London Mathematical Society ISSN 1461–1570

SPIDER DIAGRAMS

JOHN HOWSE, GEM STAPLETON AND JOHN TAYLOR

Abstract

The use of diagrams in mathematics has traditionally been restricted to guiding intuition and communication. With rare exceptions such as Peirce's α and β systems, purely diagrammatic formal reasoning has not been in the mathematician's or logician's toolkit. This paper develops a purely diagrammatic reasoning system of 'spider diagrams' that builds on Euler, Venn and Peirce diagrams. The system is known to be expressively equivalent to first-order monadic logic with equality. Two levels of diagrammatic syntax have been developed: an 'abstract' syntax that captures the structure of diagrams, and a 'concrete' syntax that captures topological properties of drawn diagrams. A number of simple diagrammatic transformation rules are given, and the resulting reasoning system is shown to be sound and complete.

1. *Introduction*

The value of diagrams is widely acknowledged in information representation and informal reasoning. In mathematical and logical reasoning, however, diagrams have traditionally been allowed only as a heuristic tool. Although proofs may use diagrams to aid comprehension and communication, they have only been permitted if the underlying argument is expressible in some (formal) text-based language. In [37], Sun-Joo Shin gives a cogent summary of this long-standing 'prejudice' against diagrammatic reasoning, before developing two (sound and complete) reasoning systems of Venn diagrams. In this paper, we develop a purely diagrammatic formal reasoning system, equivalent in expressive power to monadic first-order logic with equality.

Circles or contours (simple closed curves) have been in use for the representation of classical syllogisms at least as far back as the Middle Ages [30]. Euler introduced the notation that we now call *Euler diagrams* [2] to illustrate relations between sets. This notation uses the topological properties of enclosure, exclusion and intersection to represent the set-theoretic notions of subset, disjointness, and intersection, respectively. For example, the Euler diagram in Figure 1 denotes that A and B are disjoint, and that $C \subseteq B$.

John Venn used contours to represent logical propositions [43]. In a Venn diagram, each pair of contours intersects. Moreover, for each non-empty subset of the contours, the intersection of the interiors of the contours in the subset is a non-empty connected region of the diagram. Shading is used to indicate that a particular region of the diagram denotes the empty set. Figure 2 shows a Venn diagram capturing the same information as the Euler diagram in Figure 1.

This research was partially supported by EPSRC grant number GR /R63516/01. The second author was supported by an EPSRC studentship 01800274

Received 2 November 2004, revised 16 May 2005; *published* 6 September 2005.

2000 Mathematics Subject Classification 03B22

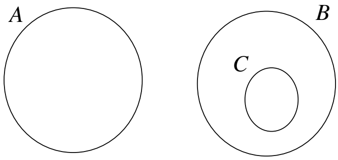

Figure 1: An Euler diagram.

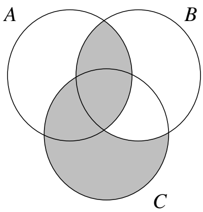

Figure 2: A Venn diagram.

Charles Peirce augmented Venn diagrams by adding ‘X-sequences’ as a means for denoting elements [34]. An X-sequence connecting a number of ‘minimal regions’ of a Venn diagram indicates that their union is not empty. Formal semantics and sound and complete inference rules have been developed for Venn–Peirce diagrams by Shin [37], and for Euler circles by Hammer [19].

Spider diagrams [17] are a natural extension of Venn–Peirce and Euler diagrams; they are based on Euler diagrams, so the topological properties of the diagrams are important, but they also contain *spiders*, a generalization of Peirce’s X-sequences, and shading. The spider diagram in Figure 3 denotes that $C \subseteq B$, there are exactly two elements in $A - B$, and there is at least one element in $B - A$.

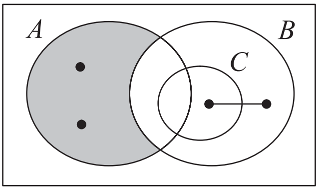

Figure 3: A spider diagram.

Spider diagrams emerged from work on constraint diagrams [28], introduced as a visual technique intended to be used in conjunction with the Unified Modeling Language (UML) [33] for object-oriented modelling. The constraint diagram in Figure 4 expresses, among other constraints, an invariant on a model of a car-hire business: *the specification of the car assigned to a reservation must be the same as or better than the specification reserved.*

$$\forall r \in \text{Reservations} \bullet r.\text{assigned.spec} = r.\text{reserved} \vee \; r.\text{assigned.spec} \in r.\text{reserved.better}$$

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

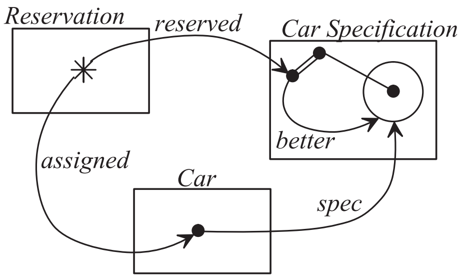

Figure 4: A constraint diagram.

