;

(b) for all $\ell \in L$, there is a zone $z \in Z$ such that $\ell \in in(z)$;

---

†In this paper we allow sets in a partition to be empty.

The set $Z = Z(d)$ is called the set of *zones* in d.

(ii) The set $Z^\bullet = Z^\bullet(d) \subseteq Z(d)$ is the set of *shaded zones*.

Henceforth, we will use 'ED' to mean 'abstract EDS'. In the following definition we introduce some terminology that will be used in the rest of the paper.

DEFINITION 2.3. Let $d = \langle Z, Z^\bullet \rangle$ be an ED.

(i) The set of *unshaded zones* of d is $Z^\circ(d) = Z(d) - Z^\bullet(d)$.

(ii) A *region* is a non-empty set of zones; the set of regions in d is $R = R(d) = \mathbb{P}(Z) - \{\varnothing\}$. We also define $R^\circ = R^\circ(d) = \mathbb{P}(Z^\circ) - \{\varnothing\}$ to be the set of *unshaded regions* in d and $R^\bullet = R^\bullet(d) = \mathbb{P}(Z^\bullet) - \{\varnothing\}$ to be the set of *shaded regions* in d.

(iii) We refer to the set of unshaded zones of d, $Z^\circ(d)$, as the *unshaded region of d* and to the set of shaded zones of d, $Z^\bullet(d)$, as the *shaded region of d*.

(iv) Given any region $r \in R(d)$, its *unshaded subregion* is

$$r^\circ = \{z \in r : z \in Z^\circ(d)\} = r \cap Z^\circ(d)$$

and its *shaded subregion* is

$$r^\bullet = \{z \in r : z \in Z^\bullet(d)\} = r \cap Z^\bullet(d).$$

(v) Given any set of labels $L \subseteq \mathcal{L}$, the *Venn zone set on L*, $VZ(L)$, is the set of all possible zones formed using L,

$$VZ(L) = \{(x, L - x) : x \subseteq L\}.$$

(vi) If a diagram d contains all of the possible zones, that is, if $Z(d) = VZ(L(d))$, then d is said to be in *Venn form* or is simply called a *Venn diagram*. The (abstract) zones in the set $VZ(L(d)) - Z(d)$ are said to be *missing from d* and the set $VZ(L(d)) - Z(d)$ is called the *missing zone set* of d, denoted $Z^m(d)$.

(vii) The zone outside all of the contours $(\varnothing, L(d))$ is called the *outside zone* of d although, unlike some authors, we do not assume this zone is present in every ED d.

2.2. *Semantics of Euler diagrams*

The zones and regions in an ED represent sets; missing and shaded zones represent the empty set. The following definitions, modified from those given in [**47**], make this precise. We first interpret the abstract labels, zones and regions as subsets of some universal set U and then define a 'semantics predicate' that captures the conditions that missing and shaded zones represent the empty set. Augmenting EDS increases expressiveness; for example, no unshaded ED can express '$A = \varnothing$'. It also allows us to overcome some drawability problems. For example, the diagrams d_1 and d_2 in Figure 4 each express '$A = B$'. Without the use of shading, we need to use concurrent contours (contours that 'run along one another') to represent this equality as in d_3 in Figure 4. Also, some notations based on EDs, such as spider or constraint diagrams, use shading to represent upper bounds on set cardinalities rather than just denoting the empty set.

DEFINITION 2.4. An *interpretation* is a pair (U, Ψ) where U is a set, called the *universal set*, and $\Psi: \mathcal{L} \cup \mathcal{Z} \cup \mathcal{R} \to \mathbb{P}(U)$ is a function that interprets contour labels, zones and regions as subsets of U such that the images of the zones and regions are completely determined by the images of the contour labels as follows.

(i) For each zone, $(x, y) \in \mathcal{Z}$,

$$\Psi(x, y) = \bigcap_{\ell \in x} \Psi(\ell) \cap \bigcap_{\ell \in y} \overline{\Psi(\ell)}$$
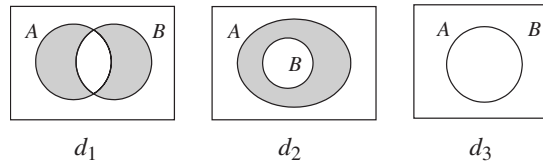
FIGURE 4. *Three Euler diagrams representing $A = B$.*

where $\overline{\Psi(\ell)} = U - \Psi(\ell)$. We also define $\bigcap_{\ell \in \varnothing} \Psi(\ell) = U = \bigcup_{\ell \in \varnothing} \overline{\Psi(\ell)}$.
  (ii) For each region, $r \in \mathcal{R}$,

$$\Psi(r) = \bigcup_{z \in r} \Psi(z)$$

and we take $\bigcup_{z \in \varnothing} \Psi(z) = \varnothing$.

DEFINITION 2.5. Let d be an ED and let $I = (U, \Psi)$ be an interpretation. We define the *semantics predicate* of d, denoted $P_d(I)$, to be the conjunction of the following conditions.
  (i) *Shaded zones condition.* The shaded region of d represents the empty set, $\Psi(Z^\bullet(d)) = \varnothing$.
  (ii) *Missing zones condition.* The missing region of d represents the empty set, $\Psi(Z^m(d)) = \varnothing$.

Both the shaded zones condition and the missing zones condition can be rephrased at the individual (shaded or missing) zone level. This is captured in the following lemma.

LEMMA 2.1. *Let d be an ED and let $I = (U, \Psi)$ be an interpretation.*

Shaded zones condition. The shaded zones condition is equivalent to the following condition:

$$\text{Each shaded zone represents the empty set:} \quad \bigwedge_{z \in Z^\bullet(d)} \Psi(z) = \varnothing.$$

Missing zones condition. The missing zones condition is equivalent to the following condition:

$$\text{Each missing zone represents the empty set:} \quad \bigwedge_{z \in Z^m(d)} \Psi(z) = \varnothing.$$

DEFINITION 2.6. Let d be an ED and $I = (U, \Psi)$ be an interpretation.
  We say I is a *model* for d, denoted $I \models d$, if and only if $P_d(I)$ is true. Also, we say d is *satisfiable* if and only if d has a model.

DEFINITION 2.7. Let d_1 and d_2 be EDs. Then d_1 *semantically entails* d_2 (and d_2 is a *semantic consequence* of d_1), denoted $d_1 \vDash d_2$, if every interpretation that is a model for d_1 is also a model for d_2; that is,

$$d_1 \vDash d_2 \text{ if } I \models d_1 \text{ implies } I \models d_2 \text{ for all interpretations } I.$$

If $d_1 \vDash d_2$ and $d_2 \vDash d_1$, then we say that d_1 and d_2 are *semantically equivalent*, denoted $d_1 \equiv_\vDash d_2$.

3. *Manipulating diagrams*

In this section, we describe ways of manipulating diagrams by adding, moving or removing syntactic elements and we consider the semantic consequences of these manipulations. The

manipulations we introduce here will be used in the next section to obtain a normal form. We are primarily interested in EDs for information representation and reasoning, so we require our diagram manipulations to respect the semantics in the sense that if d_2 is obtained from d_1 by a syntactic manipulation, then $d_1 \vDash d_2$. So, for example, our definition of 'add contour' is more restrictive than the 'insert contour' rule given in [**55**]. Thus, the manipulations described in this section are examples of 'reasoning rules'; see [**47**] for a complete set of reasoning rules for the spider diagram system.

3.1. *Adding and removing syntactic elements*

We define four operations: add and remove a contour label and add and remove a shaded zone. These four operations are sufficient to be able to 'navigate' the semantic equivalence class of a diagram $\langle d \rangle$; any diagram semantically equivalent to d may be obtained from d by applying a sequence of these four syntactic operations. Identifying manipulation rules that navigate semantic equivalence classes and those that 'move between' equivalence classes is likely to be important in the systematic development of reasoning and information presentation systems more generally than the ED system considered here.

We wish to define a syntactic operation that removes a label from a diagram. For $\ell \in \mathcal{L}$, we first define a *remove label* function on the set of abstract zones.

DEFINITION 3.1.
(i) Let $\ell \in \mathcal{L}$ be a contour label. The *remove ℓ* function is defined on the set of abstract zones $\mathcal{Z}$ by
$$r_\ell : \mathcal{Z} \to \mathcal{Z}, \quad r_\ell(a,b) = (a - \{\ell\}, b - \{\ell\}).$$
The function extends naturally to the set of abstract regions $\mathcal{R}$ by defining
$$r_\ell : \mathcal{R} \to \mathcal{R}, \quad r_\ell(r) = \{r_\ell(z) : z \in r\}.$$
(ii) Let d be an ED. The restriction of the domain of r_ℓ to $Z(d)$ is denoted
$$r_{\ell,d} : Z(d) \to \mathcal{Z}.$$
The extension of $r_{\ell,d}$ to regions of d is also denoted $r_{\ell,d}$,
$$r_{\ell,d} : R(d) \to \mathcal{R}, \quad r_{\ell,d}(r) = \{r_{\ell,d}(z) : z \in r\}.$$

EXAMPLE 3.1. Consider the diagram d_1 in Figure 5. Removing the contour labelled B from d_1 results in the diagram d_2. Note that d_2 is not semantically equivalent to d_1; for example, d_1 expresses that $\overline{A} \cap B \cap C \cap E = \varnothing$ (among other things) but d_2 makes no statement about B. Simply to erase the contour B from d_1 without making any other adjustments would produce a diagram in which the zone inside contour E was 'part shaded'. Since there is no notion of a 'partially shaded zone' within our system, in order to obtain a well-formed diagram, this zone is unshaded in d_2.

The zone set of d_1 is
$$Z(d_1) = \{(\{A\}, \{B,C,E\}), (\{A,B\}, \{C,E\}), (\{C\}, \{A,B,E\}), (\{B,C\}, \{A,E\}),$$
$$(\{B,C,E\}, \{A\}), (\{C,E\}, \{A,B\}), (\{B\}, \{A,C,E\}), (\varnothing, \{A,B,C,E\})\}.$$

The zones of d_1 are arranged in pairs with the property that removing B from each zone in a pair produces the same zone in d_2. Thus, the zones of d_2 are
$$Z(d_2) = \{(\{A\}, \{C,E\}), (\{C\}, \{A,E\}), (\{C,E\}, \{A\}), (\varnothing, \{A,C,E\})\}$$
$$= \{r_{B,d_1}(x,y) : (x,y) \in Z(d_1)\}$$
$$= r_{B,d_1}(Z(d_1))$$
and the mapping $r_{B,d_1} : Z(d_1) \to Z(d_2)$ is two-to-one.
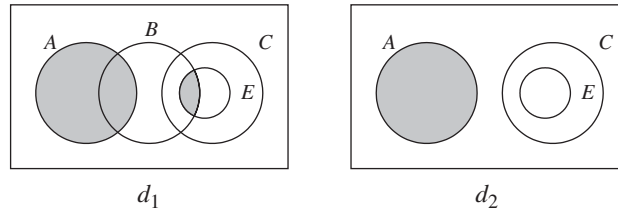
FIGURE 5. *Removing a contour: losing information.*

The zone inside E in d_2, $(\{E\}, \{A, C\})$ corresponds to the pair of zones $(\{B, E\}, \{A, C\})$ and $(\{E\}, \{A, B, C\})$ in d_1. Only one of these zones in d_1 is shaded; since we cannot have partially shaded zones in our diagrams, the corresponding zone in d_2 is left unshaded.

The shaded zones of d_1 and d_2 are

$$Z^{\bullet}(d_1) = \{(\{A\}, \{B, C, E\}), (\{A, B\}, \{C, E\}), (\{B, C, E\}, \{A\})\}$$
$$Z^{\bullet}(d_2) = \{(\{A\}, \{C, E\})\}.$$

Note that if $(x, y) \in Z(d_2)$ is a zone in d_2 then the corresponding zones in d_1 are

$$r_{B,d_1}^{-1}(x, y) = \{(x \cup \{B\}, y), (x, y \cup \{B\})\}.$$

Hence, $(x, y) \in Z(d_2)$ is shaded if and only if *both* the zones in $r_{B,d_1}^{-1}(x, y)$ are shaded in d_1.

In the example above, for each zone $(x, y) \in Z(d_2)$, the inverse image $r_{B,d_1}^{-1}(x, y)$ contains two zones. In general, if d has missing zones and $\ell \in L(d)$, then the set $r_{\ell,d}^{-1}(x, y)$ may contain only a single zone. For example, if we were to remove the contour labelled C from the diagram d_1 in Figure 5, then the zone $(\{A, B\}, \{E\})$ in the resulting diagram, has inverse image

$$r_{C,d_1}^{-1}(\{A, B\}, \{E\}) = \{(\{A, B\}, \{C, E\})\}.$$

This is because the zone $(\{A, B, C\}, \{E\})$ is missing from d_1. The zone $(\{A, B\}, \{E\})$ would be shaded in the diagram obtained from d_1 by removing C since the zone $(\{A, B\}, \{C, E\})$ in the inverse image is shaded in d_1. The general situation is described in the following definition.

DEFINITION 3.2. Let $d = \langle Z(d), Z^{\bullet}(d) \rangle$ be an ED and let $\ell \in \mathcal{L}$. The ED d with ℓ removed, denoted $r_{\ell}(d) = d - \ell$, is $d' = \langle Z(d'), Z^{\bullet}(d') \rangle$ where:
(i) $Z(d') = r_{\ell,d}(Z(d))$; and
(ii) $Z^{\bullet}(d') = \{(x, y) \in Z(d') : r_{\ell,d}^{-1}(x, y) \subseteq Z^{\bullet}(d)\}$.

The following are straightforward consequences of Definition 3.2.

LEMMA 3.1. Let $d = \langle Z(d), Z^{\bullet}(d) \rangle$ be an ED. Then:
(1) $L(r_{\ell}(d)) = L(d) - \{\ell\}$;
(2) if $\ell \in \mathcal{L} - L(d)$, then $r_{\ell,d} : Z(d) \to Z(d)$ is the identity mapping so that $r_{\ell}(d) = d$; and
(3) if $\ell \in L(d)$, then for each zone $(x, y) \in Z(d - \ell)$, $r_{\ell,d}^{-1}(x, y) = r_{\ell}^{-1}(x, y) \cap Z(d)$.

It is straightforward to generalise Definition 3.2 to define removing a set of labels $\mathscr{L} \subset \mathcal{L}$.

DEFINITION 3.3 (Remove set of labels).
(i) Let $\mathscr{L} \subset \mathcal{L}$. The *remove labels in $\mathscr{L}$* function, $r_{\mathscr{L}}$, is defined on the set of abstract zones by

$$r_{\mathscr{L}} : \mathcal{Z} \to \mathcal{Z}, \quad r_{\mathscr{L}}(a, b) = (a - \mathscr{L}, b - \mathscr{L}).$$
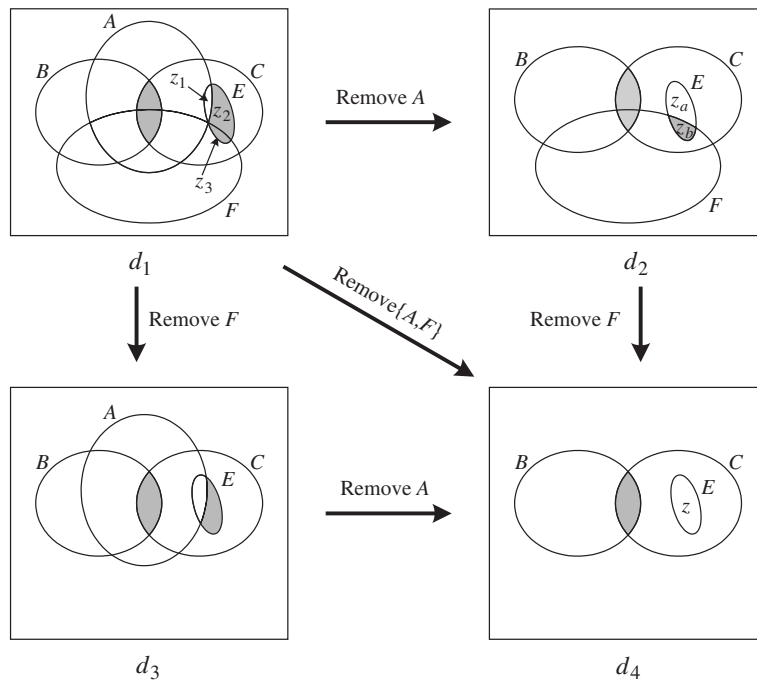
FIGURE 6. *Removing two contours.*

If d is an ED, then the restriction of $r_{\mathscr{L}}$ to $Z(d)$ is denoted

$$r_{\mathscr{L},d} : Z(d) \to \mathcal{Z}.$$

(ii) Let $d = \langle Z(d), Z^{\bullet}(d) \rangle$ be an ED and let $\mathscr{L} \subset \mathcal{L}$. The ED d *with labels in $\mathscr{L}$ removed*, denoted $r_{\mathscr{L}}(d) = d - \mathscr{L}$, is $d' = \langle Z(d'), Z^{\bullet}(d') \rangle$ where:
    (a) $Z(d') = r_{\mathscr{L}}(Z(d))$; and
    (b) $Z^{\bullet}(d') = \{(x,y) \in Z(d') : r_{\mathscr{L},d}^{-1}(x,y) \subseteq Z^{\bullet}(d)\}$.
Thus, $r_{\mathscr{L}}$ defines a function $\mathscr{E}_{\mathrm{shade}} \to \mathscr{E}_{\mathrm{shade}}$.

Theorem 3.2 below shows that removing a pair of labels is equivalent to removing first one label and then the other (in either order). It follows by a simple inductive argument that removing the labels $\mathscr{L} \subseteq L(d)$ is equivalent to removing the labels in $\mathscr{L}$ one at a time (in any order).

Figure 6 illustrates the commutativity of removing labels. We wish to demonstrate the commutativity of the upper triangle in Figure 6; that is, removing A then removing F produces the same diagram as removing $\{A, F\}$. It is easy to verify that the zones in $d - \{A, F\}$ and $(d - A) - F$ agree. We need to verify that the shaded zones are the same in the two diagrams. Consider, for example, the zone $z = (\{C, E\}, \{B\})$ in d_4 in Figure 6. First we consider d_4 as $d_1 - \{A, F\}$. Then $r_{\{A,F\},d_1}^{-1}(z)$ contains three (of the possible four) zones

$$r_{\{A,F\},d_1}^{-1}(z) = \{(\{A, C, E\}, \{B, F\}), (\{C, E\}, \{A, B, F\}), (\{C, E, F\}, \{A, B\})\}$$
$$= \{z_1, z_2, z_3\}.$$

Since only two of these zones are shaded, the zone $z = (\{C, E\}, \{B\})$ is unshaded in $d_4 = d_1 - \{A, F\}$.

Next we consider d_4 as $(d_1 - A) - F$. First, $r_{F,d_1-A}^{-1}(z)$ comprises two zones,

$$r_{F,d_1-A}^{-1}(z) = \{(\{C,E\},\{B,F\}),(\{C,E,F\},\{B\})\} = \{z_a, z_b\}$$

and we need to consider the inverse image under r_{A,d_1} of each of these zones. Now

$$\begin{aligned}
r_{A,d_1}^{-1}(z_a) &= r_{A,d_1}^{-1}(\{C,E\},\{B,F\}) \\
&= \{(\{A,C,E\},\{B,F\}),(\{C,E\},\{A,B,F\})\} \\
&= \{z_1, z_2\},
\end{aligned}$$

but since only one of these zones is shaded in d_1, it follows that $z_a = (\{C,E\},\{B,F\})$ is unshaded in d_2. Also

$$r_{A,d_1}^{-1}(z_b) = r_{A,d_1}^{-1}(\{C,E,F\},\{B\}) = \{(\{C,E,F\},\{A,B\})\} = \{z_3\}$$

and since this zone is shaded in d_1, it follows that $z_b = (\{C,E,F\},\{B\})$ is shaded in d_2. Therefore, only one of the two zones in

$$r_{F,d_1-A}^{-1}(z) = \{(\{C,E,F\},\{B\}),(\{C,E\},\{B,F\})\} = \{z_a, z_b\}$$

is shaded in d_2 so the zone $z = (\{C,E\},\{B\})$ is unshaded in $d_4 = (d_1 - A) - F$.

THEOREM 3.2. *Let d be an ED and let $\ell_1, \ell_2 \in \mathcal{L}$ be two contour labels. Then*

$$r_{\{\ell_1,\ell_2\}}(d) = (r_{\ell_2} \circ r_{\ell_1})(d) = (r_{\ell_1} \circ r_{\ell_2})(d),$$

where $\circ$ denotes composition.

Proof. If $\ell_2 \notin L(d)$, then $r_{\ell_2,d}$ and $r_{\ell_2,d-\ell_1}$ are identity mappings and $r_{\{\ell_1,\ell_2\},d} = r_{\ell_1,d}$. The result then follows. Similarly, if $\ell_1 \notin L(d)$ the result follows. So assume that $\ell_1, \ell_2 \in L(d)$.

As functions defined on abstract zones, $r_{\{\ell_1,\ell_2\}} = r_{\ell_2} \circ r_{\ell_1} : \mathcal{Z} \to \mathcal{Z}$ due to the set identity, $X - \{\ell_1,\ell_2\} = (X - \{\ell_1\}) - \{\ell_2\}$. It follows that

$$Z(d - \{\ell_1,\ell_2\}) = Z((d - \ell_1) - \ell_2) = \{r_{\{\ell_1,\ell_2\}}(z) : z \in Z(d)\}.$$

Now, for an abstract zone $(x, y) \in \mathcal{Z}$, the set $r_{\{\ell_1,\ell_2\}}^{-1}(x, y)$ comprises four zones as follows:

$$\begin{aligned}
r_{\{\ell_1,\ell_2\}}^{-1}(x, y) = \{&(x \cup \{\ell_1,\ell_2\}, y),(x \cup \{\ell_1\}, y \cup \{\ell_2\}), \\
&(x \cup \{\ell_2\}, y \cup \{\ell_1\}),(x, y \cup \{\ell_1,\ell_2\})\} \subseteq \mathcal{Z}.
\end{aligned}$$

We consider which zones are present in the diagrams. For any zone $(x, y) \in Z(d - \{\ell_1,\ell_2\})$,

$$\begin{aligned}
r_{\{\ell_1,\ell_2\},d}^{-1}(x, y) = \{&(x \cup \{\ell_1,\ell_2\}, y),(x \cup \{\ell_1\}, y \cup \{\ell_2\}), \\
&(x \cup \{\ell_2\}, y \cup \{\ell_1\}),(x, y \cup \{\ell_1,\ell_2\})\} \cap Z(d).
\end{aligned}$$

Hence, $(x, y) \in Z(d - \{\ell_1,\ell_2\})$ is shaded if and only if each of the zones

$$(x \cup \{\ell_1,\ell_2\}, y), \quad (x \cup \{\ell_1\}, y \cup \{\ell_2\}), \quad (x \cup \{\ell_2\}, y \cup \{\ell_1\}), \quad (x, y \cup \{\ell_1,\ell_2\})$$

that appears in d is shaded:

$$\{(x \cup \{\ell_1,\ell_2\}, y),(x \cup \{\ell_1\}, y \cup \{\ell_2\}),(x \cup \{\ell_2\}, y \cup \{\ell_1\}),(x, y \cup \{\ell_1,\ell_2\})\} \cap Z(d) \subseteq Z^{\bullet}(d).$$
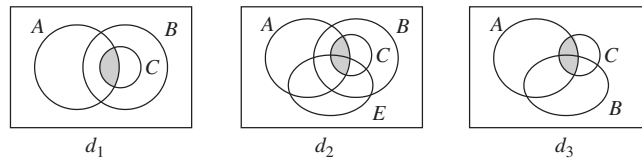
FIGURE 7. *Adding a contour.*

Now we consider the same zone (x, y) but in the diagram $(d - \ell_1) - \ell_2$. We have that $(x, y) \in Z((d - \ell_1) - \ell_2)$ is shaded if the zone(s) in

$$r_{\ell_2, d - \ell_1}^{-1}(x, y) = \{(x \cup \{\ell_2\}, y), (x, y \cup \{\ell_2\})\} \cap Z(d - \ell_1)$$

are all shaded in $d - \ell_1$. We consider each zone in turn. First, $(x \cup \{\ell_2\}, y)$ is shaded in $Z(d - \ell_1)$ if the zones in

$$r_{\ell_1, d}^{-1}(x \cup \{\ell_2\}, y) = \{(x \cup \{\ell_1, \ell_2\}, y), (x \cup \{\ell_2\}, y \cup \{\ell_1\})\} \cap Z(d)$$

are shaded. Similarly, $(x, y \cup \{\ell_2\})$ is shaded in $Z(d - \ell_1)$ if the zones in

$$r_{\ell_1, d}^{-1}(x, y \cup \{\ell_2\}) = \{(x \cup \{\ell_1\}, y \cup \{\ell_2\}), (x, y \cup \{\ell_1, \ell_2\})\} \cap Z(d)$$

are shaded.