Currently, such constraints can be expressed in UML only by using the Object Constraint Language (OCL) [45], essentially a stylized, textual version of first-order predicate logic.

In this paper we modify and extend the spider diagram systems given by Molina in [31]. Our spider diagrams are based on Euler diagrams, whereas the previous spider diagram systems SD1 and SD2 are based on Venn diagrams [23, 25, 31]. Although not more expressive than SD2, our 'Euler-based' spider diagrams provide a more user-friendly system: Venn diagrams look cluttered when more than three contours are present. A spider diagram system, ESD2, introduced in [31], allows Euler-based diagrams; however, all reasoning, with the exception of one reasoning rule, takes place at the Venn-diagram level. In [41], it is shown that the system introduced in this paper is expressively equivalent to first-order monadic logic with equality.

There is a need to express both disjunctive and conjunctive information, achieved by drawing a collection of diagrams. In all previous spider diagram systems, this information was restricted to a conjunctive normal form; we remove this restriction. This more flexible approach should aid diagrammatic modelling and reasoning.

In Section 2 we give the syntax of spider diagrams. The semantics are defined in Section 3. Reasoning rules for the systems are developed in Section 4. Soundness and completeness results are given in Sections 5 and 6, respectively. The expressiveness of the system is discussed in Section 7. More details can be found in [38].

2. *Spider diagrams: syntax*

In this section we introduce the syntax of spider diagrams. Following [21, 22], we define two layers of spider diagram syntax: an *abstract* or *type* syntax, and a *concrete* or *token* syntax. In order to define a rigorous reasoning system of spider diagrams and to explore its formal properties, it is helpful to have a definition of spider diagrams that is independent of the fine-grained topological properties of diagrams. This is provided by our definition of an abstract spider diagram as a certain many-sorted algebra that captures the structural properties of a diagram. However, the *raison d'être* of our system is precisely that it is *diagrammatic*, and the abstract definition loses this. Thus we also define the notion of a *concrete spider diagram* that formalizes 'drawn' diagrams (on paper or a computer monitor, say) and captures the topological properties of a diagram. Separating the structural from the topological aspects of the spider diagram syntax helps to clarify and formalize the reasoning within the system, and has avoided some of the difficulties faced by Shin in her Venn systems [37], which were noted by Scotto di Luzio [36].

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

We begin by giving an informal description of unitary spider diagrams. Essentially, these are a hybrid of Euler, Venn and Peirce diagrams: roughly speaking, we preserve the topological notions of enclosure and disjointness employed in Euler diagrams, the use of shading employed in Venn diagrams to represent empty sets (although, in spider diagrams, shaded regions do not necessarily represent empty sets), and the use of X-sequences to represent elements, employed in Peirce diagrams.

Informally, a concrete spider diagram is a subset of the plane $\mathbb{R}^2$ containing various syntactic elements. (A more formal description is given in Section 2.2.) A *contour* is (the image of) a simple closed plane curve. Each contour is labelled. All of the contours in a spider diagram are enclosed by a *boundary rectangle* which, formally, is not itself a contour. A *basic region* is the bounded region of the plane enclosed by a contour or the boundary rectangle. A *region* is defined recursively as follows: any basic region is a region; if r_1 and r_2 are regions, then the union, intersection or difference of r_1 and r_2 is a region, provided that these are non-empty. A *zone* is a region having no other region contained within it. Thus a zone is a bounded subset of the complement of the contours and boundary rectangle; we will also impose a well-formedness condition that each zone is a connected component of the complement of the contours and boundary rectangle. Zones may be *shaded* or *unshaded*. A region is shaded if each of its component zones is shaded.

A *spider* is a plane tree with vertices (called *feet*) placed in different zones, and edges (called *legs*) which are straight-line segments. All spiders are contained within the boundary rectangle. A spider *touches* a zone if one of its feet is placed in that zone. It follows that a spider can touch any zone at most once. A spider is said to *inhabit* the region which is the union of the zones that it touches; this region is called the *habitat* of the spider. A (concrete) *unitary spider diagram* comprises a single boundary rectangle, together with a finite collection of contours and spiders. No two contours in a unitary spider diagram can have the same label.

Semantically, the regions of a spider diagram denote sets, and each spider denotes the existence of an element in the set represented by its habitat. Distinct spiders denote distinct elements. Shading a region denotes that it contains no elements other than those indicated by the spiders touching the region; in particular, a shaded region that is not touched by any spider denotes the empty set.

Example 2.1. The diagram in Figure 5 contains three contours, labelled A, B and C, and six zones; for example, the region inside the contour B but outside the contours A and C is a zone. Two of the zones are shaded. The diagram contains two spiders – a single-footed spider whose habitat is the zone inside C and outside A and B, and the 'articulated' spider whose habitat is the basic region inside contour A.