Therefore, the zone (x, y) is shaded in $(d - \ell_1) - \ell_2$ if and only if all of the zones in

$$\begin{aligned} r_{\ell_1, d}^{-1}(r_{\ell_2, d - \ell_1}^{-1}(x, y)) = \{ &(x \cup \{\ell_1, \ell_2\}, y), (x \cup \{\ell_2\}, y \cup \{\ell_1\}), \\ &(x \cup \{\ell_1\}, y \cup \{\ell_2\}), (x, y \cup \{\ell_1, \ell_2\})\} \cap Z(d) \end{aligned}$$

are shaded.

This is precisely the condition for (x, y) to be shaded when considered as a zone of $d - \{\ell_1, \ell_2\}$. Therefore, the shaded zones of $d - \{\ell_1, \ell_2\}$ and $(d - \ell_1) - \ell_2$ agree. $\square$

We may add a contour that is not already present to a diagram. In order to obtain a semantically equivalent diagram, the new contour will be added in such a way that it splits each existing zone into two new zones. Adding the contour labelled E to the diagram d_1 in Figure 7 produces d_2.

DEFINITION 3.4. Let d be an ED and let $\ell \in \mathcal{L}$ be a contour label not in d, $\ell \notin L(d)$. The ED d with ℓ *added*, denoted $d + \ell$, is $d' = \langle Z(d'), Z^\bullet(d') \rangle$ where:
  (i) $L(d') = L(d) \cup \{\ell\}$;
  (ii) $Z(d') = \{(x \cup \{\ell\}, y) : (x, y) \in Z(d)\} \cup \{(x, y \cup \{\ell\}) : (x, y) \in Z(d)\}$; and
  (iii) $Z^\bullet(d') = \{(x \cup \{\ell\}, y) : (x, y) \in Z^\bullet(d)\} \cup \{(x, y \cup \{\ell\}) : (x, y) \in Z^\bullet(d)\}$.

Note that the operations 'add ℓ' and 'remove ℓ' do not in general commute. If we first add ℓ to d (where $\ell \notin L(d)$) and then remove ℓ from the resulting diagram, we obtain d: $(d + \ell) - \ell = d$. However, if we remove ℓ from d' (where $\ell \in L(d')$) and then add ℓ to the resulting diagram, we do not always obtain d': $(d' - \ell) + \ell \neq d'$, in general. For example, in Figure 7, $(d_1 - B) + B$ produces the diagram d_3 which is different (syntactically and semantically) from d_1.

In terms of the semantics of the ED system, both shaded and missing zones denote the empty set. This allows for a variety of representations of set-theoretic relationships. Systems based on EDS may allow users a choice of representation. Hence, it is desirable, at the syntactic level, to be able to move between different representations of the empty set; that is, to introduce a missing zone into a diagram as a shaded zone and to delete a shaded zone from a diagram.
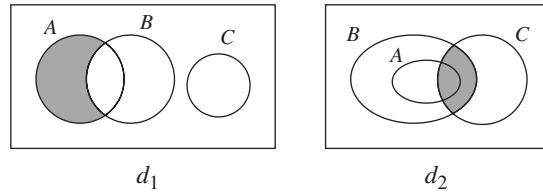
FIGURE 8. *Adding and removing zones.*

For example, in Figure 8, the diagram d_2 can be obtained from d_1 by removing the shaded zone $(\{A\}, \{B, C\})$ and adding two new shaded zones $(\{A, B, C\}, \varnothing)$ and $(\{B, C\}, \{A\})$ as shaded zones.

DEFINITION 3.5. Let d be an ED and let $z \in Z^m(d)$ be a missing zone of d. The ED d *with z added*, denoted $d + z$, is d' where:
  (i) $L(d') = L(d)$;
  (ii) $Z(d') = Z(d) \cup \{z\}$; and
  (iii) $Z^\bullet(d') = Z^\bullet(d) \cup \{z\}$.

Removing a shaded zone is complicated by the fact that it may result in the removal of one or more labels from the diagram. For example, referring back to Figure 5, removing the shaded zone $(\{A\}, \{C, E\})$ from d_2 has the effect of removing A entirely from the diagram. This is because the removed zone is the only zone in the diagram that contains the contour label A. The resulting diagram is not semantically equivalent to d_2 since it does not assert $\Psi(A) = \varnothing$.

DEFINITION 3.6. Let d be an ED and let $z \in Z^\bullet(d)$ be a shaded zone of d. Let $\mathscr{L} \subseteq in(z)$ be the set of those labels ℓ for which the zone z is the only zone in d that has ℓ in its containing set. The ED d *with z removed*, denoted $d - z$, is d' where:
  (i) $L(d') = L(d) - \mathscr{L}$;
  (ii) $Z(d') = \{(x, y - \mathscr{L}) : (x, y) \in Z(d) - \{z\}\}$; and
  (iii) $Z^\bullet(d') = \{(x, y - \mathscr{L}) : (x, y) \in Z^\bullet(d) - \{z\}\}$.

If $\mathscr{L} = \varnothing$ (so that no labels are removed from d), then $Z(d') = Z(d) - \{z\}$ and $Z^\bullet(d') = Z^\bullet(d) - \{z\}$. Also, if $\mathscr{L} \neq \varnothing$, then $d - z$ is the same diagram as that obtained by removing the label set $\mathscr{L}$, $d - \mathscr{L}$.

We require our diagram manipulations to be valid in the sense that if we manipulate d_1 to obtain d_2, then d_2 is a semantic consequence of d_1, $d_1 \vDash d_2$. Each of the four manipulations described above is valid in this sense although it is not always the case that applying one of the manipulations produces a semantically equivalent diagram.

THEOREM 3.3. *Let d be an ED.*
  (i) *If $\ell \in L(d)$ is a contour label in d, then $d \vDash d - \ell$.*
  (ii) *If $\ell \in \mathcal{L} - L(d)$ is a contour label not in d, then $d \equiv_\vDash d + \ell$.*
  (iii) *If $z \in Z^m(d)$ is a zone missing from d, then $d \equiv_\vDash d + z$.*
  (iv) *If $z \in Z^\bullet(d)$ is a shaded zone, then $d \vDash d - z$.*

Proof. The proofs of parts (i), (ii) and (iii), in the context of spider diagrams, can be found in [**47**] where the emphasis is on reasoning with the system of diagrams. We provide a sketch of the proofs of parts (ii) and (iv).

(ii) Let ℓ be a contour label not in d. For each zone $(x, y) \in \mathcal{Z}$, $(x, y) \in Z(d)$ if and only if the zones $(x \cup \{\ell\}, y), (x, y \cup \{\ell\}) \in Z(d + \ell)$. Furthermore, (x, y) is shaded in (respectively, missing from) d if and only if $(x \cup \{\ell\}, y)$ and $(x, y \cup \{\ell\})$ are both shaded in (respectively, missing from) $d + \ell$.

For any interpretation (U, Ψ), $\Psi(x \cup \{\ell\}, y) = \Psi(x, y) \cap \Psi(\ell)$ and $\Psi(x, y \cup \{\ell\}) = \Psi(x, y) \cap \overline{\Psi(\ell)}$. Hence, $\Psi(x \cup \{\ell\}, y) \cup \Psi(x, y \cup \{\ell\}) = \Psi(x, y)$. Therefore, $\Psi(x, y) = \varnothing$ if and only if $\Psi(x \cup \{\ell\}, y) = \varnothing$ and $\Psi(x, y \cup \{\ell\}) = \varnothing$. It follows that the shaded zones condition and missing zones condition for d and for $d + \ell$ are equivalent.

(iv) Let z be a shaded zone of d.

If removing z from d does not remove any labels from d, then it is the reverse operation to adding a missing zone, $(d - z) + z = d$. Hence, $d - z \equiv_{\models} d$ in this case by part (iii).

Now suppose that removing z also removes all of the contour labels in some non-empty set $\mathcal{L}$. Then z is of the form $z = (x' \cup \mathcal{L}, y')$ where $x' \cup y' \cup \mathcal{L} = L(d)$. Furthermore, z is the only zone in d for which $\mathcal{L} \subseteq in(z)$. Note also that the zone (x', y') is missing in $d - z$.

Let (U, Ψ) be an interpretation that is a model for d. Note that, if (x, y) is a zone of $d - z$, then

$$\Psi(x, y) = \bigcup_{\{P, Q\}} \Psi(x \cup P, y \cup Q)$$

where the union is taken over all partitions $\{P, Q\}$ of $\mathcal{L}$. Hence, if $\Psi(x \cup P, y \cup Q) = \varnothing$ for all partitions of $\mathcal{L}$, then $\Psi(x, y) = \varnothing$. We use this observation several times in what follows.

Let (x, y) be a shaded zone of $d - z$. Then $(x, y \cup \mathcal{L})$ is a shaded zone in d. Every zone of the form $(x \cup P, y \cup Q)$ where $\{P, Q\}$ is a partition of $\mathcal{L}$ and $P \neq \varnothing$ is missing from d. Therefore, by the missing and shaded zones conditions for d, $\Psi(x \cup P, y \cup Q) = \varnothing$ for every partition $\{P, Q\}$ of $\mathcal{L}$. Hence, $\Psi(x, y) = \varnothing$ so (U, Ψ) satisfies the shaded zones condition for $d - z$.

Let (x, y) be a missing zone of $d - z$. Then either $(x, y) = (x', y')$ or $(x, y \cup \mathcal{L})$ is missing in d. In the first case, every zone of the form $(x' \cup P, y' \cup Q)$ where $\{P, Q\}$ is a partition of $\mathcal{L}$ and $P \neq \mathcal{L}$ is missing from d. Also $z = (x' \cup \mathcal{L}, y')$ is shaded in d. Therefore, by the missing and shaded zones conditions for d, $\Psi(x' \cup P, y' \cup Q) = \varnothing$ for every partition $\{P, Q\}$ of $\mathcal{L}$. Hence, $\Psi(x', y') = \varnothing$.

In the second case, every zone of the form $(x \cup P, y \cup Q)$ where $\{P, Q\}$ is a partition of $\mathcal{L}$ and $P \neq \varnothing$ is missing from d. Therefore, since $(x, y \cup \mathcal{L})$ is also missing, every zone of the form $(x \cup P, y \cup Q)$ where $\{P, Q\}$ is a partition of $\mathcal{L}$ is missing from d. Hence, $\Psi(x', y') = \varnothing$ by the missing zones condition for d. Therefore, (U, Ψ) satisfies the missing zones condition for $d - z$.

Hence, (U, Ψ) is a model for $d - z$.   □

3.2. *Nomads*

If every zone 'inside' a contour is shaded, the diagram asserts that the set assigned to the label of the contour is empty.

EXAMPLE 3.2. In Figure 9, each of the diagrams d_1, d_2, d_3 and d_4 asserts:

$$A \cap C = \varnothing \quad \text{and} \quad E = \varnothing.$$

Essentially, the only difference between the diagrams is the placing of the entirely shaded contour labelled E within the diagram. Note that the number of zones that lie inside the contour E also differs between the diagrams: E contains 2, 1, 1 and 4 zones in d_1, d_2, d_3 and d_4, respectively.

Provided there is at least one zone inside E and all of the zones inside E are shaded, there are many diagrams that are semantically equivalent to those in Figure 9 with different placing
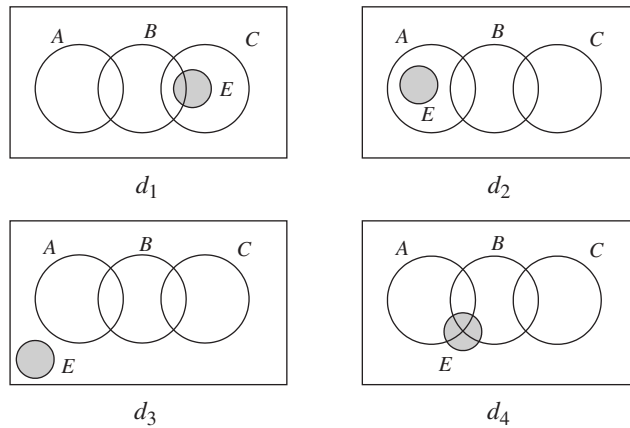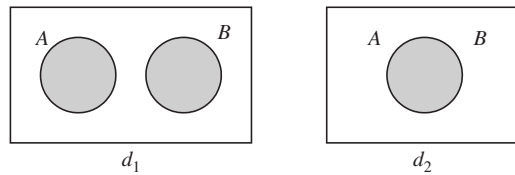
FIGURE 9. *Four diagrams where E is a 'nomad'.*

FIGURE 10. *Moving nomads.*

of the contour E. Informally, we can think of E as being 'free to wander around' the diagram and, following John [**54**], we refer to the contour E as a '*nomad*'.

Of these four diagrams, d_2 and d_3 have the fewest zones. It appears that d_3 is 'visually simpler' than d_2 in the sense that the assertion '$E = \varnothing$' is represented visually by the shaded contour labelled E in a way that is 'separated from' the other contours of the diagram.

The following definition, which is equivalent to that in [**54**], formalises the notion of a nomad.

DEFINITION 3.7. Let d be an ED and $\ell \in L(d)$ be a label. We say ℓ is a *nomad* in d if the set of zones within ℓ is entirely shaded,

$$\{(x,y) \in Z(d) : \ell \in x\} \subseteq Z^{\bullet}(d).$$

We denote the set of nomads in d by $\mathcal{N} = \mathcal{N}(d)$.

EXAMPLE 3.3. Consider the diagram d_1 in Figure 10 which has two separated nomads labelled A and B. In d_2, these contours are concurrent. The diagrams d_1 and d_2 are semantically equivalent. Conflating the contour label with the set it represents, both diagrams assert:

$$A \cap \overline{B} = \varnothing \quad \text{and} \quad \overline{A} \cap B = \varnothing; \tag{1}$$

$$A \cap B = \varnothing. \tag{2}$$

The equations in (1) follow from the shaded zones condition for d_1 and the missing zones condition for d_2. Similarly, equation (2) follows from the missing zones condition for d_1 and the shaded zones condition for d_2. The conditions in (1) and (2) are equivalent to the assertion $A = \varnothing$ and $B = \varnothing$.
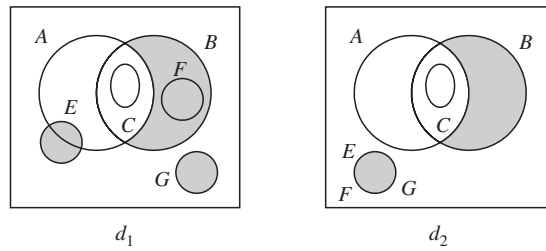
FIGURE 11. *Moving several nomads.*

Combining the ideas in Examples 3.2 and 3.3, an ED containing several nomads is semantically equivalent to a diagram where all of the nomads have been moved to the outside zone and 'overlaid' to form a single shaded zone. This is illustrated in Figure 11 where diagram d_1 has three nomads, the contours labelled E, F and G. In the semantically equivalent d_2, these three contours form a single shaded zone situated in the outside zone. We refer to this as *exiling the nomads* and we say that d_2 has *exiled nomads*.

DEFINITION 3.8. Let d be an ED with a non-empty set of nomads, $\mathcal{N}(d) \neq \varnothing$.
(i) The set of zones $Z(d)$ partitions into the zones inside nomads

$$Z_{in_{\mathcal{N}}}(d) = \{(x, y) \in Z(d) : x \cap \mathcal{N} \neq \varnothing\}$$

and the set of zones outside nomads

$$Z_{out_{\mathcal{N}}}(d) = \{(x, y) \in Z(d) : x \cap \mathcal{N} = \varnothing\}.$$

The set of shaded zones also partitions in a similar way into the two sets $Z_{in_{\mathcal{N}}}^{\bullet}(d)$ and $Z_{out_{\mathcal{N}}}^{\bullet}(d)$.
(ii) The ED d *with exiled nomads* is d' where:
(a) the unshaded zones of d' are the same as the unshaded zones of d,

$$Z^{\circ}(d') = Z^{\circ}(d);$$

(b) the shaded zones in d_2 are the shaded zones outside nomads in d together with a single additional shaded zone inside all of the nomads,

$$Z^{\bullet}(d') = Z_{out_{\mathcal{N}}}^{\bullet}(d) \cup \{(\mathcal{N}(d), L(d) - \mathcal{N}(d))\}.$$

THEOREM 3.4. *Let d be an ED with a non-empty set of nomads, $\mathcal{N}(d) \neq \varnothing$, and let d' be d with exiled nomads. Then d is semantically equivalent to d', $d \equiv_{\vDash} d'$.*

Proof. Since the sets of nomads in d and d' are equal, we will write $\mathcal{N}$ for either $\mathcal{N}(d)$ or $\mathcal{N}(d')$.
Suppose that (U, Ψ) is a model for d. The missing zones of d' are

$$Z^m(d') = Z^m(d) \cup (Z_{in_{\mathcal{N}}}^{\bullet}(d) - \{(\mathcal{N}, L(d) - \mathcal{N}(d))\}).$$

Since each missing zone of d' is either missing or shaded in d, the missing zones condition for d' follows from the shaded zones and missing zones conditions for d.
The shaded zones condition for d implies that $\Psi(n) = \varnothing$ for each nomad $n \in \mathcal{N}$. Since $\Psi(\mathcal{N}, L(d) - \mathcal{N}(d)) \subseteq \Psi(n)$ (for any nomad n), we have $\Psi(\mathcal{N}, L(d) - \mathcal{N}(d)) = \varnothing$. The other shaded zones in d' are those in $Z_{out_{\mathcal{N}}}^{\bullet}(d)$. Hence, the shaded zones condition follows for d'.
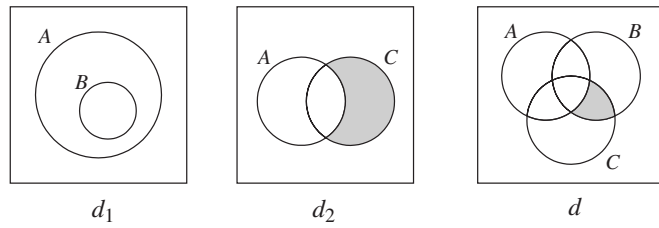
FIGURE 12. *Capturing the semantic information common to two diagrams.*

Therefore, (U, Ψ) is a model for d', so $d \vDash d'$.

Conversely, suppose that (U, Ψ) is a model for d'. The missing zone set of d satisfies $Z^m(d) \subseteq Z^m(d') \cup \{(\mathcal{N}, L(d) - \mathcal{N}(d))\}$. Therefore, the missing zones condition for d follows from the missing zones and shaded zones conditions for d', since $(\mathcal{N}, L(d) - \mathcal{N}(d))$ is shaded in d'.

The shaded zone set of d is $Z^\bullet(d) = Z^\bullet_{in_\mathcal{N}}(d) \cup Z^\bullet_{out_\mathcal{N}}(d)$. Now $Z^\bullet_{out_\mathcal{N}}(d) = Z^\bullet_{out_\mathcal{N}}(d')$ so $\Psi(z) = \varnothing$ for any zone $z \in Z^\bullet_{out_\mathcal{N}}(d)$. For any zone $z \in Z^\bullet_{in_\mathcal{N}}(d)$, z lies inside some nomad n so $\Psi(z) \subseteq \Psi(n)$. But the missing zones and shaded zones conditions for d' imply that $\Psi(n) = \varnothing$. Hence, $\Psi(z) = \varnothing$. Therefore, the shaded zones condition holds for d.

Hence, (U, Ψ) is a model for d', so $d' \vDash d$.

Therefore, $d \equiv_\vDash d'$.   □

3.3. *Capturing common semantic information*

Let d_1 and d_2 be two EDs. If d_1 and d_2 are not semantically equivalent, we wish to define a diagram that captures the semantic information that is common to both d_1 and d_2. As a simple, example consider the diagrams d_1 and d_2 in Figure 12. Using (non-italic) X informally to denote the set represented by contour X, d_1 asserts that B $\subseteq$ A and d_2 asserts that A $\cap \overline{\text{C}} = \varnothing$. The second assertion is equivalent to C $\subseteq$ A so the semantic information common to both diagrams is B $\cap$ C $\subseteq$ A. This common information is asserted by diagram d.

For simplicity, we define the 'common information diagram' to be a Venn diagram with label set incorporating all of the labels of d_1 and d_2. A zone of the common information diagram will be shaded only when *both* of the corresponding zones in d_1 and d_2 represent the empty set (by being either shaded or missing from the diagram). The formal definition is the following.

DEFINITION 3.9. Let d_1 and d_2 be two EDs. The *semantic coproduct* of d_1 and d_2 is the ED $d = d_1 \sqcup d_2$ defined as follows:
  (i) $L(d) = L(d_1) \cup L(d_2)$;
  (ii) d is in Venn form: $Z(d) = VZ(L(d))$;
 (iii) a zone $(x, y) \in Z(D)$ is shaded if and only if both $(x \cap L(d_1), y \cap L(d_1)) \in Z^\bullet(d_1) \cup Z^m(d_1)$ and $(x \cap L(d_2), y \cap L(d_2)) \in Z^\bullet(d_2) \cup Z^m(d_2)$:

$$Z^\bullet(d) = \{(x, y) \in Z(d) : (x \cap L(d_i), y \cap L(d_i)) \in Z^\bullet(d_i) \cup Z^m(d_i), \ i = 1, 2\}.$$

In Figure 12, the zone $(\{B, C\}, \{A\})$ is shaded in the semantic coproduct d because the zone $(\{B\}, \{A\}) = (\{B, C\} \cap L(d_1), \{A\} \cap L(d_1))$ is missing from d_1 and the zone $(\{C\}, \{A\}) = (\{B, C\} \cap L(d_2), \{A\} \cap L(d_2))$ is shaded in d_2.

The terminology for $d_1 \sqcup d_2$ is motivated by the following theorem.

THEOREM 3.5. Let d_1, and d_2 be two EDs. The semantic coproduct $d_1 \sqcup d_2$ satisfies the following properties:
  (i) $d_1 \vDash d_1 \sqcup d_2$ and $d_2 \vDash d_1 \sqcup d_2$;
  (ii) if d is an ED such that $d_1 \vDash d$ and $d_2 \vDash d$, then $d_1 \sqcup d_2 \vDash d$.
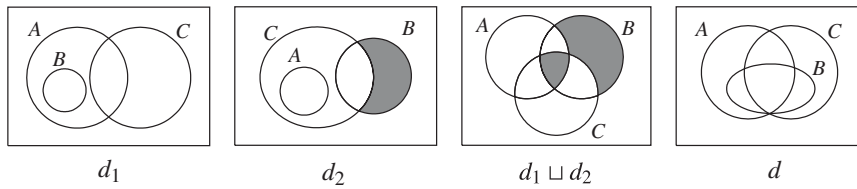
FIGURE 13. *Illustrating the semantic coproduct.*

Proof. In each part of the proof we will use the observation that, for Venn diagrams with the same contour labels, $d_a \vDash d_b$ if and only if every shaded zone in d_b is also shaded in d_a, $Z^\bullet(d_b) \subseteq Z^\bullet(d_a)$.

(i) We begin by replacing d_1 with a diagram d_1^V which is in Venn form and has the same label set as $d_1 \sqcup d_2$. We do this in two stages. First, we add the missing zones of d_1 to form a Venn diagram and, second, we add the contour labels in $L(d_2) - L(d_1)$ to the resulting diagram; the result is d_1^V. By repeated applications first of Theorem 3.3(iii) and second of Theorem 3.3(ii), it follows that $d_1 \equiv_\vDash d_1^V$.

Now $d_1 \sqcup d_2$ and d_1^V are Venn diagrams with the same label set and every shaded zone in $d_1 \sqcup d_2$ is also shaded in d_1^V. By the observation above, it follows that $d_1^V \vDash d_1 \sqcup d_2$ and therefore $d_1 \vDash d_1 \sqcup d_2$. A similar proof clearly works for d_2.

(ii) Suppose that d is an ED such that $d_1 \vDash d$ and $d_1 \vDash d$.

We modify each of d_1, d_2 and d to give a semantically equivalent Venn diagram where all of the modified diagrams have the same contour labels. For example, first add the missing zones of d_1 to form a Venn diagram and then add the contour labels in $(L(d) \cup L(d_2)) - L(d_1)$ to form a new diagram d_1^*. The diagrams d_2^* and d^* are formed similarly. These diagrams satisfy the following:

$$d_1 \equiv_\vDash d_1^*, \quad d_2 \equiv_\vDash d_2^*, \quad d \equiv_\vDash d^* \quad \text{and} \quad L(d_1^*) = L(d_2^*) = L(d^*).$$

Add the contour labels in $L(d) - (L(d_1) \cup L(d_2))$ to $d_1 \sqcup d_2$ to give a semantically equivalent diagram $(d_1 \sqcup d_2)^* = d_1^* \sqcup d_2^*$.

Since $d_1^* \vDash d^*$ and $d_2^* \vDash d^*$ it follows by the observation above that every shaded zone in d^* is also shaded in both d_1^* and d_2^*. Therefore, by definition of the semantic coproduct, every shaded zone in d^* is also shaded in $d_1^* \sqcup d_2^*$. Hence, $d_1^* \sqcup d_2^* \equiv_\vDash d^*$ so $d_1 \sqcup d_2 \equiv_\vDash d$ as required. □

COROLLARY 3.6. *Let d_1 and d_2 be two semantically equivalent EDs. Then each is semantically equivalent to $d_1 \sqcup d_2$.*

Figure 13 illustrates Theorem 3.5. Note that $d_1 \vDash d$ and $d_2 \vDash d$ since, informally, d asserts less information than that asserted by each of d_1 and d_2. For example, d_1 asserts $B \subseteq A - C$ and d_2 asserts $A \cap B = \varnothing$ and $A \cup B \subseteq C$. The diagram d makes the assertion $B \subseteq A \cup C$ which is weaker than that made either by d_1 or d_2.

In d_1 the missing zones are $(\{A, B, C\}, \varnothing)$, $(\{B\}, \{A, C\})$ and $(\{B, C\}, \{A\})$. In d_2 there are missing zones $(\{A, B, C\}, \varnothing)$, $(\{A, B\}, \{C\})$ and $(\{A\}, \{B, C\})$ and a single shaded zone $(\{B\}, \{A, C\})$. Therefore, their semantic coproduct $d_1 \sqcup d_2$ is a Venn diagram whose shaded zones are those that are shaded or missing in both d_1 and d_2, namely $(\{A, B, C\}, \varnothing)$ and $(\{B\}, \{A, C\})$. Hence, $d_1 \sqcup d_2$ asserts $A \cap B \cap C = \varnothing$ and $\overline{A} \cap B \cap \overline{C} = \varnothing$. The second of these is equivalent to $B \subseteq A \cup C$ which is what the diagram d asserts. Hence, $d_1 \sqcup d_2 \vDash d$.

4. The normal form

In this section we develop our normal form for EDs. We first present a syntactic characterisation of when a contour label is semantically redundant and may therefore be removed from a diagram whilst remaining within the same semantic equivalence class. Then we use the transformations introduced in the §3 to obtain, from a given diagram d, a semantically equivalent diagram d^N that has a simple syntactic description and where the syntactic properties of d^N will define our normal form.

4.1. Redundant contour labels

DEFINITION 4.1. Let d be an ED. A contour label $\ell \in L(d)$ is *redundant* in d if removing it results in a semantically equivalent diagram, $d \equiv_\models d - \ell$. More generally, a set of contour labels $\mathscr{L} \subseteq L(d)$ is *redundant* in d if removing it results in a semantically equivalent diagram, $d \equiv_\models d - \mathscr{L}$.

THEOREM 4.1. *Let d be an ED and let $\ell_1, \ell_2 \in L(d)$ be two labels in d. Then $\{\ell_1, \ell_2\}$ is redundant in d if and only if ℓ_1 is redundant in d and ℓ_2 is redundant in $d - \ell_1$.*

Proof. Suppose that $\{\ell_1, \ell_2\}$ is redundant in d. Then $d - \{\ell_1, \ell_2\} \equiv_\models d$. Clearly ℓ_1 is redundant in d, so $d - \ell_1 \equiv_\models d$. From Theorem 3.2, we have $d - \{\ell_1, \ell_2\} = (d - \ell_1) - \ell_2$ so $(d - \ell_1) - \ell_2 \equiv_\models d \equiv_\models d - \ell_1$. Therefore, ℓ_2 is redundant in $d - \ell_1$.

Conversely, suppose that ℓ_1 is redundant in d and ℓ_2 is redundant in $d - \ell_1$. Then $d \equiv_\models (d - \ell_1) \equiv_\models (d - \ell_1) - \ell_2 = d - \{\ell_1, \ell_2\}$. Therefore, $\{\ell_1, \ell_2\}$ is redundant in d.   □

We wish to find syntactic conditions for a contour label to be redundant in a diagram. By Theorem 3.3, adding a contour label ℓ to a diagram d produces a semantically equivalent diagram $d + \ell$; thus ℓ is redundant in $d + \ell$. For example, in Figure 7, the contour labelled E is redundant in $d_2 = d_1 + E$. When adding E to d_1, each zone in d_1 is split in two. Hence, in d_2, every zone that is contained by E has a corresponding zone, which (following [54]) we call its 'E-twin', that is excluded by E. Furthermore, the E-twin of every shaded zone is also shaded. We say that the contour E 'completely splits' both the shaded and unshaded regions of d_2. We now make these ideas precise for abstract diagrams.

DEFINITION 4.2.
(i) For $\ell \in \mathcal{L}$, we define the *move label* function on the set of abstract zones by

$$m_\ell : \mathcal{Z} \to \mathcal{Z}, \ m_\ell(x, y) = \begin{cases} (x - \{\ell\}, y \cup \{\ell\}) & \text{if } \ell \in x, \\ (x \cup \{\ell\}, y - \{\ell\}) & \text{if } \ell \in y, \\ (x, y) & \text{otherwise.} \end{cases}$$

(ii) A pair of distinct zones $z_1, z_2 \in \mathcal{Z}$ are called *ℓ-twins* if $z_2 = m_\ell(z_1)$. Thus, a pair of distinct zones are ℓ-twins if and only if they are of the form $(x - \{\ell\}, y \cup \{\ell\})$ and $(x \cup \{\ell\}, y - \{\ell\})$.
(iii) Let d be an ED. A zone $z \in Z(d)$ is an *ℓ-single in d* if its ℓ-twin does not belong to d.
(iv) Let $r \in R(d)$ be a region in an ED d and let $\ell \in L(d)$. We say that ℓ *completely splits* r if, for each zone $z \in r$, its ℓ-twin is also in r, $m_\ell(z) \in r$.

The presence or absence of ℓ-twins in a diagram is important in determining whether ℓ is redundant but not, perhaps, in the most obvious way. Since $d \equiv_\models d + \ell$ (where ℓ is not a
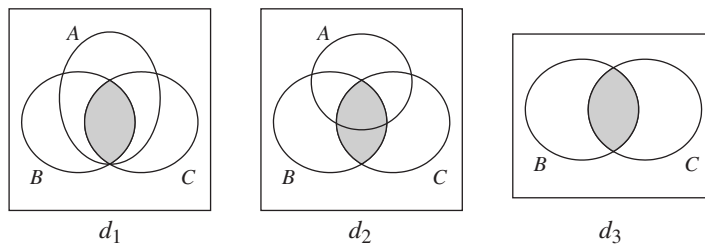
Figure 14. *Missing twins.*

label in d), it follows that ℓ is redundant in $d + \ell$ and, by construction, ℓ completely splits both the shaded and unshaded regions of $d + \ell$. Therefore, an obvious initial conjecture is that completely splitting both the shaded and unshaded regions is the syntactic condition for contour redundancy. However, this is not correct.

First, nomads can complicate the picture. Consider the four diagrams containing a nomad in Figure 9. In each diagram, the contour labelled B is redundant. Although B completely splits the unshaded region in each diagram, B does not completely split the shaded regions in d_2 or d_3. Similarly in Figure 14 below, the contour labelled A is redundant in both d_1 and d_2; removing A from either of these diagrams gives the semantically equivalent diagram d_3. Although A completely splits the shaded and unshaded regions in d_2, it does not split the shaded region in d_1 because the zone $(\{B, C\}, \{A\})$, which is the A-twin of the shaded zone, is missing in d_1. These examples suggest that we only need consider the unshaded zones to determine whether a contour is redundant.

DEFINITION 4.3. Let d be an ED. A label $\ell \in L(d)$ is a *splitting label* for d if ℓ completely splits $Z^\circ(d)$, the unshaded region of d.

The following lemma captures straightforward consequences of Definition 4.3.

LEMMA 4.2. *Let d be an ED.*
(i) *If $\ell \in L(d)$ is a splitting label for d, then for each shaded zone $z \in Z^\bullet(d)$, its ℓ-twin $m_\ell(z)$ is either shaded or missing from d.*
(ii) *If $n \in L(d)$ is a nomad in d, then n is a splitting label for d if and only if d is entirely shaded.*

As the following theorem shows, the notion of splitting label is the syntactic characterisation we are seeking for a contour label to be redundant.

THEOREM 4.3. *A contour label ℓ is redundant in an ED d if and only if ℓ is a splitting label for d.*

COROLLARY 4.4. *Let d be an ED with an odd number of unshaded zones. Then d has no redundant contours.*

Before embarking on the proof of Theorem 4.3, we illustrate the approach in establishing the implication 'if ℓ is not a splitting label then it is not redundant' through the following example.

EXAMPLE 4.1. Suppose that ℓ is not a splitting label in d. Then it does not split the unshaded region $Z^\circ(d)$ so there is an unshaded zone in $Z^\circ(d)$ such that its ℓ-twin is either shaded or
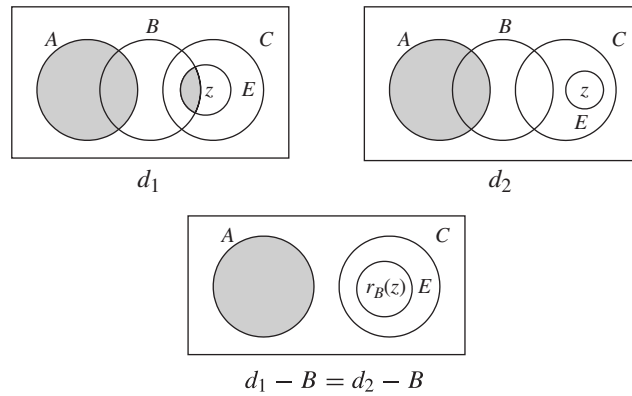
$$d_1$$

$$d_2$$

$$d_1 - B = d_2 - B$$

FIGURE 15. *The contour B does not split the unshaded region.*

missing, $m_\ell(z) \in Z^\bullet(d)$ or $m_\ell(z) \in Z^m(d)$. These two possibilities are illustrated in Figure 15 for the zone $z = (\{C, E\}, \{A, B\})$, where the contour labelled B does not split the unshaded region. In d_1, the B-twin of z is shaded and in d_2 this B-twin is missing.

To show that $d_1 - B$ is not semantically equivalent to d_1, we construct a model for $d_1 - B$ for which $\Psi(m_B(z)) \neq \varnothing$; this is therefore not a model for d_1 since $\Psi(m_B(z))$ is empty in any model for d_1. We start with a model for $d_1 - B$ in which $\Psi(r_B(z))$ is non-empty. To construct such a model, take the universal set U to be the set of positive integers $\mathbb{Z}^+$ and define a set assignment $\Psi : \mathcal{L} \to U$ by

$$\Psi(A) = \varnothing, \quad \Psi(C) = \{1, 2\}, \quad \Psi(E) = \{2\}$$

and

$$\Psi(\ell) = \mathbb{Z}^+ \quad \text{for all } \ell \in \mathcal{L} - L(d_1 - B) = \mathcal{L} - \{A, C, E\}.$$

To obtain this model, we listed, as $z_1, z_2, \ldots$, the unshaded zones within any contour and then defined Ψ such that $\Psi(z_i) = \{i\}$ and $\Psi(z^\bullet) = \varnothing$ for any shaded zone $z^\bullet$.

It is easy to verify that this interpretation $(\mathbb{Z}^+, \Psi)$ is a model for $d_1 - B$ but it is not quite the model we need. Since $B \in \mathcal{L} - \{A, C, E\}$, we defined $\Psi(B) = \mathbb{Z}^+$. We now redefine $\Psi(B)$ as follows (but without changing $\Psi(\ell)$ for $\ell \neq B$):

$$\Psi(B) = \Psi(r_B(z)) = \Psi(C) \cap \Psi(E) \cap \overline{\Psi(A)} = \{1, 2\} \cap \{2\} \cap \mathbb{Z}^+ = \{2\}.$$

This new interpretation $(\mathbb{Z}^+, \Psi)$ is still a model for $d_1 - B$ since neither the shaded zones condition nor the missing zones condition in $d_1 - B$ involve $\Psi(B)$.

However, in d_1, $m_B(z) = (\{B, C, E\}, \{A\})$ so

$$\Psi(m_B(z)) = \Psi(B) \cap \Psi(C) \cap \Psi(E) \cap \overline{\Psi(A)} = \{2\} \cap \{1, 2\} \cap \{2\} \cap \mathbb{Z}^+ = \{2\}.$$

Since $\Psi(m_B(z)) \neq \varnothing$ but $m_B(z)$ is shaded in d_1, this interpretation is not a model for d_1 (or d_2). Therefore, $d_1 \not\models\models d_1 - B$ so B is not redundant in d_1.

Before we prove Theorem 4.3, we establish the following technical results on the relationship between the interpretation of a zone z and the related zones $r_\ell(z)$ and $m_\ell(z)$.

LEMMA 4.5. *Let d be an ED and let (U, Ψ) be a model for d. Let $\ell \in L(d)$ be a contour label in d and let $z \in Z(d)$ be a zone of d. Then*

$$\Psi(z) = \begin{cases} \Psi(r_\ell(z)) \cap \Psi(\ell) & \text{if } \ell \in in(z), \\ \Psi(r_\ell(z)) \cap \overline{\Psi(\ell)} & \text{if } \ell \in out(z). \end{cases}$$

Proof. Let $r_\ell(z) = (x, y)$.
If $\ell \in in(z)$, then $z = (x \cup \{\ell\}, y)$ so

$$\Psi(z) = \bigcap_{m \in x} \Psi(m) \cap \Psi(\ell) \cap \bigcap_{m \in y} \overline{\Psi(m)} = \Psi(r_\ell(z)) \cap \Psi(\ell).$$

If $\ell \in out(z)$, then $z = (x, y \cup \{\ell\})$ so

$$\Psi(z) = \bigcap_{m \in x} \Psi(m) \cap \bigcap_{m \in y} \overline{\Psi(m)} \cap \overline{\Psi(\ell)} = \Psi(r_\ell(z)) \cap \overline{\Psi(\ell)}. \qquad \square$$

COROLLARY 4.6. *Let d be an ED and let (U, Ψ) be a model for d. Let $\ell \in L(d)$ be a contour label in d and let $z \in Z(d)$ be a zone of d. Then*

$$\Psi(m_\ell(z)) = \begin{cases} \Psi(r_\ell(z)) \cap \overline{\Psi(\ell)} & \text{if } \ell \in in(z), \\ \Psi(r_\ell(z)) \cap \Psi(\ell) & \text{if } \ell \in out(z). \end{cases}$$

Proof of Theorem 4.3. We first deal with the case where ℓ is itself a nomad. If ℓ is a splitting label, then d is entirely shaded by Lemma 4.2(ii). Hence, $d \equiv_\models d - \ell$ since both diagrams assert that the universe U is empty. Conversely, if $d \equiv_\models d - \ell$ where ℓ is a nomad, then the universe $U = \varnothing$ in any model so the diagrams are entirely shaded. Hence, ℓ is trivially a splitting label.

Henceforth, suppose that ℓ is not a nomad. First suppose that ℓ is a splitting label for d. Since ℓ splits the unshaded region, any ℓ-singles are shaded. If d has any ℓ-singles then we may add the ℓ-twin of each such zone to obtain d'. We know from Theorem 3.3(iii) that adding a missing zone (as a shaded zone) gives a semantically equivalent diagram; hence, $d \equiv_\models d'$.

All of the shaded zones in d' are ℓ-twins by construction and, since ℓ is a splitting label for d, all of the unshaded zones are also ℓ-twins. Therefore, ℓ splits both the shaded and unshaded regions of d' so that, syntactically, d' is the same as $(d' - \ell) + \ell$. It therefore follows that $d' \equiv_\models d' - \ell$, by Theorem 3.3(ii).

Hence, $d \equiv_\models d' - \ell$ and it remains to show that $d - \ell = d' - \ell$. The only zones introduced into d to give d' were the twins of ℓ-singles in d. Since a zone and its ℓ-twin both map to the same zone under the remove label mapping $r_{\ell,d}$, the zones of $d - \ell$ and $d' - \ell$ are equal, $Z(d-\ell) = Z(d'-\ell)$. Since ℓ is a splitting label for d, ℓ splits $Z^\circ(d)$ so the zones introduced into d to give d' are the twins of shaded zones in d. Hence, for each zone $z \in Z(d-\ell) = Z(d'-\ell)$, the only case where $r_{\ell,d}^{-1}(z)$ is different in d and d' is when $r_{\ell,d}^{-1}(z)$ is a single shaded zone in d but a pair of shaded twins in d'. In such a case, z is shaded in both $d - \ell$ and $d' - \ell$. Therefore, the shaded zones of $d - \ell$ and $d' - \ell$ are equal, $Z^\bullet(d - \ell) = Z^\bullet(d' - \ell)$. Hence, $d - \ell = d' - \ell$ and so $d \equiv_\models d - \ell$. Hence, ℓ is a redundant contour label in d.

Conversely, suppose that ℓ is not a splitting label for d. Then ℓ does not split the unshaded region $Z^\circ(d)$. Hence, there is an unshaded zone z in $Z^\circ(d)$ such that its ℓ-twin is either shaded or missing, $m_\ell(z) \in Z^\bullet(d)$ or $m_\ell(z) \in Z^m(d)$. (This is the situation considered in Example 4.1.)

In either case, in any model of d, we have $\Psi(m_\ell(z)) = \varnothing$. Now, by Corollary 4.6,

$$\Psi(m_\ell(z)) = \begin{cases} \Psi(r_\ell(z)) \cap \overline{\Psi(\ell)} & \text{if } \ell \in in(z), \\ \Psi(r_\ell(z)) \cap \Psi(\ell) & \text{if } \ell \in out(z). \end{cases}$$

Since z is unshaded in d it follows that the corresponding zone $r_\ell(z)$ in $d - \ell$ is unshaded. Hence, there are models for $d - \ell$ for which $\Psi(r_\ell(z))$ is not empty. Choose a model (U, Ψ) for $d - \ell$ for which $\Psi(r_\ell(z)) \neq \varnothing$. Since neither the shaded zones condition nor the missing zones condition of $d - \ell$ involves $\Psi(\ell)$, we may redefine $\Psi(\ell)$ as follows, but without changing $\Psi(\ell')$ for any $\ell' \neq \ell$:

$$\Psi(\ell) = \begin{cases} \overline{\Psi(r_\ell(z))} & \text{if } \ell \in in(z), \\ \Psi(r_\ell(z)) & \text{if } \ell \in out(z). \end{cases}$$

Now, with this choice of $\Psi(\ell)$, we have

$$\Psi(m_\ell(z)) = \begin{cases} \Psi(r_\ell(z)) \cap \overline{\Psi(\ell)} = \Psi(r_\ell(z)) \cap \overline{\overline{\Psi(r_\ell(z))}} = \Psi(r_\ell(z)) & \text{if } \ell \in in(z), \\ \Psi(r_\ell(z)) \cap \Psi(\ell) = \Psi(r_\ell(z)) \cap \Psi(r_\ell(z)) = \Psi(r_\ell(z)) & \text{if } \ell \in out(z). \end{cases}$$

Since $\Psi(m_\ell(z)) = \Psi(r_\ell(z)) \neq \varnothing$, it follows that (U, Ψ) is not a model for d. Therefore, d is not semantically equivalent to $d - \ell$, and so ℓ is not a redundant contour label in d.

4.2. A normal form

In this section, we describe a normal form for EDS. The normal form will be structurally simple although, at the concrete level, it need not have a nice drawing. Further, each ED will be semantically equivalent to a unique diagram in normal form. A diagram d in normal form will have the minimal number of contour labels and the minimal number of zones amongst all of the diagrams in its semantic equivalence class $\langle d \rangle$.

In general, if d has no nomads then each shaded zone may be removed and the resulting diagram is semantically equivalent to d by Theorem 3.3(iv). The resulting abstract diagram may not have a corresponding concrete diagram that is 'nicely drawable'; for example, it may be the case that the concrete diagram is only drawable with 'concurrent contours': that is, contours that run over one another for some or all of their length.

This process is illustrated in Figure 16. The diagram d_1 has three shaded zones

$$z_1 = (\{B, C\}, \{A\}), \quad z_2 = (\{B\}, \{A, C\}), \quad z_3 = (\{A\}, \{B, C\}).$$

Deleting these in turn produces the diagrams d_2, d_3 and d_4, respectively. In d_3, the contour labelled B is indicated by a thicker line. In this diagram, parts of the contours labelled A and B are concurrent. In diagram d_4, the contours labelled A and B are entirely concurrent. The zones of d_4 are $(\{A, B, C\}, \varnothing), (\{A, B\}, \{C\})$ and $(\varnothing, \{A, B, C\})$.

DEFINITION 4.4. An ED d^N is in *normal form* if either it contains no contours or it satisfies the following three properties:
   (i) d^N had no splitting labels;
   (ii) d^N has no shaded zones outside nomads; that is, if $z \in Z^\bullet(d^N)$, then $z \in Z_{in_\mathcal{N}}(d)$;
   (iii) if d^N has nomads, then these are represented by a single shaded zone outside all of the other contours; that is, if $\mathcal{N}(d^N) \neq \varnothing$, then $(\mathcal{N}(d^N), L(d^N) - \mathcal{N}(d^N)) \in Z^\bullet(d^N)$.
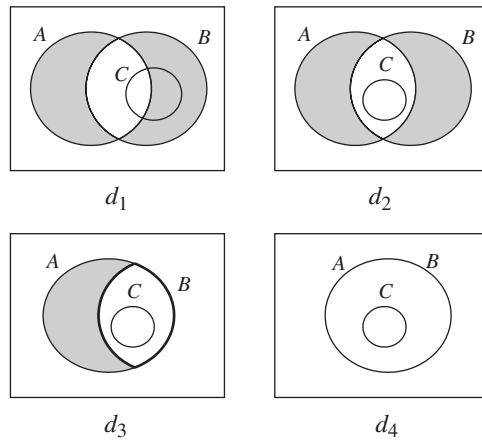
FIGURE 16. *Deleting shaded zones.*

Note that conditions (ii) and (iii) imply that if $\mathcal{N}(d^N) \neq \varnothing$, then $Z^{\bullet}(d^N) = \{(\mathcal{N}(d^N), L(d^N) - \mathcal{N}(d^N))\}$.