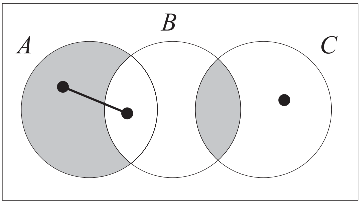

Figure 5: A concrete spider diagram.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

2.1. *Abstract spider diagrams*

To describe the diagrammatic elements abstractly in a concrete diagram, we need to specify its contours and zones, including which zones are shaded, and list the spiders together with their habitats. Each zone is uniquely specified by the contours which enclose it and those which it lies outside. For example, in Figure 5, the right-hand shaded zone lies inside contours B and C, and outside contour A. Thus we may identify this zone by the ordered pair, $(\{B, C\}, \{A\})$, comprising the sets of labels of the contours that the zone lies inside and outside of, respectively. In Figure 5, the six zones are therefore represented as follows:

$$(\varnothing, \{A, B, C\});$$
$$(\{A\}, \{B, C\});$$
$$(\{A, B\}, \{C\});$$
$$(\{B\}, \{A, C\});$$
$$(\{B, C\}, \{A\});$$
$$(\{C\}, \{A, B\}).$$

The set of shaded zones is

$$\big\{(\{A\}, \{B, C\}), (\{B, C\}, \{A\})\big\}$$

and the spiders have habitats

$$\big\{(\{A\}, \{B, C\}), (\{A, B\}, \{C\})\big\} \qquad \text{and} \qquad \big\{(\{C\}, \{A, B\})\big\}.$$

Note that only the *labels* of contours are needed to identify a zone. Thus, for abstract diagrams, we identify contours and labels. It is also convenient for the labels of the contour to be drawn from a fixed, countably infinite set $\mathcal{L}$ of *contour labels*. A zone will be defined to be an ordered pair of disjoint finite sets of contour labels. There appears to be redundancy in using *pairs* of contour labels to define zones since, for a given unitary diagram, we can identify a zone by only those contour labels that contain it. However, later we will consider sets of spider diagrams, each (potentially) with a different contour label, where the single set of containing contour labels is not sufficient to distinguish distinct zones. We denote the set of all finite subsets of a set S by $\mathbb{F}S$.

Definition 2.1. A *zone* with labels in $\mathcal{L}$ is an ordered pair (a, b), where $a, b \subseteq \mathbb{F}\mathcal{L}$ and $a \cap b = \varnothing$. Define $\mathcal{Z}$ to be the set of *zones* on $\mathcal{L}$:

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

If $z = (a, b) \in \mathcal{Z}$, then the set $a = c(z)$ is called the set of contour labels that *contain* z, and $b = e(z)$ is the set of contour labels that *exclude* z. A *region* with labels in $\mathcal{L}$ is a set of zones; $\mathcal{R} = \mathbb{P}\mathcal{Z}$ denotes the set of *regions* on $\mathcal{L}$.

In [23, 25, 31], a diagram contains a set S of spiders, together with a 'habitat function' $\eta : S \longrightarrow R$, that gives the habitat $\eta(s)$ of each spider s as a region of the diagram. Here, we prefer an approach that avoids having a set of spiders with a habitat function but, instead, describes the spiders directly in terms of their habitats. However, describing a spider as a set of zones (its habitat) is not, in general, sufficient to identify a unique spider, since different spiders may have the same habitat. Our approach is to indicate, for a region r, the *number* of spiders whose habitat is r: if there are $n > 0$ spiders in the region r, then we say that the pair (n, r) is a *spider identifier*. Although this is perhaps a less intuitive description than having a set of spiders, one significant advantage is that every concrete unitary spider diagram has a unique abstraction.

DEFINITION 2.2. A *unitary spider diagram* with labels in $\mathcal{L}$ is a tuple $d = \langle L, Z, Z^*, \mathrm{SI}\rangle$ whose components are defined as follows.

1. $L = L(d) \in \mathbb{F}\mathcal{L}$ is a finite set of contour labels.
2. $Z = Z(d) \subseteq \{(a, L - a) : a \subseteq L\}$ is a set of zones ($Z(d) \subseteq \mathcal{Z}$) such that
   (i) for all $l \in L$ there exists $(a, L - a) \in Z$ such that $l \in a$;
   (ii) $(\varnothing, L) \in Z$.

We define $R = R(d) = \mathbb{P}Z - \{\varnothing\}$ to be the set of *regions*.

3. $Z^* = Z^*(d) \subseteq Z$ is the set of *shaded zones*. We define $R^* = R^*(d) = \mathbb{P}Z^* - \{\varnothing\}$ to be the set of *shaded regions*.
4. $\mathrm{SI} = \mathrm{SI}(d) \subset \mathbb{Z}^+ \times R(d)$ is a (finite) set of *spider identifiers* such that

$$\forall (n_1, r_1), (n_2, r_2) \in \mathrm{SI} \bullet r_1 = r_2 \Longrightarrow n_1 = n_2.$$

If $(n, r) \in \mathrm{SI}$, we say that there are n *spiders* whose *habitat* is r.

Additionally, the diagram $\bot = \langle \varnothing, \varnothing, \varnothing, \varnothing\rangle$ is a unitary spider diagram.