Let d be an ED. If d is semantically equivalent to d^N, where d^N is in normal form, then we say that d^N is the *normal form of* d.

Note that there are two diagrams in normal form that contain no contours. One is the unshaded diagram d° with $L(d^{\circ}) = \varnothing$, $Z(d^{\circ}) = \{(\varnothing, \varnothing)\}$ and $Z^{\bullet}(d^{\circ}) = \varnothing$. In fact, d° meets the three conditions in the second part of the definition. The other is the completely shaded diagram $d^{\bullet}$ with $L(d^{\bullet}) = \varnothing$ and $Z(d^{\bullet}) = \{(\varnothing, \varnothing)\} = Z^{\bullet}(d^{\bullet})$. Any unshaded Venn diagram has d° as its normal form and any completely shaded diagram has $d^{\bullet}$ as its normal form.

There is an algorithm to obtain the normal form d^N of a diagram d. The steps in obtaining d^N are described informally as follows.

  (1) If d is completely shaded, $Z(d) = Z^{\bullet}(d)$, then replace d with $d^{\bullet}$.
  (2) Otherwise, carry out the following steps:
      (i) remove any splitting labels;
     (ii) remove any shaded zones that do not lie within a nomad;
    (iii) exile any nomads.

Figure 17 illustrates the algorithm to obtain the normal form. The diagram d in Figure 17 has two redundant contours; these are the splitting labels B and G. The first step is to remove these to obtain d_1. Next the shaded zone $(\{A, C\}, \{E, F\})$ in d_1 that is not part of a nomad is removed to obtain d_2. Finally, the two nomads E and F are moved to form a single zone 'in the outside zone'. This gives the diagram d^N which is in normal form.

The following theorem follows from the semantic equivalences established in Theorems 4.3, 3.3(iv) and 3.4.

THEOREM 4.7. *Let d be an ED that is not entirely shaded and let d^N be the ED in normal form obtained by applying the following three steps to d.*

  (1) *Remove all splitting labels. Replace d with $d_1 = d - \mathscr{L}$ where $\mathscr{L} = \{\ell \in L(d) : \ell$ is a splitting label for $d\}$.*

  (2) *Remove any shaded zones that do not lie within a nomad. Replace d_1 with the diagram d_2 obtained by removing each zone in $Z^{\bullet}_{out_{\mathcal{N}}}(d_1)$.*

  (3) *Exile all nomads. If $\mathcal{N}(d_2) \neq \varnothing$, replace d_2 by d^N which is d_2 with nomads exiled. Then d is semantically equivalent to d^N.*
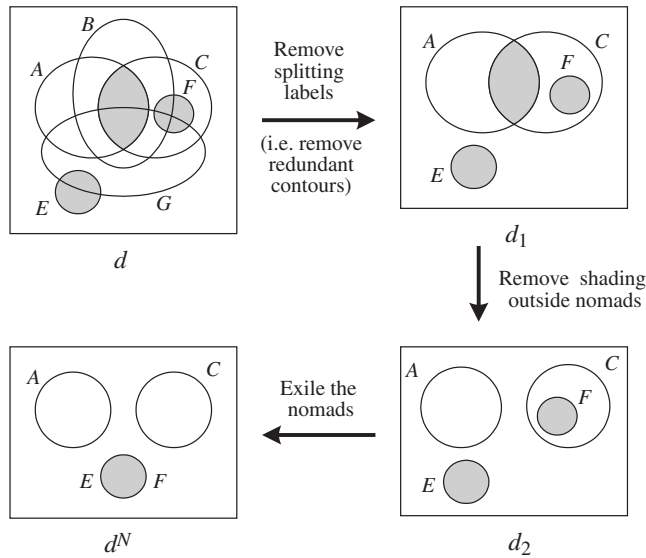
FIGURE 17. *Obtaining the normal form.*

THEOREM 4.8. *Let d_1 and d_2 be an EDs and suppose that their normal forms d_1^N and d_2^N are not the same diagram. Then d_1 and d_2 are not semantically equivalent.*

Proof. First note that if $d_1^N = d^\bullet$ but $d_2^N \neq d^\bullet$, then d_1 is entirely shaded but d_2 is not. The only model of d_1 had $U = \varnothing$ but d_2 has models with a non-empty universe, so d_1 and d_2 are not semantically equivalent.

Now assume that neither diagram is completely shaded. By Theorem 4.7, it is sufficient to prove that, if d_1 and d_2 are two different EDs, both in normal form, then d_1 and d_2 are not semantically equivalent. So suppose that d_1 and d_2 are in normal form and $d_1 \neq d_2$. Then $Z(d_1) \neq Z(d_2)$ or $Z^\bullet(d_1) \neq Z^\bullet(d_2)$.

First, consider the case where $Z^\bullet(d_1) \neq Z^\bullet(d_2)$. Since d_1 and d_2 are in normal form, this implies that the two diagrams have different sets of nomads, $\mathcal{N}(d_1) \neq \mathcal{N}(d_2)$. Without loss of generality, we may suppose that d_1 contains a nomad $n \in \mathcal{L}$ that is not in $L(d_2)$. In any model of d_1, we have $\Psi(n) = \varnothing$ but, since $n \notin L(d_2)$, there will be models for d_2 where the interpretation of n is not empty. Therefore, d_1 and d_2 are not semantically equivalent.

Now suppose that $Z^\bullet(d_1) = Z^\bullet(d_2)$ but $Z(d_1) \neq Z(d_2)$. Without loss of generality, we may suppose that d_1 contains a non-shaded zone $z \in Z^\circ(d_1)$ that is missing in d_2. In any model of d_2, we have $\Psi(z) = \varnothing$. However, there will be models for d_1 for which $\Psi(z) \neq \varnothing$. For example, we may modify any model (U, Ψ) for d_1 by replacing U with $U \cup \{u^*\}$, where $u^* \notin U$ and, for each $\ell \in in(z)$, replace $\Psi(\ell)$ with $\Psi(\ell) \cup \{u^*\}$. Thus, d_1 and d_2 are not semantically equivalent.   □

5. *Extension to Euler diagrams with projections*

5.1. *Syntax of projections*

The introduction of *projections* or *projected contours* into EDs has been considered by various authors [**30**, **31**, **52**–**54**]. Projections give 'local' information about a set relative to a 'context' and hence allow the representation of partial information about the set; projections provide the freedom to show only those intersections that are of interest. They also provide a choice
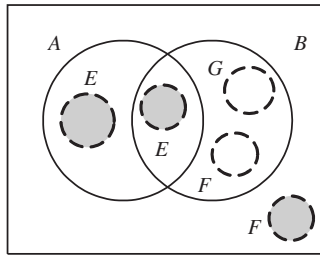
FIGURE 18. *An Euler diagram with projections.*

of ways of representing relationships between sets, just as shading and missing zones provide different ways of making the same assertions. Syntactically, we distinguish projected contours from standard or 'given' contours by using dashed lines to represent projected contours.

EXAMPLE 5.1. Figure 18 shows an EDP. The projected contours are those labelled E, F and G. Since they express 'local' information about the interaction of the contour and its context, we allow more than one projected contour with the same label in a diagram. In fact, we will (usually) draw EDP so that each projected contour is drawn within a zone of the 'underlying' ED defined by the given contours. For a discussion of different ways of drawing EDP, see § 6.2 below.

The *context* of a projected contour is the smallest region defined in terms of the non-projected contours that contains the projected contour. Thus, the context of the contour labelled E is the region inside the contour A. Similarly, the context of the contour labelled F is the region outside the contour A and the context of the projected contour G is the region inside B and outside the contour A.

A projected contour only gives information about the intersection of the set denoted by the contour label with the set denoted by its context. Thus, if we use (non-italic) X informally to denote the set represented by contour X, the diagram in Figure 18 expresses the following:
(1) $A \cap E = \varnothing$ since both the projected contours labelled E are shaded;
(2) $(F \cap G) \cap (B \cap \overline{A}) = \varnothing$ since, within the region inside B but outside A, the projected contours labelled F and G are disjoint;
(3) $F \cap (\overline{A} \cap \overline{B}) = \varnothing$ since the projected contour labelled F that is outside both A and B is shaded.

DEFINITION 5.1. An EDP d (with labels in $\mathcal{L}$) is a quadruple $\langle L_G(d), L_P(d), Z(d), Z^{\bullet}(d) \rangle$ whose components are described as follows.
   (i) The subsets $L_G(d)$ and $L_P(d)$ are disjoint subsets of $\mathcal{L}$. The contour labels in $L_G(d)$ are called *given contour labels* and those in $L_P(d)$ are called *projected contour labels*.
  (ii) A *zone* in d is a quadruple[†] (x_G, x_P, y_G, y_P) where:
  (a) $\{x_G, y_G\}$ is a partition of $L_G(d)$ (where x_G and y_G are allowed to be empty);
  (b) x_P and y_P are disjoint subsets of $L_P(d)$;
  (c) if (x_G, x_P, y_G, y_P) and (x'_G, x'_P, y'_G, y'_P) are zones such that $x_G = x'_G$ and $y_G = y'_G$, then $x_P \cup y_P = x'_P \cup y'_P$.
The set $Z(d)$ is a set of zones such that for every $\ell \in L_G(d)$ there is a zone $(x_G, x_P, y_G, y_P) \in Z(d)$ such that $\ell \in x_G$.
 (iii) The set $Z^{\bullet}(d) \subseteq Z(d)$ is the set of *shaded zones*.

[†]Sometimes it will be convenient to regard a zone as a pair (z_G, z_P) where $z_G = (x_G, y_G)$ and $z_P = (x_P, y_P)$.

The diagram in Figure 18 has the following abstract description:
(1) the given contour label set is $L_G = \{A, B\}$;
(2) the projected contour label set is $L_P = \{E, F, G\}$;
(3) the zone set is

$$Z = \{(\varnothing, \varnothing, \{A, B\}, \{F\}), (\varnothing, \{F\}, \{A, B\}, \varnothing),$$
$$(\{A\}, \varnothing, \{B\}, \{E\}), (\{A\}, \{E\}, \{B\}, \varnothing),$$
$$(\{A, B\}, \varnothing, \varnothing, \{E\}), (\{A, B\}, \{E\}, \varnothing, \varnothing),$$
$$(\{B\}, \varnothing, \{A\}, \{F, G\}), (\{B\}, \{F\}, \{A\}, \{G\}), (\{B\}, \{G\}, \{A\}, \{F\})\};$$

(4) the shaded zone set is

$$Z^{\bullet} = \{(\varnothing, \{F\}, \{A, B\}, \varnothing), (\{A\}, \{E\}, \{B\}, \varnothing), (\{A, B\}, \{E\}, \varnothing, \varnothing)\}.$$

DEFINITION 5.2. Let d be an EDP.
(i) The *underlying ED* (without projections) of d is $d_U = \langle Z(d_U), Z^{\bullet}(d_U) \rangle$, the ED with label set $L(d_U) = L_G(d)$ where:
    (a) $Z(d_U) = \{(x_G, y_G) : \exists x_P, y_P \text{ such that } (x_G, x_P, y_G, y_P) \in Z(d)\}$
    (b) $Z^{\bullet}(d_U) = \{(x_G, y_G) \in Z(d_U) : \text{every } (x_G, x_P, y_G, y_P) \in Z(d) \text{ is shaded}\}$.
    The zone set of d_U is called the *set of underlying zones* of d, denoted $Z_U(d)$. Thus $Z_U(d) = Z(d_U)$ and we use these two notations interchangeably.
    The function $v : Z(d) \to Z_U(d)$, given by $v(x_G, x_P, y_G, y_P) = (x_G, y_G)$ returns the underlying zone of each zone $(x_G, y_G) \in Z(d)$. A *missing underlying zone* of d is a zone that is missing in the underlying diagram d_U; the set of missing underlying zones is $VZ(L(d_U)) - Z(d_U)$.
(ii) Let $\ell_P \in L_P(d)$ be a projected contour label. The *context* of ℓ_P, $\kappa(\ell_P)$, is the region in the underlying diagram 'containing' ℓ_P,

$$\kappa(\ell_P) = \{(x_G, y_G) : \exists (x_G, x_P, y_G, y_P) \in Z(d) \text{ such that } \ell_P \in x_P \cup y_P\}.$$

(iii) Let $(x_G, y_G) \in Z_U(d)$ be a zone of the underlying diagram. The set of *intersectors of* (x_G, y_G), $\Lambda(x_G, y_G)$, is the set of projected contour labels that intersect with the zone,

$$\Lambda(x_G, y_G) = \{\ell_P \in L_P(d) : (x_G, y_G) \in \kappa(\ell_P)\}.$$

The set of *zones in* (x_G, y_G) is

$$Z(x_G, y_G) = \{z \in Z(d) : z = (x_G, x_P, y_G, y_P) \text{ for some } x_P, y_P\}.$$

The set of *shaded zones in* (x_G, y_G) is

$$Z^{\bullet}(x_G, y_G) = Z(x_G, y_G) \cap Z^{\bullet}(d).$$

EXAMPLE 5.2. Consider the diagram in Figure 18. The underlying diagram is the Venn diagram with contour labels A and B; that is, d_U has zones $(\varnothing, \{A, B\}), (\{A\}, \{B\}), (\{B\}, \{A\})$ and $(\{A, B\}, \varnothing)$.
The contexts of the projected contour labels are

$$\kappa(E) = \{(x_G, y_G) : (x_G, x_P, y_G, y_P) \in Z(d) \text{ and } E \in x_P\}$$
$$= \{(\{A\}, \{B\}), (\{A, B\}, \varnothing)\};$$
$$\kappa(F) = \{(x_G, y_G) : (x_G, x_P, y_G, y_P) \in Z(d) \text{ and } F \in x_P\}$$
$$= \{(\varnothing, \{A, B\}), (\{B\}, \{A\})\};$$
$$\kappa(G) = \{(x_G, y_G) : (x_G, x_P, y_G, y_P) \in Z(d) \text{ and } G \in x_P\}$$
$$= \{(\{B\}, \{A\})\}.$$

Therefore, the sets of intersectors of the underlying zones are

$$\Lambda(\varnothing, \{A, B\}) = \{\ell_P \in \{E, F, G\} : (\varnothing, \{A, B\}) \in \kappa(\ell_P)\} = \{F\},$$
$$\Lambda(\{A\}, \{B\}) = \{\ell_P \in \{E, F, G\} : (\{A\}, \{B\}) \in \kappa(\ell_P)\} = \{E\},$$
$$\Lambda(\{B\}, \{A\}) = \{\ell_P \in \{E, F, G\} : (\{B\}, \{A\}) \in \kappa(\ell_P)\} = \{F, G\},$$
$$\Lambda(\{A, B\}, \varnothing) = \{\ell_P \in \{E, F, G\} : (\{A, B\}, \varnothing) \in \kappa(\ell_P)\} = \{E\}.$$

LEMMA 5.1. *Let d be an EDP and let (x_G, y_G) be an underlying zone of d. Then $\Lambda(x_G, y_G) = \{x_P \cup y_P :$ there exists $(x_G, x_P, y_G, y_P) \in Z(d)\}$.*

Let d be an EDP. For each zone (a, b) of the underlying diagram d_U, the intersectors of the zone form a 'projected diagram embedded within the zone'. For example, in the diagram of Figure 18, the projected diagram embedded in the zone $(\{B\}, \{A\})$ comprises two separated contours labelled F and G. The following definition makes this precise.

DEFINITION 5.3. *Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP and let $(x_G, y_G) \in Z_U(d)$ be a zone in the underling diagram d_U. The projected diagram of d embedded in (x_G, y_G), $d_P(x_G, y_G)$, is the ED (without projections) that has:*
(1) *label set $L(d_P(x_G, y_G)) = \Lambda(x_G, y_G)$,*
(2) *zone set $Z(d_P(a, b)) = \{(x_P, y_P) : (x_G, x_P, y_G, y_P) \in Z(x_G, y_G)\}$*
(3) *shaded zone set $Z^\bullet(d_P(x_G, y_G)) = \{(x_P, y_P) : (x_G, x_P, y_G, y_P) \in Z^\bullet(x_G, y_G)\}$.*
We will also refer to $d_P(x_G, y_G)$ as an embedded projected diagram.

LEMMA 5.2. *Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP and let $(x_G, y_G) \in Z_U(d)$ be a zone in the underlying diagram d_U. The mapping $w : Z(x_G, y_G) \to Z(d_P(x_G, y_G))$ given by*

$$w(x_G, x_P, y_G, y_P) = (x_P, y_P)$$

is a bijection and, hence, defines a correspondence between the zones in $Z(x_G, y_G)$ and the zones in the projected diagram of d embedded in (x_G, y_G).

EXAMPLE 5.3. Consider the diagram d in Figure 19. First, we evaluate $d_P(\{A\}, \{B\})$, the projected diagram embedded in the zone $(\{A\}, \{B\})$ of the underlying diagram.

The intersectors of the zone are $\Lambda(\{A\}, \{B\}) = \{E, F\}$. The sets of zones and shaded zones in $(\{A\}, \{B\})$ are

$$Z(\{A\}, \{B\}) = \{(\{A\}, \varnothing, \{B\}, \{E, F\}), (\{A\}, \{E\}, \{B\}, \{F\}),$$
$$(\{A\}, \{E, F\}, \{B\}, \varnothing), (\{A\}, \{F\}, \{B\}, \{E\})\},$$
$$Z^\bullet(\{A\}, \{B\}) = Z(\{A\}, \{B\}) \cap Z^\bullet(d) = \{(\{A\}, \{E\}, \{B\}, \{F\})\}.$$

Therefore, the sets of zones and shaded zones of $d_P(\{A\}, \{B\})$ are

$$Z(d_P(\{A\}, \{B\})) = \{(x_P, y_P) : (\{A\}, x_P, \{B\}, y_P) \in Z(x_G, y_G)\}$$
$$= \{(\varnothing, \{E, F\}), (\{E\}, \{F\}), (\{E, F\}, \varnothing), (\{F\}, \{E\})\},$$
$$Z^\bullet(d_P(\{A\}, \{B\})) = \{(x_P, y_P) : (\{A\}, x_P, \{B\}, y_P) \in Z^\bullet(\{A\}, \{B\})\}$$
$$= \{(\{E\}, \{F\})\}.$$

The embedded projected diagrams $d_P(\{A\}, \{B\})$ and $d_P(\{B\}, \{A\})$ are shown in Figure 19. The remaining two embedded projected diagrams, $d_P(\varnothing, \{A, B\})$ and $d_P(\{A, B\}, \varnothing)$, are the diagram d°, the diagram with empty label set.
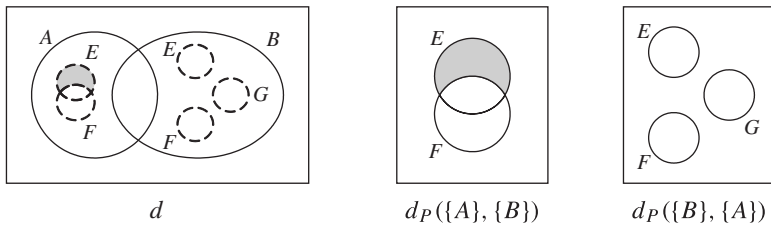
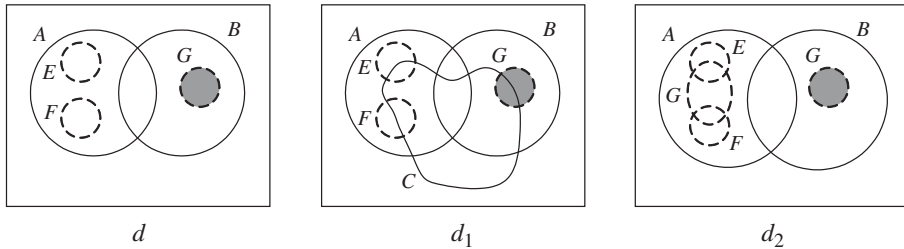FIGURE 19. *The projected diagrams within underlying zones.*

FIGURE 20. *Adding contours to an Euler diagram with projections.*

We will need to be able to add and remove syntactic elements to EDP in a similar way to adding and removing syntactic elements to EDs. However, we will need to distinguish between given and projected syntactic elements when defining the operations to add or delete syntactic elements. We illustrate this with a consideration of how we add contours to EDP.

Let d be an EDP and let ℓ_g be a contour label not in $L(d)$. Adding ℓ_g as a given contour is similar to adding a contour to an ED: see Definition 3.4. In Figure 20, the diagram d_1 is obtained from d by adding C as a given contour, $d_1 = d + C$. As for EDs, the new contour C splits each existing zone into two new zones in $d + C$.

DEFINITION 5.4. Let d be an EDP and let ℓ_g be a contour label not in $L(d)$. The diagram d with ℓ_g added as a *given contour*, $d + \ell_g$, has the following components:
  (i) $L_G(d + \ell_g) = L(d) \cup \{\ell_g\}$;
  (ii) $L_P(d + \ell_g) = L_P(d)$;
  (iii) for every zone $(x_G, x_P, y_G, y_P) \in L(d)$ there are zones $(x_G \cup \{\ell_g\}, x_P, y_G, y_P)$ and $(x_G, x_P, y_G \cup \{\ell_g\}, y_P)$ in $Z(d + \ell_g)$ so that

$$Z(d + \ell_g) = \{(x_G \cup \{\ell_g\}, x_P, y_G, y_P) : (x_G, x_P, y_G, y_P) \in Z(d)\}$$
$$\cup \{(x_G, x_P, y_G \cup \{\ell_g\}, y_P) : (x_G, x_P, y_G, y_P) \in Z(d)\};$$

  (iv) we have $Z^\bullet(d + \ell_g) = \{(x_G \cup \{\ell_g\}, x_P, y_G, y_P) : (x_G, x_P, y_G, y_P) \in Z^\bullet(d)\}$
$$\cup \{(x_G, x_P, y_G \cup \{\ell_g\}, y_P) : (x_G, x_P, y_G, y_P) \in Z^\bullet(d)\}.$$

To add a contour label as a projected contour label, we need to specify the underlying zone into which it is to be inserted. In Figure 20, the diagram d_2 is obtained from d by adding the projected contour G into the zone $(\{A\}, \{B\})$ of the underlying diagram of d. We will denote d_2 by $d + (G \to (\{A\}, \{B\}))$. Note that the contour G that is added as a projected contour already exists as a projected contour in the diagram but not in the underlying zone $(\{A\}, \{B\})$. The zones outside $(\{A\}, \{B\})$ are unchanged by the addition of G and all the zones inside $(\{A\}, \{B\})$ are split by G. We now formalise this notion.

DEFINITION 5.5. Let d be an EDP and let $(x_G, y_G) \in Z_U(d)$ be a zone of the underlying diagram. Let $\ell_p \in \mathcal{L}$ be a contour label that is not a given contour of d and is not an intersector of (x_G, y_G), $\ell_p \notin L_G(d) \cup \Lambda(x_G, y_G)$. The ED d with ℓ_p added into (x_G, y_G), denoted $d + (\ell_p \to (x_G, y_G))$, is $d' = \langle L_G(d'), L_P(d'), Z(d'), Z^{\bullet}(d') \rangle$ where:

(i) $L_G(d') = L_G(d)$;

(ii) $L_P(d') = L_P(d) \cup \{\ell_p\}$;

(iii) $Z(d') = \{z : z \in Z(d), z \notin Z(x_G, y_G)\}$
$\qquad \cup \{(x_G, x_P \cup \{\ell_p\}, y_G, y_P) : (x_G, x_P, y_G, y_P) \in Z(x_G, y_G)\}$
$\qquad \cup \{(x_G, x_P, y_G, y_P \cup \{\ell_p\}) : (x_G, x_P, y_G, y_P) \in Z(x_G, y_G)\}$; and

(iv) $Z^{\bullet}(d') = \{z : z \in Z^{\bullet}(d), z \notin Z(x_G, y_G)\}$
$\qquad \cup \{(x_G, x_P \cup \{\ell_p\}, y_G, y_P) : (x_G, x_P, y_G, y_P) \in Z^{\bullet}(x_G, y_G)\}$
$\qquad \cup \{(x_G, x_P, y_G, y_P \cup \{\ell_p\}) : (x_G, x_P, y_G, y_P) \in Z^{\bullet}(x_G, y_G)\}$.