In a concrete spider diagram, every contour contains at least one zone; condition 2.2(1) ensures that abstract diagrams preserve this property. Also, any concrete diagram contains the zone inside the boundary rectangle but outside all the contours; this property is ensured at the abstract level by condition 2.2(2). Note that we have also 'lost' the boundary rectangle in Definition 2.2. In a concrete diagram, the boundary rectangle simply represents 'where the diagram stops', and thus is not required in the abstract description.

EXAMPLE 2.2. The concrete diagram in Figure 6 has abstract description $d = \langle L, Z, Z^*, \mathrm{SI}\rangle$ where:

the set of contour labels is $L(d) = \{A, B, C\}$;
the set of zones is $Z(d) = \big\{(\varnothing, \{A, B, C\}), (\{A\}, \{B, C\}), (\{A, B\}, \{C\}), (\{C\}, \{A, B\})\big\}$;
the set of shaded zones is $Z^*(d) = \big\{(\{A\}, \{B, C\})\big\}$;
the set of spider identifiers is

$$\mathrm{SI}(d) = \Big\{\big(2, \big\{(\{A\}, \{B, C\}), (\{A, B\}, \{C\})\big\}\big), \big(1, \big\{(\{C\}, \{A, B\})\big\}\big)\Big\}.$$

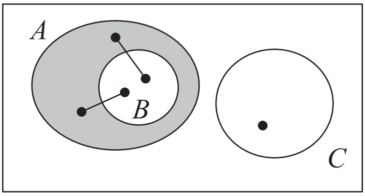

Figure 6: A concrete spider diagram.

The spider diagrams considered in this paper have 'underlying diagrams' that are Euler diagrams rather than Venn diagrams; in other words, if $a \subseteq L(d)$, then the diagram need not contain the zone $(a, L(d) - a)$. For example, the spider diagram in Figure 6 does not contain the zones $(\{B\}, \{A, C\})$, $(\{A, C\}, \{B\})$, $(\{B, C\}, \{A\})$ and $(\{A, B, C\}, \varnothing)$. We say that these zones are 'missing' from the diagram in Figure 6.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

DEFINITION 2.3. Let d be a unitary spider diagram. We define the *Venn zone set* of d, $\mathrm{VZ}(d)$, to be the set of zones in the corresponding Venn diagram with labels $L(d)$:

$$\mathrm{VZ}(d) = \{(a, L(d) - a) : a \subseteq L(d)\}.$$

A diagram d is said to be *in Venn form* if $Z(d) = \mathrm{VZ}(d)$. If $z \in \mathrm{VZ}(d) - Z(d)$, then we say that the zone z is *missing* from d.

Although our formal Definition 2.2 introduces spiders only as 'spider identifiers', we shall need to identify individual spiders and consider the sets of spiders enclosed within, or touching, a particular region. In our definition of a unitary spider diagram, we could have explicitly defined a set of spiders, in a similar way to that given in the definition below, rather than using spider identifiers (which is, essentially, a bag of spiders). However, the definition of spiders using spider identifiers is more concise.

DEFINITION 2.4. Let d be a unitary spider diagram.

1. If $(n, r) \in \mathrm{SI}(d)$, then the region r contains n spiders, which we denote by $s_1(r), s_2(r), \ldots, s_n(r)$. We define $S(d)$ to be the set of all spiders in d:

$$S(d) = \{s_i(r) : (n_r, r) \in \mathrm{SI}(d) \text{ and } 1 \leqslant i \leqslant n_r\}.$$

The habitat mapping $\eta\colon S(d) \longrightarrow R(d)$ is given by $\eta(s_i(r)) = r$, and we say that the spider $s_i(r)$ has *habitat* $\eta(s_i(r))$.

2. Let r be a region of d. The set of complete spiders *inhabiting* r in diagram d is:

$$S(r, d) = \{s \in S(d) : \eta(s) \subseteq r\}.$$

The set of spiders *touching* region r in diagram d is

$$T(r, d) = \{s \in S(d) : \eta(s) \cap r \neq \varnothing\}.$$

For any region r' not in $R(d)$, we define $S(r', d) = \varnothing$ and $T(r', d) = \varnothing$.

The following lemma, whose proof is omitted here, describes the cardinalities of $S(r, d)$ and $T(r, d)$ in terms of spider identifiers in the obvious way.

LEMMA 2.1. *Let d be a unitary diagram, and let $r \in R(d)$. Then*

$$|S(r, d)| = \sum_{\substack{r' \subseteq r \\ (n, r') \in \mathrm{SI}(d)}} n \qquad \textit{and} \qquad |T(r, d)| = \sum_{\substack{r' \cap r \neq \varnothing \\ (n, r') \in \mathrm{SI}(d)}} n. \qquad \square$$

Thus far, we have considered only single (or unitary) spider diagrams, each of which represents a collection of statements about sets and their elements. We shall need to combine diagrams to represent both disjunctive and conjunctive information. Following the approach introduced by Shin [37], previous systems of spider diagrams [23, 24, 25, 31] have only represented expressions in conjunctive normal form. In each of these systems, a *compound diagram* is a set of unitary diagrams, and a *multi-diagram* is a set of compound diagrams. The semantic interpretation of a compound diagram is the disjunction of the expressions represented by each of its unitary diagrams. Similarly, the semantic interpretation of a multi-diagram is the conjunction of the expressions represented by each of its compound diagrams. There are two advantages in defining compound and multi-diagrams as sets of diagrams: repetitions of diagrams are automatically ignored, and the commutativity and idempotency rules for logical disjunction and conjunction are 'built-in'. For example, if

d_1, d_2 and d_3 are unitary spider diagrams, then $\{d_1, d_2\}$, $\{d_2, d_1\}$ and $\{d_1, d_2, d_1\}$ all represent the same compound diagram, and $\{\{d_1, d_2\}, \{d_3\}\}$ and $\{\{d_3\}, \{d_2, d_1, d_2\}\}$ both represent the same multi-diagram.

However, only being able to represent expressions in conjunctive normal form is somewhat restrictive. Here we present a more flexible system that allows diagrams to be combined freely using 'disjunction' and 'conjunction'. Although this allows more freedom when building 'compound diagrams' of various types, there is a penalty to pay. We need to include the idempotency laws in our reasoning rules. Also, we shall have to introduce a slightly more elaborate framework than that used in [**23**, **24**, **25**, **31**, **37**] for representing concrete 'compound' diagrams; we describe this framework in the next section.

Definition 2.5. An *abstract spider diagram* is defined as follows.

1. Any unitary diagram is a spider diagram.
2. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are finite bags of spider diagrams, then $\vee(\mathcal{D}_1 \uplus \mathcal{D}_2)$ (pronounced '$\mathcal{D}_1$ or $\mathcal{D}_2$') is a spider diagram, where $\uplus$ denotes bag union.
3. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are finite bags of spider diagrams, then $\wedge(\mathcal{D}_1 \uplus \mathcal{D}_2)$ (pronounced '$\mathcal{D}_1$ and $\mathcal{D}_2$') is a spider diagram.

If we use this definition of a spider diagram, then associativity and commutativity come for free. We will adopt the convention of writing $D_1 \vee D_2$ to mean $\vee(\{D_1\}\uplus\{D_2\})$. Similarly, we will write $D_1 \wedge D_2$ to mean $\wedge(\{D_1\}\uplus\{D_2\})$. We now define the set of all spider diagrams.

Definition 2.6. We define the set of all spider diagrams, $\mathcal{D}$, inductively as follows:

$\mathcal{D}_0$ is the set of all unitary diagrams with labels in $\mathcal{L}$;

$\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(D_1 \diamond D_2) : D_1, D_2 \in \mathcal{D}_n \wedge \diamond \in \{\vee, \wedge\}\}$;

$\mathcal{D} = \bigcup_{n\in\mathbb{N}} \mathcal{D}_n$.

2.2. *Concrete spider diagrams*

In this section we formalize the notion of a diagram drawn in the plane. The definition needs to capture the topological properties of spider diagrams, and there are various choices to be made concerning what topological features will be allowed. In order to maintain readability and avoid ambiguity, we adopt a fairly restrictive definition. Thus, for example, contours are not allowed 'to be tangential to' or 'to run along' one another – they must cross transversely. Other authors have made different choices as to which topological features are allowed [**9**, **19**, **29**, **44**].

We now give a formal definition of a concrete spider diagram, obtained from [**21**], where the 'type syntax' corresponds (roughly) to our abstract syntax, and the 'token syntax' corresponds to our concrete syntax.

Definition 2.7. A *concrete unitary spider diagram*, $\hat{d}$, with labels in $\mathcal{L}$ is a tuple $\hat{d} = \langle \hat{C}, \hat{\beta}, \hat{Z}, \hat{Z}^*, \hat{L}, \hat{l}, \hat{S}, \hat{\eta} \rangle$, whose components are defined as follows.

1. $\hat{C}$ is a finite set of (the images of) simple closed plane curves called *contours*. The *boundary rectangle* $\hat{\beta}$ is also (the image of) a simple closed curve, usually in the form of a rectangle, but is not a member of $\hat{C}$. For any contour $\hat{c}$ (including $\hat{\beta}$), we denote the interior (bounded) and exterior (unbounded) components of $\mathbb{R}^2 - \hat{c}$ by $\iota(\hat{c})$ and $\varepsilon(\hat{c})$, respectively. Each contour lies in the interior of the boundary rectangle: $\hat{c} \subset \iota(\hat{\beta})$.

The set $\hat{C}$ forms an *Euler diagram* that has the following properties.

(i) Contours meet transversely.

(ii) Each contour intersects with every other contour an even number of times (this can be zero times).

(iii) No two contours have a point in common without crossing at that point.