5.2. Semantics of projections

Projected contours only provide information 'relative to their context'. For example, in Figure 19, the diagram d does *not* express that the sets (represented by) F and G are disjoint but only their intersections with $\overline{A} \cap B$ are disjoint. Our definition of zones allows us to extend Definition 2.4 of interpretations in the obvious way. Modifying the definition of an abstract zone to be any quadruple (x_G, y_P, y_G, y_P) of pairwise disjoint finite subsets of $\mathcal{L}$, the only modification to Definition 2.4 is in the interpretation of zones

$$\Psi(x_G, y_P, y_G, y_P) = \bigcap_{\ell \in x_G \cup x_P} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P} \overline{\Psi(\ell)}.$$

To define the semantics predicate for an EDP d, we need the concept of a missing projected zone in an underlying zone, which we now define.

DEFINITION 5.6. Let d be an EDP. Let $(x_G, y_G) \in Z_U(d)$ be an underlying zone of d with intersectors $\Lambda(x_G, y_G)$. The *projected Venn zone set in* (x_G, y_G) is

$$VZ_P(x_G, y_G) = \{(x_G, x_P, y_G, y_P) : x_P \cup y_P = \Lambda(x_G, y_G)\}.$$

The set of *missing projected zones in* (x_G, y_G) is

$$Z_P^m(x_G, y_G) = VZ_P(x_G, y_G) - Z(x_G, y_G).$$

LEMMA 5.3. *Let d be an EDP and let $(x_G, y_G) \in Z_U(d)$ be an underlying zone of d with intersectors $\Lambda(x_G, y_G)$. The mapping $w : Z(x_G, y_G) \to Z(d_P(x_G, y_G))$, defined by $w(x_G, x_P, y_G, y_P) = (x_P, y_P)$ extends to a bijection $Z_P^m(x_G, y_G) \to Z^m(d_P(x_G, y_G))$. Hence, there is a bijective correspondence between the missing projected zones in (x_G, y_G) in d and the missing zones of the embedded projected diagram $d_P(x_G, y_G)$.*

EXAMPLE 5.4. Consider the underlying zone $(\{B\}, \{A\})$ in the diagram d in Figure 19. The intersectors of the zone are
$$\Lambda(\{B\}, \{A\}) = \{E, F, G\}$$
so

$$VZ_P(\{B\}, \{A\}) = \{(\{B\}, x_P, \{A\}, y_P) : x_P \cup y_P = \{E, F, G\}\}$$
$$= \{(\{B\}, \{E, F, G\}, \{A\}, \varnothing), (\{B\}, \{E, F\}, \{A\}, \{G\}),$$
$$(\{B\}, \{E, G\}, \{A\}, \{F\}), (\{B\}, \{F, G\}, \{A\}, \{E\}),$$
$$(\{B\}, \{E\}, \{A\}, \{F, G\}), (\{B\}, \{F\}, \{A\}, \{E, G\}),$$
$$(\{B\}, \{G\}, \{A\}, \{E, F\}), (\{B\}, \varnothing, \{A\}, \{E, F, G\})\}.$$
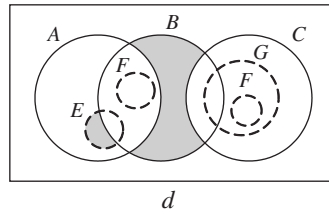
FIGURE 21. *The semantics of projections.*

Hence, the missing projected zones in $(\{B\}, \{A\})$ are

$$Z_P^m(\{B\}, \{A\}) = \{(\{B\}, \{E, F, G\}, \{A\}, \varnothing), (\{B\}, \{E, F\}, \{A\}, \{G\}),$$
$$(\{B\}, \{E, G\}, \{A\}, \{F\}), (\{B\}, \{F, G\}, \{A\}, \{E\})\}.$$

By contrast, the underlying zone $(\{A\}, \{B\})$ has intersectors $\Lambda(\{A\}, \{B\}) = \{E, F\}$ and projected Venn zone set

$$VZ_P(\{A\}, \{B\}) = \{(\{A\}, x_P, \{B\}, y_P) : x_P \cup y_P = \{E, F\}\}$$
$$= \{(\{A\}, \{E, F\}, \{B\}, \varnothing), (\{A\}, \{E\}, \{B\}, \{F\}),$$
$$(\{A\}, \{F\}, \{B\}, \{E\}), (\{A\}, \varnothing, \{B\}, \{E, F\})\}.$$

Since $Z(\{A\}, \{B\}) = VZ_P(\{A\}, \{B\})$, there are no missing projected zones in $(\{A\}, \{B\})$,

$$Z_P^m(\{A\}, \{B\}) = \varnothing.$$

Definitions 2.5 and 2.7 extend to EDP in the obvious way.

DEFINITION 5.7. Let d be an EDP and let $I = (U, \Psi)$ be an interpretation. We define the *semantics predicate* of d, denoted $P_d(I)$, to be the conjunction of the following conditions.
   (i) *Shaded zones condition.* Each shaded zone represents the empty set: $\bigwedge_{z \in Z^\bullet(d)} \Psi(z) = \varnothing$.
  (ii) *Missing zones condition.* Each missing zone represents the empty set.
       For convenience, we split this into the conjunction of two sub-conditions.
     (1) *Missing underlying zones condition.* Each missing underlying zone represents the empty set:
$$\bigwedge_{z \in Z^m(d_U)} \Psi(z) = \varnothing.$$

     (2) *Missing projected zones condition.* Each missing projected zone represents the empty set:
$$\bigwedge_{(x_G, y_G) \in Z_U(d)} \bigwedge_{z \in Z_P^m(x_G, y_G)} \Psi(z) = \varnothing.$$

We say that I is a *model* for d, denoted $I \models d$, if and only if $P_d(I)$ is true. Also, we say d is *satisfiable* if and only if d has a model.

DEFINITION 5.8. Let d_1 and d_2 be EDP. Then d_1 *semantically entails* d_2 (and d_2 is a *semantic consequence* of d_1), denoted $d_1 \vDash d_2$, if every interpretation (U, Ψ) that is a model for d_1 is also a model for d_2.
   If $d_1 \vDash d_2$ and $d_2 \vDash d_1$ then we say that d_1 and d_2 are *semantically equivalent*, denoted $d_1 \equiv_\vDash d_2$.

EXAMPLE 5.5. Consider the EDP d given in Figure 21.

There are two shaded zones, $(\{A\}, \{E\}, \{B, C\}, \varnothing)$ and $(\{B\}, \varnothing, \{A, C\}, \varnothing)$, so the shaded zones condition for d is

$$(\Psi(A) \cap \Psi(E) \cap \overline{\Psi(B)} \cap \overline{\Psi(C)} = \varnothing) \wedge (\overline{\Psi(A)} \cap \Psi(B) \cap \overline{\Psi(C)} = \varnothing).$$

Note that the second of these shaded zones is a shaded zone of the underlying diagram d_U. In this case, since the interpretation of the zone does not involve the interpretation of the projected contour labels, the condition

$$\overline{\Psi(A)} \cap \Psi(B) \cap \overline{\Psi(C)} = \varnothing$$

is just the shaded zones condition for the underlying diagram.

There are two missing underlying zones, $(\{A, B, C\}, \varnothing)$ and $(\{A, C\}, \{B\})$. Hence, the missing underlying zones condition for d is

$$(\Psi(A) \cap \Psi(B) \cap \Psi(C) = \varnothing) \wedge (\Psi(A) \cap \overline{\Psi(B)} \cap \Psi(C) = \varnothing).$$

There are missing projected zones in two of the six underlying zones. In the underlying zone $(\{A, B\}, \{C\})$, the projected zone $(\{A, B\}, \{E, F\}, \{C\}, \varnothing)$ is missing and in the underlying zone $(\{C\}, \{A, B\})$ the projected zone $(\{C\}, \{F\}, \{A, B\}, \{G\})$ is missing. Hence, the missing projected zones condition for d is

$$\Psi(A) \cap \Psi(B) \cap \overline{\Psi(C)} \cap \Psi(E) \cap \Psi(F) = \varnothing$$

$$\wedge\ \overline{\Psi(A)} \cap \overline{\Psi(B)} \cap \Psi(C) \cap \Psi(F) \cap \overline{\Psi(G)} = \varnothing.$$

In Table 5.2, we give the interpretations of all of the zones in the diagram d in Figure 21. The zones of d are listed in column 3, grouped by their underlying zones. The interpretations of the underlying zones and the zones in d are given in the second and fourth columns respectively. To aid readability of the interpretation columns of the table, we have again replaced $\Psi(X)$ with a (non-italic) X for each contour label.

TABLE 5.2. *Interpreting the zones of d in Figure* 21.

Underlying zone	Interpretation	Zone	Interpretation
$(\varnothing, \{A, B, C\})$	$\overline{A} \cap \overline{B} \cap \overline{C}$	$(\varnothing, \varnothing, \{A, B, C\}, \varnothing)$	$\overline{A} \cap \overline{B} \cap \overline{C}$
$(\{A\}, \{B, C\})$	$A \cap \overline{B} \cap \overline{C}$	$(\{A\}, \varnothing, \{B, C\}, \{E\})$	$A \cap \overline{B} \cap \overline{C} \cap \overline{E}$
		$(\{A\}, \{E\}, \{B, C\}, \varnothing)$	$A \cap \overline{B} \cap \overline{C} \cap E$
$(\{A, B\}, \{C\})$	$A \cap B \cap \overline{C}$	$(\{A, B\}, \varnothing, \{C\}, \{E, F\})$	$A \cap B \cap \overline{C} \cap \overline{E} \cap \overline{F}$
		$(\{A, B\}, \{E\}, \{C\}, \{F\})$	$A \cap B \cap \overline{C} \cap E \cap \overline{F}$
		$(\{A, B\}, \{F\}, \{C\}, \{E\})$	$A \cap B \cap \overline{C} \cap \overline{E} \cap F$
$(\{B\}, \{A, C\})$	$\overline{A} \cap B \cap \overline{C}$	$(\{B\}, \varnothing, \{A, C\}, \varnothing)$	$\overline{A} \cap B \cap \overline{C}$
$(\{B, C\}, \{A\})$	$\overline{A} \cap B \cap C$	$(\{B, C\}, \varnothing, \{A\}, \{G\})$	$\overline{A} \cap B \cap C \cap \overline{G}$
		$(\{B, C\}, \{G\}, \{A\}, \varnothing)$	$\overline{A} \cap B \cap C \cap G$
$(\{C\}, \{A, B\})$	$\overline{A} \cap \overline{B} \cap C$	$(\{C\}, \varnothing, \{A, B\}, \{F, G\})$	$\overline{A} \cap \overline{B} \cap C \cap \overline{F} \cap \overline{G}$
		$(\{C\}, \{G\}, \{A, B\}, \{F\})$	$\overline{A} \cap \overline{B} \cap C \cap \overline{F} \cap G$
		$(\{C\}, \{F, G\}, \{A, B\}, \varnothing)$	$\overline{A} \cap \overline{B} \cap C \cap F \cap G$

Table 5.2 highlights that the interpretation of each zone z in d can be expressed as

$$\Psi(z) = \Psi(v(z)) \cap \Psi(w(z)),$$

where $v(z)$ is the underlying zone of z in the underlying diagram d_U and $w(z)$ is the corresponding zone in the embedded projected diagram $d_P(v(z))$. For example, for the zone $z = (\{C\}, \{G\}, \{A, B\}, \{F\})$ we have $v(z) = (\{C\}, \{A, B\})$, $w(z) = (\{G\}, \{F\})$ and

$$\Psi(z) = (\overline{A} \cap \overline{B} \cap C) \cap (\overline{F} \cap G) = \Psi(v(z)) \cap \Psi(w(z)).$$

LEMMA 5.4. *Let d be an EDP and let $z \in Z(d)$ be a zone in d. For any interpretation (U, Ψ),*

$$\Psi(z) = \Psi(v(z)) \cap \Psi(w(z)),$$

where $v(z)$ is the underlying zone of z, $v(z) \in Z(d_U)$ and $w(z)$ is the corresponding zone in the embedded projected diagram $w(z) \in Z(d_P(v(z)))$.

Proof. Let $z = (x_G, x_P, y_G, y_P)$ be a zone of the d that lies in the underlying zone $v(z) = (x_G, y_G)$. Then $w(z) = (x_P, y_P)$. Now

$$\begin{aligned}
\Psi(z) &= \bigcap_{\ell \in x_G \cup x_P} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P} \overline{\Psi(\ell)} \\
&= \bigcap_{\ell \in x_G} \Psi(\ell) \cap \bigcap_{\ell \in y_G} \overline{\Psi(\ell)} \cap \bigcap_{\ell \in x_P} \Psi(\ell) \cap \bigcap_{\ell \in y_P} \overline{\Psi(\ell)} \\
&= \Psi(x_G, y_G) \cap \Psi(x_P, y_P) = \Psi(v(z)) \cap \Psi(w(z)). \qquad \square
\end{aligned}$$

The next lemma extends Theorem 3.3(ii) by showing that adding a (given or projected) contour to an EDP results in a semantically equivalent diagram.

LEMMA 5.5. *Let d be an EDP.*
(i) *Let $\ell_g \in \mathcal{L} - L(d)$. Then $d \equiv_\models d + \ell_g$.*
(ii) *Let z_U be an underlying zone of d and let $\ell_p \in \mathcal{L} - (L_G(d) \cup \Lambda(z_U))$.*
Then $d \equiv_\models d + (\ell_p \to z_U)$.

Proof.   (1) This is essentially the same as Theorem 3.3(ii).
(2) Let (U, Ψ) be an interpretation that is a model for d and let $d' = d + (\ell_p \to z_U)$ where $z_U = (x_G, y_G)$ is an underlying zone of d.

Suppose that z is a shaded zone of d' in z_U. Then $z = (x_G, x_P \cup \{\ell_p\}, y_G, y_P)$ or $z = (x_G, x_P, y_G, y_P \cup \{\ell_p\})$ for some shaded zone (x_G, x_P, y_G, y_P) of d in z_U. In the first case, $\Psi(z) = \Psi(x_G, x_P, y_G, y_P) \cap \Psi(\ell_p)$ and in the second case, $\Psi(z) = \Psi(x_G, x_P, y_G, y_P) \cap \overline{\Psi(\ell_p)}$. Hence, by the shaded zones condition for d, $\Psi(z) = \varnothing$. Since the zones outside z_U remain unchanged, (U, Ψ) satisfies the shaded zones condition for d'.

Since the underlying diagrams for d and d' are equal, the missing underlying zones condition for d' follows from that for d.

A missing projected zone $z = (x_G, x_P \cup \{\ell_p\}, y_G, y_P)$ or $z = (x_G, x_P, y_G, y_P \cup \{\ell_p\})$ for some missing projected zone (x_G, x_P, y_G, y_P) of d in z_U. Then, as above, $\Psi(z) = \varnothing$ by the missing projected zones condition for d. Hence, (U, Ψ) satisfies the missing projected zones condition for d'.

Therefore, (U, Ψ) is a model for d'.

Conversely, let (U, Ψ) be an interpretation that is a model for $d' = d + (\ell_p \to z_U)$.

Suppose that $z = (x_G, x_P, y_G, y_P)$ is a shaded zone of d in z_U. Then the zones $(x_G, x_P \cup \{\ell_p\}, y_G, y_P)$ and $(x_G, x_P, y_G, y_P \cup \{\ell_p\})$ are shaded zones of d' in z_U. We have

$$
\begin{aligned}
\Psi(z) &= \bigcap_{\ell \in x_G \cup x_G} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P} \overline{\Psi(\ell)} \\
&= \left(\bigcap_{\ell \in x_G \cup x_G} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P} \overline{\Psi(\ell)} \right) \cap \left(\Psi(\ell_p) \cup \overline{\Psi(\ell_p)} \right) \\
&= \left(\bigcap_{\ell \in x_G \cup x_P \cup \{\ell_p\}} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P} \overline{\Psi(\ell)} \right) \cup \left(\bigcap_{\ell \in x_G \cup x_P} \Psi(\ell) \cap \bigcap_{\ell \in y_G \cup y_P \cup \{\ell_p\}} \overline{\Psi(\ell)} \right) \\
&= \Psi(x_G, x_P \cup \{\ell_p\}, y_G, y_P) \cup \Psi(x_G, x_P, y_G, y_P \cup \{\ell_p\}).
\end{aligned}
$$

Hence, $\Psi(z) = \varnothing$ by the shaded zones condition for d'. Therefore, (U, Ψ) satisfies the shaded zones condition for d.

Again, the missing underlying zones condition for d follows trivially from that for d'.

Let $z = (x_G, x_P, y_G, y_P)$ be a missing projected zone in z_U in d. Then the projected zones $(x_G, x_P \cup \{\ell_p\}, y_G, y_P)$ and $(x_G, x_P, y_G, y_P \cup \{\ell_p\})$ are missing projected zones of d' in z_U. Hence, as above, $\Psi(z) = \Psi(x_G, x_P \cup \{\ell_p\}, y_G, y_P) \cup \Psi(x_G, x_P, y_G, y_P \cup \{\ell_p\}) = \varnothing$ by the missing projected zones condition for d'. Therefore, (U, Ψ) satisfies the missing projected zones condition for d.

Hence, (U, Ψ) is a model for d. $\qquad\square$

To link models of an EDP d with models of its embedded projected diagrams, we will need to modify the interpretations of the projected diagrams. To see why this is, consider the shaded zone $(\{A\}, \{E\}, \{B, C\}, \varnothing)$ in the diagram d in Figure 21. Suppose that (U, Ψ) is an interpretation that is a model for d. By the shaded zones condition, $\Psi(\{A\}, \{E\}, \{B, C\}, \varnothing) = \varnothing$ which gives

$$
\Psi(A) \cap \Psi(E) \cap \overline{\Psi(B)} \cap \overline{\Psi(C)} = \varnothing.
$$

The projected diagram embedded in the underlying zone $(\{A\}, \{B, C\})$ contains a single contour labelled E, the zone inside E being shaded. From the equation above we cannot infer the shaded zones condition for the embedded projected diagram, $\Psi(E) = \varnothing$.

DEFINITION 5.9. Let (U, Ψ) be an interpretation. Let d be an EDP, let (x_G, y_G) be an underlying zone of d and let $d_P(x_G, y_G)$ be the embedded projected diagram in (x_G, y_G).

The *modified interpretation of* $d_P(x_G, y_G)$ is the interpretation of zones and regions in $d_P(x_G, y_G)$,

$$
\Psi^{(x_G, y_G)} : Z(d_P(x_G, y_G)) \cup R(d_P(x_G, y_G)) \to \mathbb{P}(U),
$$

defined as follows.

Each zone in $d_P(x_G, y_G)$ is (x_P, y_P) for some zone $(x_G, x_P, y_G, y_P) \in Z(x_G, y_G)$. We define

$$
\Psi^{(x_G, y_G)}(x_P, y_P) = \Psi(x_G, y_G) \cap \Psi(x_P, y_P).
$$

This is extended to regions in $d_P(x_G, y_G)$ in the natural way. If $r \in R(d_P(x_G, y_G))$, then we define

$$
\Psi^{(x_G, y_G)}(r) = \bigcup_{(x_P, y_P) \in r} \Psi^{(x_G, y_G)}(x_P, y_P).
$$

The following theorem provides the link between models of an EDP d and models of its embedded projected diagrams.

THEOREM 5.6. *Let (U, Ψ) be an interpretation. Let d be an EDP with underlying diagram d_U and let $\{d_P(z_U) : z_U \in Z(d_U)\}$ be the set of embedded projected diagrams. Suppose that (U, Ψ) is a model for d. Then:*

(i) *(U, Ψ) is a model for d_U;*

(ii) *for each underlying zone $z_U \in Z(d_U)$, the modified interpretation (U, Ψ^{z_U}) is a model for $d_P(z_U)$, the projected diagram embedded in z_U.*

Conversely, if (U, Ψ) is an interpretation satisfying parts (i) *and* (ii), *then (U, Ψ) is a model for d.*

Proof. Let (U, Ψ) be an interpretation that is a model for d.

(i) The pair (U, Ψ) is an interpretation of d_U.

If $z_U \in Z^m(d_U)$ is a missing zone in the underlying diagram, then z_U is missing in d. Hence, the missing zones condition for d_U follows from the missing zones condition for d.

If $z_U = (x_G, y_G) \in Z^\bullet(d_U)$, then every zone (x_G, x_P, y_G, y_P) in $Z(z_U)$ is shaded in d. Now

$$\Psi(z_U) = \bigcup_{z \in VZ(\Lambda(z_U))} \Psi(z)$$

and every zone in $VZ(\Lambda(z_U))$ is either shaded or missing. It follows that $\Psi(z_U) = \varnothing$ by the shaded and missing zones conditions for d. Hence, the shaded zones condition holds for d_U.

Therefore, (U, Ψ) is a model for d_U as required.

(ii) Let $z_U = (x_G, y_G) \in Z(d_U)$ be a zone of the underlying diagram d_U.

Recall from Lemma 5.2 that the map $w : Z(z_U) \to Z(d_P(z_U))$, given by $w(x_G, x_P, y_G, y_P) = (x_P, y_P)$, defines a natural bijective correspondence between the zones in the projected diagram embedded in z_U and the zones in $Z(z_U)$.

Let $(U, \Psi^{(z_U)})$ be the modified interpretation.

Let $z \in Z^m(d_P(z_U))$ be a missing zone in $d_P(z_U)$. Then the corresponding zone $z' = w^{-1}(z) = (x_G, x_P, y_G, y_P)$ is a missing projected zone in (z_U), $z' \in Z_P^m(z_U)$. By the missing projected zones condition for d, $\Psi(z') = \varnothing$. Since $\Psi(z') = \Psi^{z_U}(z)$ by Definition 5.9, the missing zones condition is satisfied.

Similarly, let $z \in Z^\bullet(d_P(z_U))$ be a shaded zone in $d_P(z_U)$. Then the corresponding zone $z' = w^{-1}(z) = (x_G, x_P, y_G, y_P)$ is a shaded zone in (z_U), $z' \in Z^\bullet(z_U)$. By the shaded zones condition for d, $\Psi_d(z') = \varnothing$. But $\Psi_d(z') = \Psi^{z_U}(z)$ by Definition 5.9 so the shaded zones condition is satisfied by the modified interpretation.

Hence, the modified interpretation (U, Ψ^{z_U}) is a model for $d_P(z_U)$, the projected diagram embedded in z_U.

Conversely let (U, Ψ) be an interpretation satisfying parts (i) and (ii).

Let z be a shaded zone, $z \in Z^\bullet(d)$. If z is a zone of the underlying diagram then $\Psi(z) = \varnothing$ by the shaded zones condition for d_U. If z is not a zone of d_U, then $z = (x_G, x_P, y_G, y_P)$ where $v(z) = (x_G, y_G)$ and $\Lambda(z) = x_P \cup y_P \neq \varnothing$. In this case $\Psi(z) = \Psi^{(x_G, y_G)}(x, y)$ by Definition 5.9. Now (x_P, y_P) is a shaded zone in $d_P(x_G, y_G)$ so

$$\Psi(z) = \Psi^{(x_G, y_G)}(x_P, y_P) = \Psi(x_G, y_G) \cap \Psi(x_P, y_P) = \varnothing$$

by the shaded zones condition for $d_P(x_G, y_G)$. Hence, the shaded zones condition holds for d.

Let z be a missing underlying zone, $z \in Z^m(d_U)$. Then $\Psi(z) = \varnothing$ by the missing zones condition for d_U. Hence, the missing underlying zones condition for d is satisfied.

Finally, let z be a missing projected zone in the underlying zone (x_G, y_G), $z \in Z_P^m(x_G, y_G)$. Then $z = (x_G, x_P, y_G, y_P)$ where $\Lambda(z) = x_P \cup y_P$ and, as above, $\Psi(z) = \Psi^{(x_G, y_G)}(x_P, y_P)$. Therefore, $\Psi(z) = \varnothing$ by the missing zones condition for $d_P(x_G, y_G)$. Hence, the missing projected zones condition for d is satisfied.

Therefore, the interpretation (U, Ψ) is a model for d. $\qquad\square$
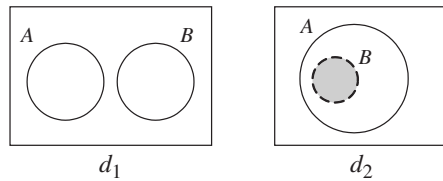
FIGURE 22. *Equivalent diagrams with inequivalent underlying diagrams.*

Let d and d' be two semantically equivalent EDP. It is not necessarily the case that their underlying diagrams d_U and d'_U are semantically equivalent. This is illustrated in Figure 22. The semantics predicate for each of the diagrams d_1 and d_2 is $\Psi(A) \cap \overline{\Psi(B)} = \varnothing$ but their underlying diagrams are not semantically equivalent.

However, for semantically equivalent diagrams where there is no contour that is given in one diagram and projected in the other, it does follow that their underlying diagrams are equivalent. Before we prove this, we need the following technical lemma.

LEMMA 5.7. *Let d be an EDP and let $\ell \in \mathcal{L} - L(d)$ be a label not in d. Then the underlying diagram of $d + \ell$ is the diagram obtained from adding ℓ to the underlying diagram of d,*

$$(d + \ell)_U = d_U + \ell.$$

Proof. Adding ℓ to d splits every zone. Hence, after removing all of the projections, ℓ splits every underlying zone. This is exactly the same as the addition of ℓ to the underlying diagram d_U. □

THEOREM 5.8. *Let d and d' be two semantically equivalent EDP with the property that there is no contour that is given in one diagram and projected in the other, $L_G(d) \cap L_P(d') = \varnothing$ and $L_P(d) \cap L_G(d') = \varnothing$. Then their underlying EDs d_U and d'_U are semantically equivalent.*

Proof. Let d and d' be semantically equivalent EDP satisfying $L_G(d) \cap L_P(d') = \varnothing$ and $L_P(d) \cap L_G(d') = \varnothing$. Suppose, for a contradiction, that d_U and d'_U are not semantically equivalent.

We first add (given) contour labels to d and d' so that they have the same given contour sets. More precisely, let

$$d_1 = d + (L_G(d') - L_G(d)) \quad \text{and} \quad d'_1 = d' + (L_G(d) - L_G(d')).$$

By repeated application of Lemma 5.5(i), $d \equiv_{\vDash} d_1$ and $d' \equiv_{\vDash} d'_1$; hence $d_1 \equiv_{\vDash} d'_1$.

Consider the underlying diagrams of d_1 and d'_1. By repeated application of Lemma 5.7, we have

$$(d_1)_U = d_U + (L_G(d') - L_G(d)) \quad \text{and} \quad (d'_1)_U = d'_U + (L_G(d) - L_G(d')).$$

Hence, by repeated application of Theorem 3.3(ii) for EDs, $(d_1)_U \equiv_{\vDash} d_U$ and $(d'_1)_U \equiv_{\vDash} d'_U$. Therefore, the underlying diagrams $(d_1)_U$ and $(d'_1)_U$ have the same contour set but are not semantically equivalent. Without loss of generality, there is an unshaded zone z in $(d_1)_U$ that is either shaded or missing in $(d'_1)_U$,

$$z \in Z^\circ((d_1)_U) \quad \text{and} \quad z \in Z^\bullet((d'_1)_U) \cup Z^m((d'_1)_U).$$

Consider the sets of zones in d_1 and d'_1 in the underlying zone z, $Z_{d_1}(z)$ and $Z_{d'_1}(z)$. Since z is unshaded in the underlying diagram $(d_1)_U$, there is a zone $\tilde{z} \in Z_{d_1}(z)$ that is unshaded.
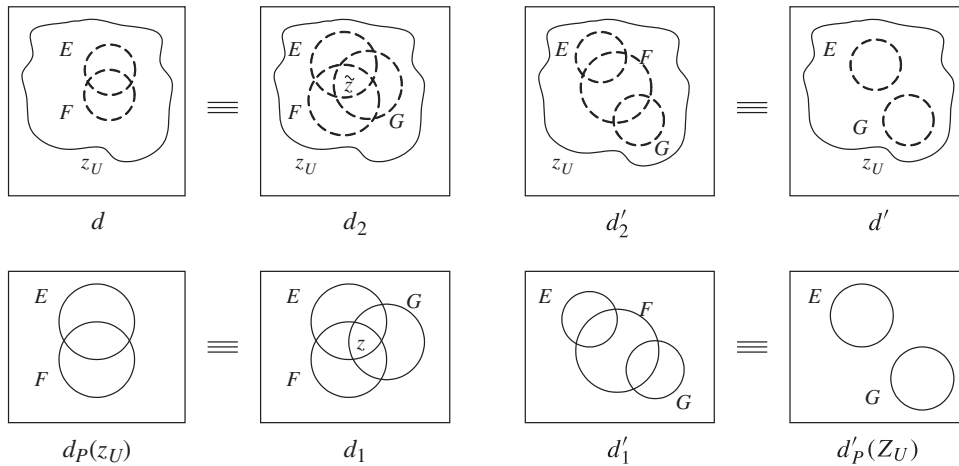
FIGURE 23. *Illustrating the diagrams in the proof of Theorem* 5.9.

However, $\tilde{z}$ is either shaded or missing in $Z_{d'_1}(z)$ depending on whether z is shaded or missing in $(d'_1)_U$.

Since $\tilde{z}$ is unshaded in d_1, there exists an interpretation (U, Ψ) that is a model for d_1 where $\Psi(\tilde{z}) \neq \varnothing$. However by the shaded zones and missing zones conditions, for every interpretation (U', Ψ') that is a model for d'_1, we have $\Psi'(\tilde{z}) = \varnothing$. Therefore, (U, Ψ) is not a model for d'_1 which contradicts $d_1 \equiv_\vDash d'_1$.

Therefore, $d_U \equiv_\vDash d'_U$, as required.   □

THEOREM 5.9. *Let d and d' be two EDP that have the same underlying diagram. Then d is semantically equivalent to d' if and only if, for each zone $z_U \in Z(d_U) = Z(d'_U)$ in the underlying diagram $d_U = d'_U$, the embedded projected diagrams $d_P(z_U)$ and $d'_P(z_U)$ are semantically equivalent.*

Proof. Let d and d' be two EDP that have the same underlying diagram.

Suppose that, for each zone $z_U \in Z(d_U) = Z(d'_U)$, the embedded projected diagrams $d_P(z_U)$ and $d'_P(z_U)$ are semantically equivalent. Let (U, Ψ) be an interpretation that is a model for d.

By Theorem 5.6, (U, Ψ) is a model for d_U and, for each zone $z_U \in Z(d)$, (U, Ψ^{z_U}) is a model for $d_P(z_U)$. Hence, (U, Ψ) is a model for d'_U and, for each zone $z_U \in Z(d'_U)$, (U, Ψ^{z_U}) is a model for $d'_P(z_U)$. Therefore, (U, Ψ) is a model for d' by Theorem 5.6.

Hence, d semantically entails d'. Reversing the roles of d and d' it follows that d' semantically entails d also, so d and d' are semantically equivalent.

Conversely, suppose that there is an underlying zone $z_U \in Z(d_U) = Z(d'_U)$ such that the embedded projected diagrams $d_P(z_U)$ and $d'_P(z_U)$ are not semantically equivalent as EDs. First note that, if z_U were shaded, then $d_P(z_U)$ and $d'_P(z_U)$ would be completely shaded and hence semantically equivalent. Therefore, we may assume that z_U is unshaded.

The various diagrams constructed in the proof are illustrated in Figure 23.

We first add contours to the embedded projected diagrams so that they have the same contour set. More precisely, let $d_1 = d_P(z_U) + (L(d'_P(z_U)) - L(d_P(z_U)))$ and $d'_1 = d'_P(z_U) + (L(d_P(z_U)) - L(d'_P(z_U)))$. Then, by Theorem 3.3(ii), $d_1 \equiv_\vDash d_P(z_U)$ and $d'_1 \equiv_\vDash d'_P(z_U)$.

We also add the corresponding projected contours into the zone z_U in d and d'. In other words, extending our notation to allow the addition of sets of projected contours in an underlying zone, let

$$d_2 = d + ((\Lambda_{d'}(z_U) - \Lambda_d(z_U)) \rightarrow z_U) \quad \text{and} \quad d'_2 = d' + ((\Lambda_d(z_U) - \Lambda_{d'}(z_U)) \rightarrow z_U).$$

Thus, $d \equiv_\vDash d_2$ and $d' \equiv_\vDash d'_2$, by Lemma 5.5(ii), and d_2 and d'_2 have the same intersectors of z_U, $\Lambda_{d_2}(z_U) = \Lambda_{d'_2}(z_U)$. Since we have added the same projected contours to z_U in d as we added (as given contours) to the embedded projected diagram $d_P(z_U)$, it follows that d_1 is the embedded projected diagram of d_2, $d_1 = (d_2)_P(z_U)$, and similarly $d'_1 = (d'_2)_P(z_U)$. In particular, there are bijections $w : Z_{d_2}(z_U) \to Z(d_1)$ and $w' : Z_{d'_2}(z_U) \to Z(d'_1)$.

Now d_1 and d'_1 are two EDs with the same contour sets that are not semantically equivalent (since $d_P(z_U) \not\equiv_\vDash d'_P(z_U)$). Without loss of generality, there exists an unshaded zone z in d_1 that is shaded or missing in d'_1,

$$z \in Z^\circ(d_1) \quad \text{and} \quad z \in Z^\bullet(d'_1) \cup Z^m(d'_1).$$

Let $\tilde{z}$ be the corresponding zone to z in z_U in d_2, $\tilde{z} = w^{-1}(z)$. Since z is unshaded in d_1 and the underlying zone z_U is unshaded in $(d_2)_U = d_U$, it follows that $\tilde{z}$ is unshaded in d_2. Hence, there is an interpretation (U, Ψ) that is a model for d_2 for which $\Psi_{d_2}(\tilde{z}) \neq \varnothing$. However, in d'_2, $\tilde{z}$ is either shaded or missing (depending on whether z is shaded or missing). For any interpretation (U', Ψ') that is a model for d'_2 we have $\Psi'(\tilde{z}) = \varnothing$. Hence, (U, Ψ) is not a model for d'_2, and so d_2 and d'_2 are not semantically equivalent. Therefore, d and d' are not semantically equivalent. $\quad\square$

5.3. *Redundancy in Euler diagrams with projections*

In EDP, removing projected contours is different from removing given contours. For projected contours, we need the notion of removing a contour from a (single) underlying zone. For given contours, we need to consider the embedded projected diagrams that lie in the various twins relative to the given contour. We begin by considering the operation of removing a projected contour label ℓ_p from an underlying zone (x_G, y_G) in Definition 5.10. If the context of ℓ_p extends beyond the underlying zone (x_G, y_G), then the projected contour set remains unchanged. If the context of ℓ_p does not contain more than (x_G, y_G) then ℓ_p is also removed from the projected contour label set. The definition of the shaded zone set follows that given for EDs: see Definition 3.2. Redundancy of projected contour labels is then defined, in Definition 5.11, in terms of this operation. Then Theorem 5.10 relates the redundancy of projected contours within underlying zones to redundancy of given contours in the corresponding embedded projected diagram.

DEFINITION 5.10. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d)\rangle$ be an EDP, let $\ell_p \in L_P(d)$ be a projected contour label, and let $(x_G, y_G) \in Z_U(d)$ be a zone in the underlying diagram d_U that lies in the context of ℓ_p. Then the EDP d with ℓ_p removed from (x_G, y_G), denoted $r_{\ell_p}^{(x_G,y_G)}(d)$, is the diagram $d' = \langle L_G(d'), L_P(d'), Z(d'), Z^\bullet(d')\rangle$ defined as follows:

  (i) $L_G(d') = L_G(d)$;
 (ii) if the context of ℓ_p comprises only the underlying zone $\kappa(\ell_p) = \{(x_G, y_G)\}$, then $L_P(d') = L_P(d) - \{\ell_p\}$; otherwise $L_P(d') = L_P(d)$;
(iii) for all underlying zones $z_U \neq (x_G, y_G)$, $Z_{d'}(z_U) = Z_d(z_U)$ and $Z_{d'}^\bullet(z_U) = Z_d^\bullet(z_U)$;
 (iv) $Z_{d'}(x_G, y_G) = \{(x_G, x_P - \{\ell_p\}, y_G, y_P - \{\ell_p\}) : (x_G, x_P, y_G, y_P) \in Z_d(x_G, y_G)\}$;
  (v) $Z_{d'}^\bullet(x_G, y_G) = \{(x_G, x_P - \{\ell_p\}, y_G, y_P - \{\ell_p\}) : r_{\ell_p,d}^{-1}(x_G, x_P - \{\ell_p\}, y_G, y_P - \{\ell_p\}) \subseteq Z_d^\bullet(x_G, y_G)\}$ where $r_{\ell_p,d}$ is the restriction to $Z(d)$ of the mapping $r_{\ell_p} : \mathcal{Z} \to \mathcal{Z}$ given by $(x_G, x_P, y_G, y_P) \mapsto (x_G, x_P - \{\ell_p\}, y_G, y_P - \{\ell_p\})$.

DEFINITION 5.11. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d)\rangle$ be an EDP, let $\ell_p \in L_P(d)$ be a projected contour label and let $(x_G, y_G) \in Z_U(d)$ be a zone in the underlying diagram d_U that lies in the context of ℓ_p. Then ℓ_p is *redundant* in (x_G, y_G) if $d \equiv_\vDash r_{\ell_p}^{(x_G,y_G)}(d)$.

THEOREM 5.10. *Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d)\rangle$ be an EDP, let $\ell_p \in L_P(d)$ be a projected contour label and let $(x_G, y_G) \in Z_U(d)$ be a zone in the underlying diagram d_U that*
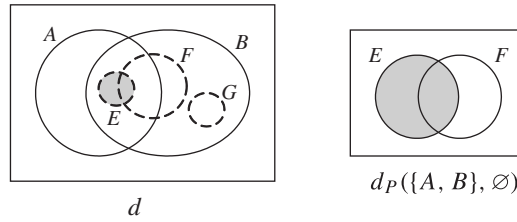
FIGURE 24. *Redundant projection in a zone.*

lies in the context of ℓ_p. Then ℓ_p is redundant in (x_G, y_G) if and only if ℓ_p is redundant, as a given contour, in the ED $d_P(x_G, y_G)$ that is the embedded projected diagram in (x_G, y_G).

Proof. Let $d' = r_{\ell_p}^{(x_G, y_G)}(d)$. Then d and d' have the same underlying diagrams.

By definition, ℓ_p is redundant in (x_G, y_G) if and only if $d \equiv_\vDash d'$. By Theorem 5.9, this is the case if and only if $d_P(z_U) \equiv_\vDash d'_P(z_U)$ for every underlying zone z_U. However, the embedded projected diagrams of d and d' are identical in every underlying zone except (x_G, y_G). Therefore, d is semantically equivalent to d' if and only if the embedded projected diagrams $d_P(x_G, y_G)$ and $d'_P(x_G, y_G)$ are semantically equivalent. But $d'_P(x_G, y_G)$ is precisely the diagram $d_P(x_G, y_G)$ with ℓ_p removed, $d'_P(x_G, y_G) = d_P(x_G, y_G) - \ell_p$. Thus, $d_P(x_G, y_G)$ and $d'_P(x_G, y_G)$ are semantically equivalent if and only if ℓ_p is redundant in $d_P(x_G, y_G)$.   □

This theorem is illustrated in Figure 24. The left-hand side of the figure shows that the projected contour label F is redundant in the underlying zone $(\{A, B\}, \varnothing)$ of d. The right-hand side shows that F is redundant when viewed as a given contour label in $d_P(\{A, B\}, \varnothing)$, the projected diagram of d embedded in $(\{A, B\}, \varnothing)$.

We provide a notion of redundancy of a projected contour label within the whole diagram in Definition 5.12 and show that it is a natural extension of redundancy within underlying zones in Theorem 5.11.

DEFINITION 5.12. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP and let $\ell_p \in L_P(d)$ be a projected contour label. Then ℓ_p is *redundant in d* if and only if $d \equiv_\vDash r_{\ell_p}(d)$ where $r_{\ell_p}(d)$ denotes the diagram obtained by successively removing ℓ_p from every zone in its context in d.

THEOREM 5.11. *Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP, and $\ell_p \in L_P(d)$. Then ℓ_p is redundant in d if and only if ℓ_p is redundant in every zone in the context of ℓ_p.*

Proof. Follows from the application of Theorem 5.10 to every underlying zone within the context of ℓ_p.   □

Recall that, when removing a contour label ℓ from an ED d (without projections), informally we may regard each pair of ℓ-twins as combining to form a single zone in $d - \ell$. If the zones in a pair of ℓ-twins represent the 'same information', that is, they are both shaded or both unshaded, then the resulting 'combined zone' in $d - \ell$ represents the common information. It may be the case that a pair of ℓ-twins represents different information; that is, one is shaded and the other is not. In this case, when removing ℓ from d, the resulting zone in $d - \ell$ is unshaded: see Figure 5. We can think of this as removing the information that is not common to both zones in a pair of ℓ-twins.

When considering EDP, different information may be represented by different embedded projected diagrams in the two twins. This situation is illustrated in Figure 25. In d_1 the two
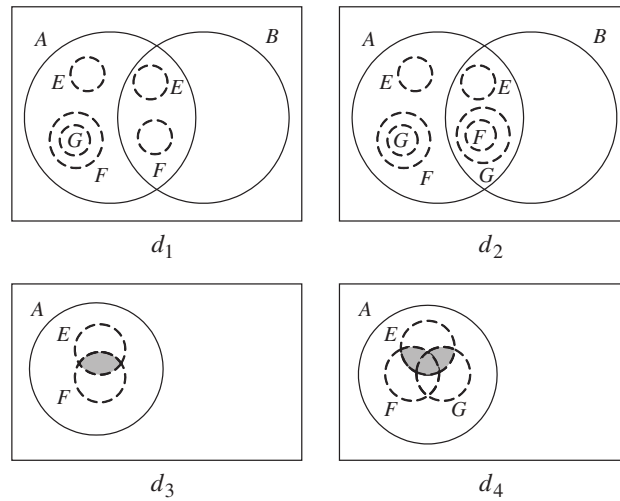
FIGURE 25. *Different information represented in B-twins.*

embedded projected diagrams assert the following:

$$d_P(\{A\}, \{B\}): \quad \Psi(E) \cap \Psi(F) = \varnothing, \quad \Psi(G) \subseteq \Psi(F), \quad \Psi(E) \cap \Psi(G) = \varnothing,$$

$$d_P(\{A, B\}, \varnothing): \quad \Psi(E) \cap \Psi(F) = \varnothing.$$

To form the diagram $d_1 - B$, we wish to retain, within the contour A, the embedded projected diagram which represents the information common to the two embedded projected diagrams $d_P(\{A\}, \{B\})$ and $d_P(\{A, B\}, \varnothing)$. Recall that the semantic coproduct of two EDs captures the semantic information common to both diagrams. Hence, the embedded projected diagram within the contour A is the semantic coproduct of the two embedded projected diagrams $d_P(\{A\}, \{B\})$ and $d_P(\{A, B\}, \varnothing)$. This is shown in diagram d_3 in Figure 25. The same process applied to removing B from the diagram d_2 in Figure 25 produces the diagram d_4.

DEFINITION 5.13. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP and let $\ell_g \in L_G(d)$ be a given contour in d. The EDP d with ℓ_g removed, $r_{\ell_g}(d) = d - \ell_g$, is the EDP d' satisfying the following conditions.

(i) The underlying diagram of d' is the underlying diagram of d with ℓ_g removed, $d'_U = d_U - \ell_g$.

(ii) For each zone (x_G, y_G) in the underlying diagram d'_U, $(x_G, y_G) \in Z(d'_U)$, the projected diagram $d'_P(x_G, y_G)$ embedded in (x_G, y_G) is the semantic coproduct of the embedded projected diagrams $d_P(x_G \cup \{\ell_g\}, y_G)$ and $d_P(x_G, y_G \cup \{\ell_g\})$,

$$d'_P(x_G, y_G) = d_P(x_G \cup \{\ell_g\}, y_G) \sqcup d_P(x_G, y_G \cup \{\ell_g\}).$$

EXAMPLE 5.6. Figure 26 shows the removal of the given contour B from the diagram d. To form $d - B$, we first form the underlying diagram, $d_U - B$ and then consider each pair of B-twins in turn.

(1) The B-twins $(\{A\}, \{B, C\})$ and $(\{A, B\}, \{C\})$. Each zone is shaded and contains an empty embedded projected diagram. Hence, the corresponding zone in $d - B$, $(\{A\}, \{C\}) = r_B(\{A\}, \{B, C\}) = r_B(\{A, B\}, \{C\})$, is also shaded and contains an empty embedded projected diagram.
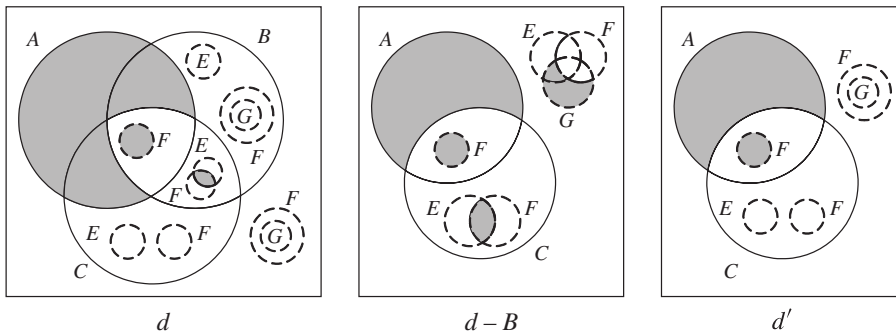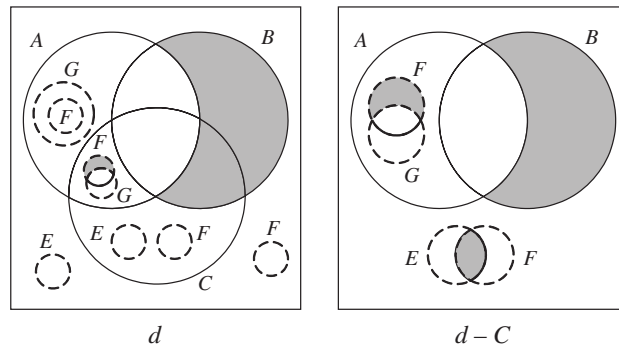
FIGURE 26. *Removing a given contour.*

FIGURE 27. *A redundant given contour.*

(2) The B-twins $(\{B\}, \{A, C\})$ and $(\varnothing, \{A, B, C\})$. The semantic coproduct of the two embedded projected diagrams is the Venn form of the embedded projected diagram in $(\varnothing, \{A, B, C\})$.

(3) The B-twins $(\{B, C\}, \{A\})$ and $(\{C\}, \{A, B\})$. The embedded projected diagrams in these two zones are semantically equivalent so their semantic coproduct is the corresponding Venn diagram with contour labels E and F.