(iv) Each component of $\mathbb{R}^2 - \bigcup_{\hat{c} \in \hat{C}} \hat{c}$ is the intersection of $\iota(\hat{c})$ for all contours $\hat{c}$ in some (possibly empty) subset X of $\hat{C}$ and $\varepsilon(\hat{c})$ for all contours $\hat{c}$ in the complement of X:

$$\bigcap_{\hat{c} \in X} \iota(\hat{c}) \cap \bigcap_{\hat{c} \in \hat{C} - X} \varepsilon(\hat{c}).$$

2. A *zone* is the intersection of a component of $\mathbb{R}^2 - \bigcup_{\hat{c} \in \hat{C}} \hat{c}$ with $\iota(\hat{\beta})$. A zone may be shaded or unshaded. The set of all zones in $\hat{d}$ is denoted by $\hat{Z}$, and the set of shaded zones is denoted by $\hat{Z}^*$.

A *region* is a non-empty set of zones. We let $\hat{R} = \mathbb{P}\hat{Z} - \{\varnothing\}$ denote the set of regions of d, and $\hat{R}^* = \mathbb{P}\hat{Z}^* - \{\varnothing\}$ denotes the set of shaded regions.

3. $\hat{L} = \hat{L}(\hat{d}) \subseteq \mathcal{L}$ is the set of *contour labels* of $\hat{d}$. The mapping $\hat{l} \colon \hat{C} \longrightarrow \hat{L}$ is a bijection that returns the label of a contour.

4. $\hat{S}$ is a finite set of plane trees, called *spiders*, whose vertices, called *feet*, lie within $\hat{Z}$ and satisfy the following properties.

(i) Each spider has at most one foot in each zone.

(ii) The edges (called *legs*) of each spider are straight-line segments.

(iii) No two spiders have a foot in common.

5. The function $\hat{\eta} \colon \hat{S} \longrightarrow \hat{R}$ returns the *habitat* of each spider:

$$\hat{\eta}(\hat{s}) = \{\hat{z} \in \hat{Z} : \hat{s} \text{ has a foot in } \hat{z}\}.$$

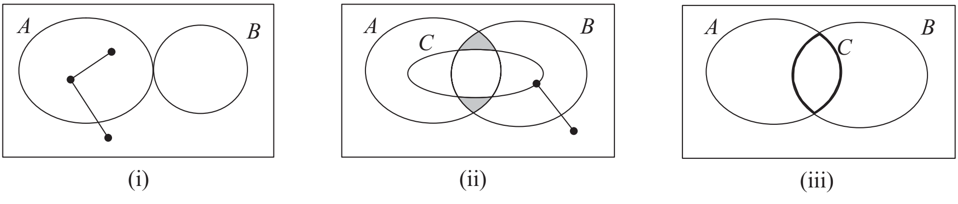

Figure 7: Non-well-formed concrete diagrams.

Example 2.3. Definition 2.7 imposes 'well-formedness' conditions on concrete diagrams. Figure 7 illustrates some of these conditions by presenting diagrams that are not well-formed. In the diagram in Figure 7(i), the contours labelled A and B touch at a point without crossing, and the spider has two feet in the same zone. The diagram in Figure 7(ii) also violates two conditions. Firstly, the shaded 'zone' is not connected: this violates condition 1(iv) of Definition 2.7, since

$$\bigcap_{\hat{c} \in X} \iota(\hat{c}) \cap \bigcap_{\hat{c} \in \hat{C} - X} \varepsilon(\hat{c}),$$

where X is the set of contours labelled A and B, is not a connected component of $\mathbb{R}^2 - \bigcup_{\hat{c} \in \hat{C}} \hat{c}$. Secondly, the spider has a foot on one of the contours (and hence does

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

not lie within $\hat{Z}$). Finally, the diagram in Figure 7(iii) has concurrent contours: C 'runs along' both A and B. This violates condition 1(i) of Definition 2.7.

The problem of 'generating' an abstract description of a concrete diagram is a relatively simple one: given a drawn diagram, one can easily list the labels and zones, and define a set of spider identifiers that achieves what is required. However, creating a concrete diagram from an abstract description is, in general, non-trivial. Flower and Howse begin to address this issue in [9], developing an algorithm to draw concrete Euler diagrams from an abstract description.

We now formalize the connection between abstract and concrete diagrams. Since concrete diagrams contain actual spiders but abstract diagrams contain 'spider identifiers', we define, for each region $\hat{r}$ of a concrete diagram $\hat{d}$, the set of spiders with habitat $\hat{r}$ to be $\hat{S}(\hat{r}) = \{\hat{s} \in \hat{S} : \eta(\hat{s}) = \hat{r}\}$.

Definition 2.8. Let $\hat{d} = \langle \hat{C}, \hat{\beta}, \hat{Z}, \hat{Z}^*, \hat{L}, \hat{l}, \hat{S}, \hat{\eta} \rangle$ be a concrete diagram, and let $d = \langle L, Z, Z^*, \mathrm{SI} \rangle$ be an abstract diagram. Then d is an *abstraction* of $\hat{d}$, denoted $ab(\hat{d}) = d$, if and only if $\hat{L} = L$ and the following two conditions are satisfied.