(4) The B-twins $(\{A, B, C\}, \varnothing)$ and $(\{A, C\}, \{B\})$. The embedded projected diagram in $(\{A, C\}, \{B\})$ is the completely shaded diagram with no contours $d^\bullet$. The embedded projected diagram in $(\{A, B, C\}, \varnothing)$ contains a single contour F where the zone inside F is shaded. The semantic coproduct of these two diagrams is the second diagram (with single contour F).

The diagram d' on the right-hand side in Figure 26 is semantically equivalent to $d - B$. To form d' we have replaced each of the embedded projected diagrams in $d - B$ with its normal form. Our definition of the semantic coproduct produces a Venn diagram although the corresponding normal form is often easier to 'read'. Thus, the semantically equivalent diagram d' is easier to read than the diagram $d - B$ that arises from removing B from d.

DEFINITION 5.14. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP and let $\ell_g \in L_G(d)$ be a given contour label in d. Then ℓ_g is *redundant* in d if $d \equiv_\models r_{\ell_g}(d) = d - \ell_g$.

EXAMPLE 5.7. In Figure 27, the contour C is redundant in the diagram d. The diagram $d - C$ is obtained in a similar manner to that described in Example 5.6. When forming $d - C$, the information in each pair of C-twins (shading and embedded projected diagrams) is the same in each case. As a result, no information is lost when removing C, so $d \equiv_\models d - C$.

We can see this more precisely by considering each C-twin in turn. For example, consider the C-twins $(\{A, C\}, \{B\})$ and $(\{A\}, \{B, C\})$. Let (U, Ψ) be an interpretation that is a model for d. Then we have

$$\Psi(A) \cap \overline{\Psi(B)} \cap \Psi(C) \cap \Psi(F) \cap \overline{\Psi(G)} = \varnothing$$

from the shaded zones condition and

$$\Psi(A) \cap \overline{\Psi(B)} \cap \overline{\Psi(C)} \cap \Psi(F) \cap \overline{\Psi(G)} = \varnothing$$

from the missing projected zones condition. The conjunction of these is equivalent to

$$\Psi(A) \cap \overline{\Psi(B)} \cap \Psi(F) \cap \overline{\Psi(G)} = \varnothing$$

which is part of the shaded zones condition for $d - C$.

THEOREM 5.12. *Let* $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ *be an EDP, and let* ℓ_g *be a given contour in* d. *Then* ℓ_g *is redundant in* d *if and only if:*
   (i) ℓ_g *is a splitting label for the underlying diagram* d_U; *and*
   (ii) *for each pair of* ℓ_g-*twins in the underlying diagram,* z_1 *and* z_2, *the embedded projected diagrams* $d_P(z_1)$ *and* $d_P(z_2)$ *have the same normal form (as EDs).*

Proof. Suppose that ℓ_g is redundant in d. Then $d \equiv_\vDash d - \ell_g$ by definition so the underlying diagrams of d and $d - \ell_g$ are semantically equivalent by Theorem 5.8, $d_U \equiv_\vDash (d - \ell_g)_U$. However, the underlying diagram of $d - \ell_g$ is $d_U - \ell_g$, so $d_U \equiv_\vDash d_U - \ell_g$. Hence, l_g is redundant in d_U. Therefore, by Theorem 4.3, ℓ_g is a splitting label for d_U so part (i) holds.

Suppose that there is a pair of ℓ_g-twins, z_1 and z_2, in d such that the embedded projected diagrams $d_P(z_1)$ and $d_P(z_2)$ do not have the same normal form. Then $d_P(z_1) \not\equiv_\vDash d_P(z_2)$ by Theorem 4.8. Without loss of generality, we may assume that there is a zone which is unshaded in $d_P(z_1)$ but is shaded or missing in $d_P(z_2)$. Since the semantic coproduct of two diagrams contains only the shading that is common to both diagrams (once they have been arranged in Venn form with the same contour set), it follows that $d_P(z_2)$ is not semantically equivalent to the coproduct $d_P(z_1) \sqcup d_P(z_2)$.

By definition, $d_P(z_1) \sqcup d_P(z_2)$ is the projected diagram embedded in z in $d - \ell_g$, where $z = r_{\ell_g}(z_1) = r_{\ell_g}(z_2)$. The schematic diagrams involved in this part of the proof are illustrated in Figure 28. The top 'row' of the diagram illustrates zones in the various EDP and the bottom 'row' gives the (symbolic representation of the) embedded projected diagrams within those zones.

Add the contour ℓ_g back into $d - \ell_g$ to give $(d - \ell_g) + \ell_g$. The underlying diagrams of d and $(d - \ell_g) + \ell_g$ are semantically equivalent since ℓ_g is a splitting label for d_U by part (i). These underlying diagrams also have the same zone sets; hence they are the same diagram. By construction, the projected diagram embedded in both z_1 and z_2 in $(d - \ell_g) + \ell_g$ is the semantic coproduct $d_P(z_1) \sqcup d_P(z_2)$. Thus, the projected diagrams embedded in z_2 in d and $(d - \ell_g) + \ell_g$ are not semantically equivalent. Hence, by Theorem 5.9, d and $(d - \ell_g) + \ell_g$ are not semantically equivalent. However, $d - \ell_g$ is semantically equivalent to $(d - \ell_g) + \ell_g$ by Lemma 5.5. Therefore, d and $d - \ell_g$ are not semantically equivalent, which is a contradiction. Thus, part (ii) holds.

Conversely, suppose that ℓ_g is a splitting label for the underlying diagram d_U, and for each pair of ℓ_g-twins in the underlying diagram, z_1 and z_2, the embedded projected diagrams $d_P(z_1)$ and $d_P(z_2)$ have the same normal form (as EDs). Let (U, Ψ) be an interpretation that is a model for d. Then, by Theorem 5.6, (U, Ψ) is a model for the underlying diagram d_U. Since ℓ_g is a splitting label for d_U, it is redundant in d_U by Theorem 4.3. Therefore, (U, Ψ) is a model for $d_U - \ell_g = (d - \ell_g)_U$.
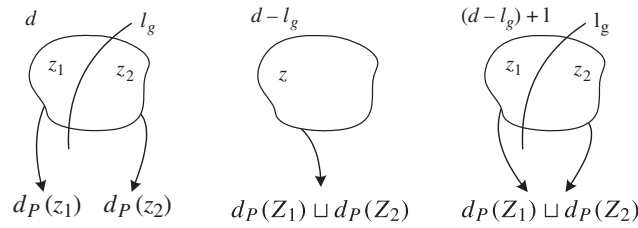
FIGURE 28. *Illustrating the diagrams in the proof of Theorem 5.12.*

Let z_1 and z_2 be a pair of ℓ_g-twins in d_U, say $z_1 = (x_G \cup \{\ell_g\}, y_G)$ and $z_2 = (x_G, y_G \cup \{\ell_g\})$ where $z = (x_G, y_G)$ is a zone in the underlying diagram $d_U - \ell_g$. By Theorem 5.6, the modified interpretation (U, Ψ^{z_1}) is a model for $d_P(z_1)$ and the modified interpretation (U, Ψ^{z_2}) is a model for $d_P(z_2)$.

Since $d_P(z_1)$ and $d_P(z_2)$ have the same normal form (as EDs), they are semantically equivalent and hence they are both semantically equivalent to their semantic coproduct. This semantic coproduct is the projected diagram embedded in the zone z of $d - \ell_g$, $(d - \ell_g)_P(z)$. Hence, the modified interpretations (U, Ψ^{z_1}) and (U, Ψ^{z_2}) are each models for $(d - \ell_g)_P(z)$.

By adding contours and missing zones where necessary, we may assume that $d_P(z_1)$ and $d_P(z_2)$ are Venn diagrams with the same contour label sets. (Strictly, we replace $d_P(z_1)$ and $d_P(z_2)$ with semantically equivalent diagrams in Venn form and with the same contour label set.) This means that we may assume that $d_P(z_1)$ and $d_P(z_2)$ and their semantic coproduct $(d - \ell_g)_P(z)$ are all the same diagram. Now for any zone z^* in this diagram,

$$\Psi^{z_1}(z^*) = \Psi(z^*) \cap \Psi(\ell_g) \quad \text{and} \quad \Psi^{z_2}(z^*) = \Psi(z^*) \cap \overline{\Psi(\ell_g)}.$$

Hence, $\Psi^{z_1}(z^*) \cup \Psi^{z_2}(z^*) = \Psi(z^*)$. If z^* is a shaded zone of $(d - \ell_g)_P(z)$, then $\Psi^{z_1}(z^*) = \varnothing$ and $\Psi^{z_2}(z^*) = \varnothing$ since (U, Ψ^{z_1}) is a model for $d_P(z_1)$ and (U, Ψ^{z_2}) is a model for $d_P(z_2)$. It follows that $\Psi(z^*) = \varnothing$ for all shaded zones z^* of $(d - \ell_g)_P(z)$. Therefore, (U, Ψ^z) is a model for $(d - \ell_g)_P(z)$.

We have shown that (U, Ψ) is a model for $(d - \ell_g)_U$ and, for each zone z of $(d - \ell_g)_U$, the modified interpretation (U, Ψ^z) is a model for $(d - \ell_g)_P(z)$. Hence, by Theorem 5.6, (U, Ψ) is a model for $d - \ell_g$.

Therefore, every model for d is a model for $d - \ell_g$, and so ℓ_g is redundant in d.   □

Finally, we provide the definition of the normal form of an EDP in Definition 5.15. Then we give an algorithm for computing the normal form of any EDP and show that the normal form of any EDP is semantically equivalent to the original diagram in Theorem 5.13. We need to be careful with the definition of the normal form since it is not necessarily the case that a given contour which is redundant in the underlying diagram d_U is redundant in d as the following example illustrates.

EXAMPLE 5.8. Consider the given contour label C in the EDP d in Figure 29. In the underlying diagram d_U, the contour C is redundant. However, C is not redundant in d because the embedded projected diagrams in the C-twins $(\{C\}, \{A, B\})$ and $(\varnothing, \{A, B, C\})$ do not have the same normal form.

In fact, the diagram d in Figure 29 has no redundant contours. To obtain the normal form for d we will remove the shaded zones to obtain the diagram d^N shown in Figure 30. This figure also shows that normal form of the underlying diagram $(d_U)^N$ is not the same as the underlying diagram of the normal form $(d^N)_U$. Since C is redundant in d_U, it does not appear in the normal form $(d_U)^N$. However, C is not redundant in d, so C appears in the normal form d^N and hence also in its underlying diagram $(d^N)_U$.
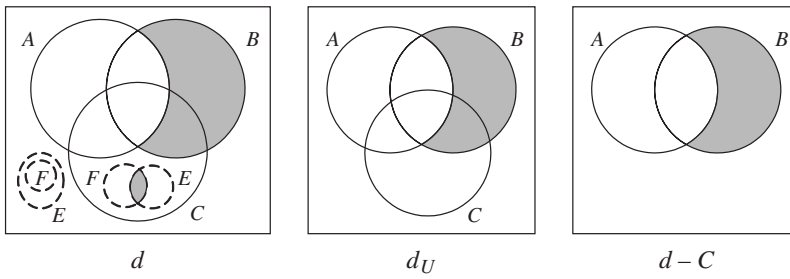
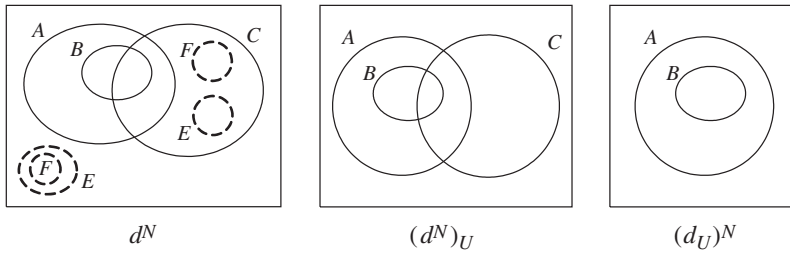FIGURE 29. *Redundant contour in d_U not redundant in d.*

FIGURE 30. *The normal form of d in Figure* 29.

DEFINITION 5.15. Let $d = \langle L_G(d), L_P(d), Z(d), Z^\bullet(d) \rangle$ be an EDP. Then d is in *normal form* if either it contains no contours or:

  (i) the underlying diagram d_U has no shaded zones outside nomads: if $z \in Z^\bullet(d_U^N)$ then $z \in Z_{in_{\mathcal{N}}}(d_U)$;

 (ii) if the underlying diagram d_U has any nomads, these are exiled; that is, if $\mathcal{N}(d_U) \neq \varnothing$, then $Z^\bullet(d_U) = \{(\mathcal{N}(d_U), L(d_U) - \mathcal{N}(d_U))\}$;

(iii) for each underlying zone z in $Z(d_U)$, the embedded projected diagram $d_P(z)$ is in normal form as an ED;

(iv) if a given contour ℓ_g is a splitting label for the underlying diagram d_U, then there exists a pair of ℓ_g-twins in d_U, z_a and z_b, with different embedded projected diagrams, $d_P(z_a) \neq d_P(z_b)$.

THEOREM 5.13. *Let d be an EDP and let d^N be the EDP that is obtained from d by applying the following steps.*

  (1) Replace d with diagram d_1 which has the same underlying diagram as d and is such that, for each underlying zone z, its embedded projected diagram $(d_1)_P(z)$ is the diagram $d_P^N(z)$ which is the normal form (as an ED) of $d_P(z)$.

  (2) In the underlying diagram of d_1, remove any shaded zones that do not lie within a nomad; that is, replace $(d_1)_U$ with the diagram obtained by removing each zone in $Z^\bullet_{out_{\mathcal{N}}}((d_1)_U)$. (From step (1), it follows that the embedded projected diagram in any shaded zone is the diagram $d^\bullet$ which is shaded and contains no contours.) For any unshaded zones of the underlying diagram, the embedded projected diagram remains unchanged.

  Call the resulting diagram d_2.

  (3) Exile any nomads in the underlying diagram of d_2, giving diagram d_3.

  As in step (2), the embedded projected diagrams (outside nomads) in d_3 are the same as the embedded projected diagrams in d_2.

(4) *Consider each splitting label ℓ in the underlying diagram of d_3 in turn. There are two possibilities.*

    (a) *For every pair of ℓ-twins, z_a and z_b say, in $(d_3)_U$ the embedded projected diagrams are equal, $(d_3)_P(z_a) = (d_3)_P(z_b)$.*

    (b) *There is a pair of ℓ-twins, z_a and z_b say, in $(d_3)_U$ for which the embedded projected diagrams are not equal, $(d_3)_P(z_a) \neq (d_3)_P(z_b)$.*

Remove from d_3 all of those splitting labels satisfying the condition in possibility (a). *The resulting diagram is denoted d^N.*

    Then d^N is an EDP in normal form, and d is semantically equivalent to d^N.

    Proof. First, it is clear by construction that d^N is in normal form.

    It follows from Theorem 5.9 that d and d_1 are semantically equivalent. Note that, in forming d_1, any projected contours in shaded underlying zones are removed since the normal form of a wholly shaded ED has no contours. By Theorem 4.7, the underlying diagrams of d_2 and d_1 are semantically equivalent. Also, for any underlying zone $z \in Z(d_2)$, the embedded projected diagrams $(d_2)_P(z)$ and $(d_1)_P(z)$ are equal. Therefore, d_2 is semantically equivalent to d_1; this follows from Theorem 5.9 together with the observation that removing a shaded underlying zone does not change the semantics predicate as that zone becomes a missing underlying zone.

    Similarly, exiling the nomads in the underlying diagram of d_2 does not alter any embedded projected diagrams and produces a diagram d_3 such that $(d_2)_U \equiv_\vDash (d_3)_U$. Hence, d_3 is semantically equivalent to d_2.

    In step (5) a splitting label ℓ in the underlying diagram of d_3 is removed from d_3 only when it is redundant in d_3, by Theorem 5.12. Therefore, d^N is semantically equivalent to d_3.

    Hence, d^N is semantically equivalent to d.   □

Figure 31 illustrates the conversion of an EDP into an EDP in normal form via the steps in Theorem 5.13. Within each of the underlying zones of the top EDP, the projected contours are all converted to be in ED normal form (steps (1) and (2) of Theorem 5.13), as shown by the semantically equivalent diagram given in the middle of the figure. This enables redundancy of the given contours to be more easily observed within the EDP. Any such redundant given contours can then be removed from the EDP, shaded underlying zones containing no projections can be removed and given contour nomads can be exiled. We obtain the diagram at the bottom of the figure.

6. *Related work*

This paper is a significant extension and improvement over the conference paper [**26**] which introduced the normal form for the EDS system only. In [**47**] model-based semantics were provided for spider diagrams, together with a set of sound and complete inference rules. The effect of variation of inference rules on the provision of shortest proofs within ED systems was investigated in [**74**], utilising an A^*-based algorithm. Swoboda and Allwein [**76**] present a graph-transformation-based method for the verification of heterogeneous first-order logic and Euler/Venn proofs.

    The distinction between the (drawn) concrete level and the abstract level was considered in [**43**]. The common approach in this area is to develop reasoning systems which act at the abstract level; this brings with it associated problems of the generation of a concrete diagram from abstract diagrams (and subsequently the challenge of the dynamic generation of diagrams which differ by some transformation, whilst preserving the mental map: considered for graphs by Eades *et al.* [**21**]). Alternative equivalent abstractions for EDs were provided in [**24**], including the consideration of diagrams as sequences of curve additions instead of as static objects (using a set of zones). Recently, a new graph-based abstraction for spider
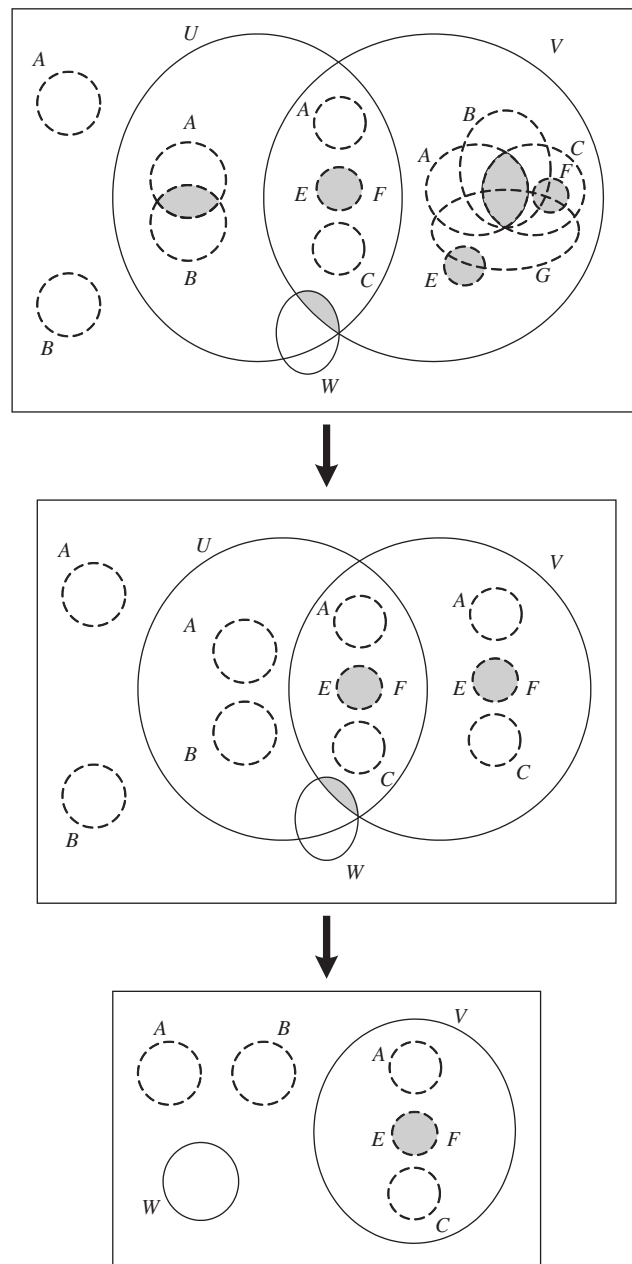
FIGURE 31. *Obtaining the normal form.*

diagrams was also introduced, in [**7**], together with a translation of the algebraic reasoning system into a graph transformation system.

Grunbaum [**32**] and Moore [**64**] provided means of constructing families of Venn diagrams. In [**63**] they produce exhaustive lists of simple monotone Venn diagrams that have some symmetry (non-trivial isometry) when drawn on the sphere. Hamburger *et al.* [**35**] linked the existence of the extensibility of Venn diagrams to the existence of Hamiltonian cycles in their dual graphs. Johnson and Pollack [**56**] link the complexity of drawing Venn diagrams with the problem of hypergraph planarity. Bertault and Eades [**6**] phrase the question of
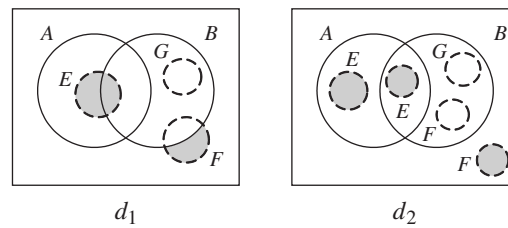
FIGURE 32. *Different drawings of an Euler diagram with projections.*

ED generation in terms of drawing hypergraphs in the subset standard. The problem of ED generation from an abstract diagram or set system has been investigated, with foundational graph-based solutions presented by Flower *et al.* [**29**] and Chow [**10**]. Variations of the wellformedness conditions imposed (for example, relaxing conditions such as: there are finite numbers of transverse intersection points) provides problem variations. The conditions imposed were natural topological conditions, originally imposed to aid in human comprehension and reasoning tasks; there have been investigations into the effect on user comprehension of relaxation of such wellformedness conditions in [**27**]. Extensions of the methodology for ED generation to admit further variations of conditions imposed have been developed in [**72**] and [**69**], for example.

Cohn *et al.* have developed logical calculi for representing and reasoning with qualitative spatial relations over regions; a survey was provided in [**11**]. Egenhofer [**22**] considered the equivalence of topological relations, and computational problems involved in developing an inference system which takes basic topological relations such as 'A overlaps B' and 'B contains C' between simply connected regions in the plane, have been extensively investigated [**62**] in relation to geographic information systems.

Barwise and Echemendy [**3**, **4**] consider heterogeneous reasoning with Hyperproof, permitting and even requiring logical reasoning with a mix of symbolic and diagrammatic logics. Other diagrammatic systems include [**37**, **49**, **50**]. Conceptual graphs from Sowa [**1**], based on the works of Peirce, provides a distinct form of diagrammatic logic. The ALC description logic has been converted into a diagrammatic logic in [**15**]. Conceptual spider diagrams [**14**] propose a system utilising a combination of the features from spider diagrams and conceptual graphs.

6.1.   *Variations on the proposed system*

Compound diagrams are built from unitary diagrams using logical connectives (disjunction, conjunction and possibly negation). For example, $d_1 \vee (d_2 \wedge \neg d_3)$ is a compound diagram, where each d_i is a unitary or compound diagram. Two normal forms for compound diagrams, called 'literal conjunctive/disjunctive normal form', were introduced in [**73**] to obtain a completeness proof for a system of compound EDS. These normal forms are akin to conjunctive and disjunctive normal forms in propositional logic.

John's thesis [**54**] provided a foundation for this work, introducing some basic concepts such as nomads; in [**55**] metrics to try to capture a notion of clutter in EDs were provided together with some empirical justification. In [**44**], a spider diagram reasoning system was introduced, viewable as a precursor to [**47**]. A notion of 'border contours' for Venn diagrams was introduced in [**46**]. Our definition of splitting label essentially extends the notion of border contour to EDs.

Previously [**30**, **31**, **52**–**54**], abstract EDP have been defined by modifying the definition of abstract EDs via the introduction of a partition of the contour label set into given and projected contours. However, with this approach there is a mismatch between the syntax and semantics. For example, the zone inside contour G in d_2 in Figure 32 is described as

$(\{B,G\},\{A,E,F\})$ in the usual way. Semantically, this zone denotes the set $\overline{A}\cap B\cap\overline{F}\cap G$ but the syntactic description also includes the contour E. A consequence of this mismatch is that one then needs to modify how an interpretation (U,Ψ) interprets zones relative to the diagram d. To avoid this issue, we have taken a different approach, defining a zone z by describing separately the given and projected contour labels that contain z and exclude z, where the excluding projected contours are only those required by the semantics. Thus, the zone inside contour G in d_2 in Figure 32 is described as a quadruple $(\{B\},\{G\},\{A\},\{F\})$ which says that this zone is inside given contour B and projected contour G and is outside given contour A and projected contour F.

In [**28**], nested EDs were defined, making precise the notion of embedding a diagram d_2 within a zone z^* of another diagram d_1, denoted $d_2 \xrightarrow{z^*} d_1$. This is closely related to the embedded projected diagrams: each embedded projected diagram $d_P(x_G,y_G)$ is embedded in the zone (x_G,y_G) of the underlying diagram d_U, $d_P(x_G,y_G) \xrightarrow{(x_G,y_G)} d_U$, except that the embedding relation does not capture the distinction between given and projected contours.

6.2. *Variations to syntax and semantics*

We have chosen to draw our EDP so that each (drawn) projected contour lies within a zone of the underlying diagram. However, there may be more than one way of drawing a given (abstract) diagram. For example, both of the diagrams in Figure 32 have the same abstract syntactic description. Note that the diagram d_2 repeats the diagram shown in Figure 18. In d_1, the distinct projected contours have different labels so that each projected contour label is represented by a single drawn contour in the diagram.

In [**54**], John distinguished two kinds of abstract syntax which he calls the 'course-grained' and 'fine-grained' abstract syntax. His 'fine-grained' abstract syntax distinguishes between the diagrams d_1 and d_2 in Figure 32 whereas his 'course-grained' abstract syntax does not. Our abstract syntax follows John's course-grained version (although we have defined zones differently to John).

Our convention for drawing diagrams, where each projected contour is drawn within a zone of the underlying diagram, can lead to certain issues. Consider, for example, the EDP d, with four zones, defined as follows:

(1) $L_G(d) = \{A,B\}$;

(2) $L_P(d) = \{E\}$;

(3) $Z(d) = \{(\varnothing,\varnothing,\{A,B\},\{E\}),(\varnothing,\{E\},\{A,B\},\varnothing),(\{A\},\{E\},\{B\},\varnothing),(\{B\},\{E\},\{A\},\varnothing)\}$;

(4) $Z^\bullet(d) = \{(\varnothing,\{E\},\{A,B\},\varnothing)\}$.

The context of the projected contour label E is the whole underlying diagram, $\kappa(E) = \{(\varnothing, \{A,B\}),(\{A\},\{B\}),(\{B\},\{A\})\}$. The diagram d_1 in Figure 33 gives a drawn representation of d.

If we attempt to draw d so that each projected contour is drawn within a zone of the underlying diagram, we obtain the diagram d_2 in Figure 33. Here there are three projected contours labelled E, two of which coincide with the given contours labelled A and B, respectively. We may regard d_2 as not well-formed (under a sensible definition of 'well-formed' for drawn EDP) since it is not possible to tell that E is a projected contour when it coincides with the given contours A or B. The diagram d_3 is semantically equivalent to d_1 and d_2 and each projected contour is contained within a zone of the underlying diagram. However, d_3 represents a syntactically different abstract diagram to d_1 and d_2.

For EDs without projections, there are alternative ways of defining the semantics predicate. In particular, the missing zones condition (Definition 2.5) is known to be equivalent (see [**47**], for example) to the 'plane tiling condition' which states that all elements lie within sets denoted
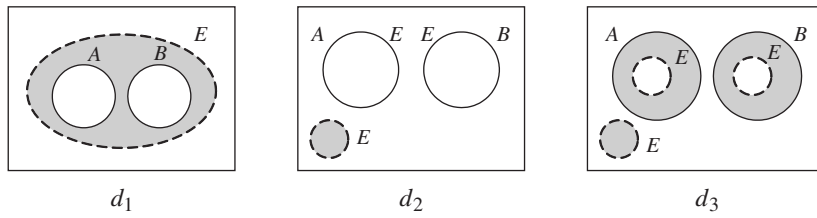
FIGURE 33. *Drawing Euler diagrams with projections.*

by zones of d:

$$\bigcup_{z \in Z(d)} \Psi(z) = U.$$

In [**54**], John uses this plane tiling condition to define the semantics predicate for EDP.

Let d be an EDP. The plane tiling condition for the underlying diagram d_U says that the union of sets represented by the underlying zones equals the universe

$$\bigcup_{z \in Z_U(d)} \Psi(z) = U$$

and this is equivalent to our missing underlying zones condition.

We explore this connection for EDP referring back to the diagram d in Figure 21 and Table 5.2 which listed the zones of d and their interpretations. The plane tiling condition for d asserts that the union of the sets in the fourth column of Table 5.2 is equal to the universe U. However, for arbitrary sets, the union of the sets in the fourth column of Table 5.2, the interpretations of the zones in d, is not equal to the union of sets in the second column, the interpretations of the underlying zones. This is because the sets

$$A \cap B \cap \overline{C} \cap E \cap F \quad \text{and} \quad \overline{A} \cap \overline{B} \cap C \cap F \cap \overline{G}$$

are missing from the right-hand column. However, these are precisely the sets that are asserted to be empty by the missing projected zones condition.

Thus, the plane tiling condition for d makes the same assertion as the *two* missing zones conditions (for underlying and projected zones). For a general EDP, this is captured by the following theorem.

THEOREM 6.1. *Let d be an EDP. The missing zones condition for d is equivalent to the plane tiling condition for d which says that all elements lie within sets denoted by zones of d:*

$$\bigcup_{z \in Z(d)} \Psi(z) = U.$$

7. *Conclusion*

We have provided normal forms for two ED systems, EDS and EDP. This provides a unique representative amongst the classes of semantically equivalent diagrams, and for each system we provide a procedure for transforming any given diagram into its normal form, thereby permitting an easy equivalence check. Along the way we develop general machinery which will be useful to the field, and observe that even conservative system extensions require a revisiting of all notions of syntactic and semantic redundancy due to potential interactions.

In a little more detail, we explore the syntax and semantics of diagram manipulation: Theorem 3.2 shows the independence of the order of contour removal at the syntactic level, Theorem 3.3 provides results on inference and equivalence of manipulation rules in EDS and Theorem 3.4 shows the semantic equivalence of any diagram d with d', which is d with its nomads exiled. We introduce the notion of a semantic coproduct to capture the idea of the common information contained in two diagrams; Theorem 3.5 shows that the semantic coproduct satisfies a 'universal property' with respect to semantic entailment which explains the 'semantic coproduct' terminology. Theorem 4.3 shows the equivalence between the syntactic and semantic conditions to capture redundancy of contours within EDS. Theorems 4.7 and 4.8 provide the normal form for the system and show equivalence results.

Introducing projections allows greater choice for representing information within the system but complicates the process of obtaining a normal form. We introduce a new syntax for EDP that allows a more natural semantics. Theorems 5.6–5.9 relate equivalence of EDP to equivalence of sets of EDS (relating to the underlying ED and its embedded projected diagrams). Then Theorems 5.10–5.12 identify redundancies of projected contours. Finally, Theorem 5.13 provides the required normal form for EDP, together with results of semantic equivalences. As illustrated in Figure 31, the normal form can afford a considerable syntactic simplification in the representation of information in EDP.

As we noted in §1, EDs form the basis for a variety of diagrammatic notations that include additional syntax to enrich the system and increase expressiveness. The work of this paper form a basis for a more systematic study of equivalences in ED-based systems. The next steps in this direction are to introduce graph-based features into the notation and extend the ideas of this paper to spider diagrams. Adding syntax to represent existence of elements introduces further complexity; for example, a syntactic description of the semantic coproduct is somewhat less obvious. Diagrammatic systems that are used in software specification and ontology modelling also contain syntax for relations between sets and a longer-term goal of this work is to provide a general framework for considering diagrammatic equivalences in these systems.

References

1. *Knowledge representation: logical, philosophical, and computational foundations* (Brooks Cole Publishing Co., Pacific Grove, CA, 2000).

2. J. F. ALLEN, 'Maintaining knowledge about temporal intervals', *Commun. ACM* 26 (1983) no. 11, 832–843.

3. J. BARWISE and J. ETCHEMENDY, *Hyperproof* (CSLI Press, Stanford, CA, 1994).

4. J. BARWISE and J. ETCHEMENDY, 'Visual information and valid reasoning', *Logical reasoning with diagrams* (eds G. Allwein and J. Barwise; Oxford University Press, Oxford, 1996) 3–25.

5. J. BARWISE and E. HAMMER, 'Diagrams and the concept of logical system', *Logical reasoning with diagrams* (eds G. Allwein and J. Barwise; Oxford University Press, Oxford, 1996).

6. F. BERTAULT and P. EADES, 'Drawing hypergraphs in the subset standard', *Proceedings of the 8th International Symposium on Graph Drawing*, Lecture Notes in Computer Science 1984 (Springer, Berlin, 2000) 164–169.

7. P. BOTTONI, A. FISH and F. PARISI-PRESICCE, 'Spider graphs: a graph transformation system for spider diagrams', *Softw. Syst. Model.*10.1007/s10270-013-0381-1.

8. P. BOTTONI and A. FISH, 'Extending spider diagrams for policy definition', *J. Vis. Lang. Comput.* 24 (2013) no. 3, 169–191.

9. P. CHAPMAN, G. STAPLETON and A. DELANEY, 'On the expressiveness of second-order spider diagrams', *J. Vis. Lang. Comput.* 24 (2013) 327–349.

10. S. C. CHOW, 'Generating and drawing area-proportional Euler and Venn diagrams', PhD Thesis, University of Victoria, 2007.

11. A. G. COHN, B. BENNETT, J. GOODAY and N. M. GOTTS, 'Qualitative spatial representation and reasoning with the region connection calculus', *Geoinformatica* 1 (1997) no. 3, 275–316.

12. C. COLLINS, G. PENN and S. CARPENDALE, 'Bubble sets: revealing set relations with isocontours over existing visualisations', *IEEE Trans. Vis. Comput. Graphics* 15 (2009) no. 6, 1009–1016.

13. G. CORDASCO, R. DE CHIARA and A. FISH, 'Interactive visual classification with Euler diagrams', *Proc. VL/HCC 2009* (IEEE Computer Society Press, Los Alamitos, CA, 2009) 185–192.

14. F. DAU and A. FISH, 'Conceptual spider diagrams', *Proc. ICCS 2008*, Lecture Notes in Computer Science 5113 (Springer, New York, 2008) 104–118.

15. F. DAU and P. EKLUND, 'A diagrammatic reasoning system for the description logic ALC', *J. Vis. Lang. Comput.* 19 (2008) no. 5, 539–573.

16. F. DAU and A. FISH, 'Conceptual spider diagrams', *16th International Conference on Conceptual Structures*, Lecture Notes in Computer Science 5113 (Springer, New York, 2008) 104–118.

17. R. DECHIARA, U. ERRA and V. SCARANO, 'VennFS: A Venn diagram file manager', *Proceedings of Information Visualisation* (IEEE Computer Society Press, Los Alamitos, CA, 2003) 120–126.

18. A. DELANEY and G. STAPLETON, 'Spider diagrams of order', *Proceedings of the VLL 2007 Workshop on Visual Languages and Logic*, CEUR-WS.org/Vol-274 (CEUR, Idaho, 2007) 27–39.

19. A. DELANEY, G. STAPLETON, J. TAYLOR and S. THOMPSON, 'Fragments of spider diagrams of order and their relative expressiveness', *Proceedings of 6th International Conference on the Theory and Application of Diagrams, Portland, OR*, Lecture Notes in Artificial Intelligence 6170 (Springer, Berlin, 2010) 69–83.

20. A. DELANEY, J. TAYLOR and S. THOMPSON, 'Spider diagrams of order and a hierarchy of star-free regular languages', *Proceedings of 5th International Conference on the Theory and Application of Diagrams, Herrsching, Germany*, Lecture Notes in Artificial Intelligence 5223 (Springer, Berlin, 2008) 172–187.

21. P. EADES, W. LAI, K. MISUE and K. SUGIYAMA, 'Layout adjustment and the mental map', *Vis. Lang. Comput.* 6 (1995) 183–210.

22. M. EIGENHOFER and R. FRANZOSA, 'On the equivalence of topological relations', *Int. J. Geogr. Inf. Syst.* 9 (1995) no. 2, 133–152.

23. L. EULER, 'Lettres a une princesse dallemagne sur divers sujets de physique et de philosophie', *Lett. Soc. Typograph. Berne* 2 (1775) 102–108.

24. A. FISH and J. FLOWER, 'Abstractions of Euler diagrams', *Proceedings of Euler Diagrams 2004,* Brighton, UK, Electronic Notes in Theoretical Computer Science 134 (Elsevier, Amsterdam, 2005) 77–101.

25. A. FISH, J. FLOWER and J. HOWSE, 'The semantics of augmented constraint diagrams', *J. Vis. Lang. Comput.* 16 (2005) 541–573.

26. A. FISH, C. JOHN and J. TAYLOR, 'A normal form for Euler diagrams with shading', *Diagrammatic representation and inference, 5th International Conference, Diagrams 2008,* Herrsching, Germany, September 19–21, 2008, Lecture Notes in Computer Science 5223 (Springer, Berlin, 2008) 206–221.

27. A. FISH, B. KHAZAEI and C. ROAST, 'User comprehension of Euler diagrams', *J. Vis. Lang. Comput.* 22 (2011) no. 5, 340–354.

28. J. FLOWER, J. HOWSE and J. TAYLOR, 'Nesting in Euler diagrams: syntax, semantics and construction', *Softw. Syst. Modell.* 3 (2004) 55–67.

29. J. FLOWER, A. FISH and J. HOWSE, 'Euler diagram generation', *Vis. Lang. Comput.* 19 (2008) no. 6, 675–694.

30. J. GIL, J. HOWSE, S. KENT and J. TAYLOR, 'Projections in Venn–Euler diagrams', *Proc. IEEE Symposium on Visual Languages* (IEEE Computer Society Press, Los Alamitos, CA, 2000) 119–126.

31. J. GIL, J. HOWSE and E. TULCHINSKY, 'Positive semantics of projections', *J. Vis. Lang. Comput.* 13 (2001) no. 2, 197–227.

32. B. GRUNBAUM, 'The construction of Venn diagrams', *College Math. J.* 15 (1984) no. 3, 238–247.

33. C. GURR, 'Effective diagrammatic communication: syntactic, semantic and pragmatic issues', *J. Vis. Lang. Comput.* 10 (1999) no. 4, 317–342.

34. C. GURR and K. TOURLAS, 'Towards the principled design of software engineering diagrams', *Proceedings of 22nd International Conference on Software Engineering* (ACM Press, New York, 2000) 509–518.

35. P. HAMBURGER and R. E. PIPPERT, 'Simple, reducible Venn diagrams on five curves and Hamiltonian cycles', *Geom. Dedicata* 68 (1997) no. 3, 245–262.

36. E. HAMMER, *Logic and visual information* (CSLI Publications, 1995).

37. E. HAMMER and N. DANNER, 'Towards a model theory of Venn diagrams', *J. Philos. Logic* 25 (1996) no. 4, 463–482.

38. E. HAMMER and S. J. SHIN, 'Euler's visual logic', *Hist. Philos. Logic* (1998) 1–29.

39. D. HAREL, 'On visual formalisms', *Diagrammatic reasoning* (eds J. Glasgow, N. H. Narayan and B. Chandrasekaran; MIT Press, Cambridge, MA, 1998) 235–271.

40. D. HAREL and H. A. KAHANA, 'On statecharts with overlapping', *ACM Trans. Softw. Eng. Method* 1 (1992) no. 4, 399–421.

41. M. HEGARTY, 'Diagrams in the mind and in the world: relations between internal and external visualizations', *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, Lecture Notes in Artificial Intelligence 2980 (Springer, New York, 2004) 1–13.

42. J. HOWSE, F. MOLINA, S.-J. SHIN and J. TAYLOR, 'Type-syntax and token-syntax in diagrammatic systems', *Proceedings of 2nd International Conference on Formal Ontology in Information Systems,* Maine, USA (ACM Press, New York, 2001) 174–185.

43. J. HOWSE, F. MOLINA, S.-J. SHIN and J. TAYLOR, 'On diagram tokens and types', *Proceedings of 2nd International Conference on the Theory and Application of Diagrams,* Georgia, USA (Springer, New York, 2002) 146–160. April.

44. J. HOWSE, F. MOLINA, J. TAYLOR, S. KENT and J. GIL, 'Spider diagrams: a diagrammatic reasoning system', *J. Vis. Lang. Comput.* 12 (2001) no. 3, 299–324.

45. J. HOWSE and S. SCHUMAN, 'Precise visual modelling', *J. Softw. Syst. Model.* 4 (2005) 310–325.

46. J. HOWSE, G. STAPLETON, J. FLOWER and J. TAYLOR, 'Corresponding regions in Euler diagrams', *Proceedings of 2nd International Conference on the Theory and Application of Diagrams,* Georgia, USA (Springer, New York, 2002) 76–90.

47. J. HOWSE, G. STAPLETON and J. TAYLOR, 'Spider diagrams', *LMS J. Comput. Math.* 8 (2005) 145–194.

48. J. HOWSE, G. STAPLETON, K. TAYLOR and P. CHAPMAN, 'Visualizing ontologies: a case study', *The semantic web ISWC 2011*, Lecture Notes in Computer Science 7031 (Springer, New York, 2011) 257–272.

49. M. JAMNIK, *Mathematical reasoning with diagrams* (CSLI Press, Stanford, CA, 2001).

50. M. JAMNIK, A. BUNDY and I. GREEN, 'Automation of diagrammatic reasoning', *Proceedings of the 15th International Joint Conference on Artificial Intelligence* 1 (Morgan Kaufmann Publishers, San Francisco, CA, 1997) 528–533.

51. J. F. SOWA, *Conceptual structures: information processing in mind and machine* (Addison-Wesley, Reading, MA, 1984).

52. C. JOHN, 'Reasoning with projected contours', *Proceedings of 3rd International Conference on the Theory and Application of Diagrams* Lecture Notes in Artificial Intelligence 2980 (Springer, New York, 2004) 147–150.

53. C. JOHN, 'Projected contours in Euler diagrams', *Euler diagrams 2004*, Electronic Notes in Theoretical Computer Science 134 (Elsevier, Amsterdam, 2005) 103–126.

54. C. JOHN, 'Measuring and reducing clutter in spider diagrams with projections', PhD Thesis, University of Brighton, 2006.

55. C. JOHN, A. FISH, J. HOWSE and J. TAYLOR, 'Exploring the notion of clutter in Euler diagrams', *Proceedings of 4th International Conference on the Theory and Application of Diagrams*, Lecture Notes in Artificial Intelligence 4045 (Springer, New York, 2006) 267–282.

56. D. S. JOHNSON and H. O. POLLAK, 'Hypergraph planarity and the complexity of drawing Venn diagrams', *J. Graph Theory* 11 (1987) no. 3, 309–325.

57. M. KARNAUGH, 'The map method for synthesis of combinational logic circuits', *Trans. Amer. Inst. Electr. Eng.* 72 (1953) no. 5, 593–599.

58. S. KENT, 'Constraint diagrams: Visualizing invariants in object oriented modelling', *Proceedings of OOPSLA97* (ACM Press, New York, 1997) 327–341.

59. H. KESTLER, A. MULLER, T. GRESS and M. BUCHHOLZ, 'Generalized Venn diagrams: a new method for visualizing complex genetic set relations', *J. Bioinformatics* 21 (2005) no. 8, 1592–1595.

60. H. A. KESTLER, A. MÜLLER, J. M. KRAUS, M. BUCHHOLZ, T. M. GRESS, H. LIU, D. W. KANE, B. R. ZEEBERG and J. WEINSTEIN, 'Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays', *BMC Bioinformatics* 9 (2008) 67.

61. J. LARKIN and H. SIMON, 'Why a diagram is (sometimes) worth ten thousand words', *J. Cognitive Sci.* 11 (1987) 65–99.

62. C. PAPADIMITRIOU, M. GRIGNI and D. PAPADIAS, 'Topological inference', *14th Conference on Artificial Intelligence* (Morgan Kaufmann, San Francisco, CA, 1995) 901–906.

63. K. MAMAKANI, W. MYRVOLD and F. RUSKEY, 'Generating simple convex Venn diagrams', *J. Discrete Algorithms* 16 (2012) 270–286.

64. T. MORE, 'On the construction of Venn diagrams', *J. Symbolic Logic* 23 (1959) 303–304.

65. N. H. RICHE and T. DWYER, 'Untangling Euler diagrams', *IEEE Trans. Vis. Comput. Graphics* 16 (2010) no. 6, 1090–1099.

66. F. RUSKEY and M. WESTON, 'A survey of Venn diagrams', *Electron. J. Combin.* (1997) updated 2001, 2005. www.combinatorics.org/Surveys/ds5/VennEJC.html.

67. A. SHIMOJIMA, 'Inferential and expressive capacities of graphical representations: survey and some generalizations', *Diagrammatic Representation and Inference: Proceedings of Diagrams 2004*, Lecture Notes in Computer Science 2980 (Springer, New York, 2004) 18–21.

68. S.-J. SHIN, *The logical status of diagrams* (Cambridge University Press, Cambridge, 1994).

69. P. SIMONETTO, D. AUBER and D. ARCHAMBAULT, 'Fully automatic visualisation of overlapping sets', *Comput. Graph. Forum* 28 (2009) 967–974.

70. G. STAPLETON and A. DELANEY, 'Evaluating and generalizing constraint diagrams', *J. Vis. Lang. Comput.* 19 (2008) 499–521.

71. G. STAPLETON, J. HOWSE, P. CHAPMAN, A. DELANEY, J. BURTON and I. OLIVER, 'Formalizing concept diagrams', *19th International Conference on Distributed Multimedia Systems, Visual Languages and Computing* (Knowledge Systems Institute, Skokie, IL, 2013) 182–187.

72. G. STAPLETON, J. HOWSE and P. RODGERS, 'A graph theoretic approach to general Euler diagram drawing', *Theoret. Comput. Sci.* 411 (2010) no. 1, 91–112.

73. G. STAPLETON and J. MASTHOFF, 'Incorporating negation into visual logics: a case study using Euler diagrams', *Visual Languages and Computing 2007* (Knowledge Systems Institute, Skokie, IL, 2007) 187–194.

74. G. STAPLETON, J. MASTHOFF, J. FLOWER, A. FISH and J. SOUTHERN, 'Automated theorem proving in Euler diagrams systems', *J. Automat. Reason.* 39 (2007) 431–470.

75. G. STAPLETON, S. THOMPSON, J. HOWSE and J. TAYLOR, 'The expressiveness of spider diagrams', *J. Logic Comput.* 14 (2004) no. 6, 857–880.

76. N. SWOBODA and G. ALLWEIN, 'Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference', *J. Softw. Syst. Model.* 3 (2004) no. 2, 136–149.

77. J. THIÈVRE, M. VIAUD and A. VERROUST-BLONDET, 'Using Euler diagrams in traditional library environments', *Euler Diagrams 2004*, Electronic Notes on Theoretical Computer Science 134 (Elsevier, Amsterdam, 2005) 189–202.

78. J. VENN, 'On the diagrammatic and mechanical representation of propositions and reasonings', *Philos. Mag. Ser. 5* 10 (1880) 1–18.

79. L. WILKINSON, 'Exact and approximate area-proportional circular Venn and Euler diagrams', *IEEE Trans. Vis. Comput. Graphics* 18 (2012) no. 2, 321–331.

Andrew Fish   *John Taylor*
School of Computing   *School of Computing*
* Engineering and Mathematics*   * Engineering and Mathematics*
University of Brighton   *University of Brighton*
Brighton   *Brighton*
United Kingdom   *United Kingdom*

Andrew.Fish@brighton.ac.uk   John.Taylor@brighton.ac.uk