1. There exists a bijection $\mu_1 : \hat{Z} \longrightarrow Z$ such that, for all $\hat{z} \in \hat{Z}$,
   (i) $\mu_1(\hat{z}) = \left(\{\hat{l}(\hat{c}) : \hat{z} \subseteq \iota(\hat{c}) \wedge \hat{c} \in \hat{C}\}, \{\hat{l}(\hat{c}) : \hat{z} \subseteq \varepsilon(\hat{c}) \wedge \hat{c} \in \hat{C}\}\right)$ and
   (ii) $\mu_1(\hat{z}) \in Z^* \iff \hat{z} \in \hat{Z}^*$.
2. There exists a mapping $\mu_2 : \hat{R} \longrightarrow \mathrm{SI}$ such that, $\mu_2(\hat{r}) = (|\hat{S}(\hat{r})|, \mu_1(\hat{r}))$ where $\mu_1 : \hat{R} \longrightarrow R$ is the natural extension of $\mu_1 : \hat{Z} \longrightarrow Z$.

If d is an abstraction of $\hat{d}$, then we say that $\hat{d}$ is an *instantiation* of d.

Example 2.4. Let d be the abstract diagram $d = \langle L, Z, Z^*, \mathrm{SI} \rangle$, where

1. $L(d) = \{A, B\}$;
2. $Z(d) = \{(\varnothing, \{A, B\}), (\{A\}, \{B\}), (\{A, B\}, \varnothing)\}$;
3. $Z^*(d) = \{(\{A, B\}, \varnothing)\}$; and
4. $\mathrm{SI}(d) = \{(1, \{(\{A, B\}, \varnothing)\}), (1, \{(\{A, B\}, \varnothing), (\varnothing, \{A, B\})\})\}$.

Then d is an abstraction of both d_1 and d_2 in Figure 8; that is, $ab(d_1) = ab(d_2) = d$. Thus the mapping ab is not injective. We can think of d_1 and d_2 as being equivalent.

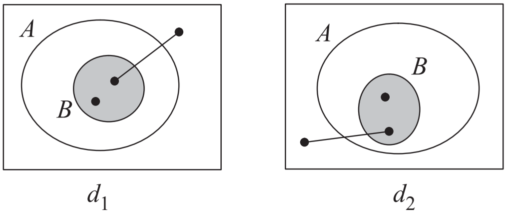

Figure 8: Equivalent concrete diagrams.

Example 2.5. The 'well-formedness' rules defining concrete diagrams are designed to ensure that concrete diagrams are readable without ambiguity. However, a consequence is that there are abstract diagrams that have no concrete instantiation. The following abstract

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

diagram is perhaps the simplest example of an abstract diagram that has no well-formed concrete representation. Let $d = \langle L, Z, Z^*, \mathrm{SI}\rangle$, where

1. $L(d) = \{A, B\}$;
2. $Z(d) = \big\{(\varnothing, \{A, B\}), (\{A, B\}, \varnothing)\big\}$;
3. $Z^*(d) = \varnothing$; and
4. $\mathrm{SI}(d) = \varnothing$.

For a concrete diagram to realize d, the contours labelled A and B would need to coincide.

The task of classifying which abstract spider diagrams have a concrete representation is challenging. If the underlying Euler diagram of an abstract spider diagram has a concrete representation, then the spider diagram also has a concrete representation, and we say that the diagram is *drawable*. In [9], the authors classify which Euler diagrams are drawable, subject to strict well-formedness conditions. In addition to our well-formedness conditions, the authors of [9] do not allow concrete diagrams to contain triple points, illustrated in Figure 9.

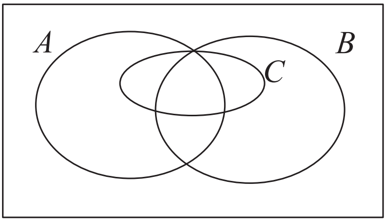

Figure 9: A diagram with a triple point.

Every concrete Euler diagram has a planar *dual graph*. In the dual graph, each zone is represented by a vertex, and two vertices are connected by an edge if and only if the corresponding zones are topologically adjacent in the plane. If two vertices are adjacent in the dual graph, then the symmetric difference of the containing label sets for the corresponding zones contains precisely one element, since contours meet transversely. This element is the label of the contour that borders the two zones. The edges in the dual graph are labelled by this element. The diagram in Figure 10 has dual graph G. Each zone labels the corresponding vertex of G.

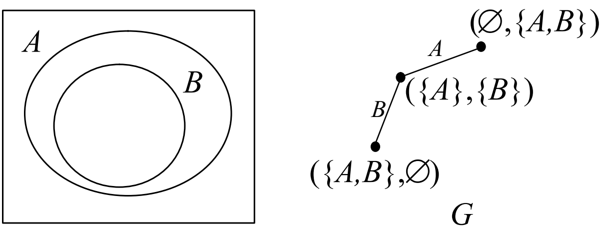

Figure 10: The dual graph.

Consider a graph, G, such that each edge is labelled by an element chosen from some set L and each vertex is labelled by an element of $\mathbb{P}L \times \mathbb{P}L$. Such a graph G satisfies the

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

connectivity conditions if it is connected and if, for all edge labels $l \in L$, the subgraphs $G^{+}(l)$ generated by vertices whose label (X, Y) satisfies $l \in X$ and $G^{-}(l)$ generated by vertices whose label (X, Y) satisfy $l \in Y$ are also connected [9]. The dual graph of a concrete diagram satisfies the connectivity conditions (given the well-formedness rules in [9]).

An abstract diagram d has a *superdual* G whose edges' labels are chosen from $L(d)$, and in which the set of vertices is $Z(d)$, and there is an edge between two vertices (X, Y) and (P, Q) if and only if $|(X-P)\cup(P-X)| = 1$. Each edge is labelled by $(X-P)\cup(P-X)$; see [9]. It follows that an abstract diagram d is not drawable if the superdual of d fails the connectivity conditions. A labelled graph that passes the connectivity conditions is potentially the dual graph of a concrete diagram, but only if it is planar. If it is not planar, then it may be possible to remove edges and obtain a planar graph that passes the connectivity conditions. As an example, edges must be removed from the superdual of the abstract diagram Venn-4 (the Venn diagram on four contours) to produce a planar dual graph of any concrete Venn-4.

Given a diagram d with a superdual that passes the connectivity conditions, the task is to remove edges (if necessary), without causing the connectivity conditions to fail (if possible), until a planar graph is found. This planar graph can then be embedded in the plane without edges crossing, and can be used to construct a concrete representation of d. To summarize, an abstract Euler diagram is drawable (given the well-formedness conditions in [9]) if and only if there exists a planar subgraph, H, of the superdual, G, such that $V(H) = V(G)$ and H passes the connectivity conditions. In [44], a different set of well-formedness conditions is given, which ensures that every Euler diagram with at most eight contours is drawable. Further research is needed to classify all drawable diagrams, given these and other sets of well-formedness conditions.

In order to represent 'compound' concrete diagrams, we need a visual framework for connecting unitary concrete diagrams. In [12], a framework is introduced for combining logic-based notations, both diagrammatic and textual, using various visual 'templates'. In this paper, we use the *box template* to define compound diagrams. This template, illustrated in Figure 11, contains a bounding box containing two or more inner boxes into which diagrams may be 'plugged', and a label, which will be either $\vee$ or $\wedge$, to denote whether the diagrams are to be taken in disjunction or conjunction. A box template with n inner boxes is called an *n-ary box*. (Of course, an n-ary box gives a well-defined diagrammatic representation because conjunction and disjunction are both associative and commutative.) We may also nest templates so that the inner box of one template may contain an n-ary box.

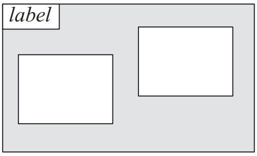

Figure 11: The box template.

EXAMPLE 2.6. Figure 12 shows two concrete compound diagrams. Figure 12(i) is an instantiation of $d_1 \vee d_2$, and Figure 12(ii) represents $(d_1 \wedge d_2) \vee (d_3 \wedge d_4)$.

We are now in a position to define general concrete spider diagrams.

DEFINITION 2.9. A *concrete spider diagram* is defined recursively as follows. Any unitary concrete diagram is a concrete spider diagram. Let D be an n-ary box. If each inner rectangle

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

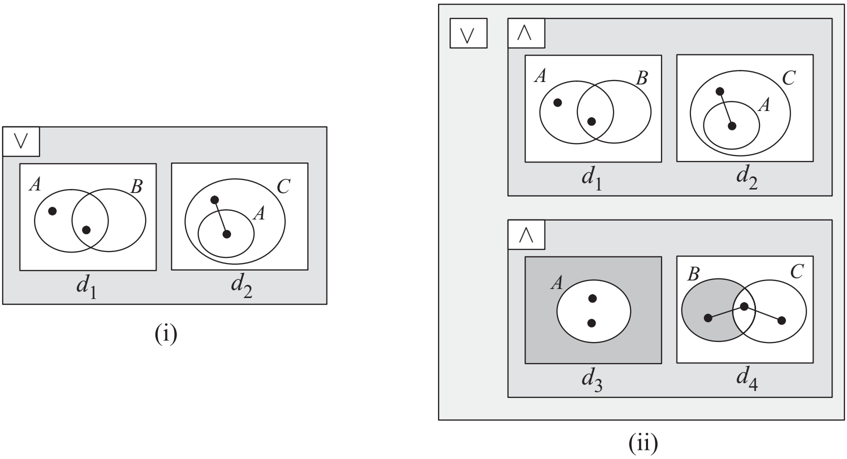

Figure 12: Compound concrete diagrams.

of D is a concrete spider diagram, then D is a concrete spider diagram *based on* an n-ary box.

The mapping between concrete and abstract unitary diagrams given in Definition 2.8 extends to compound diagrams in the natural way.

Definition 2.10. Let $\hat{D}$ be a concrete diagram based on an n-ary box (where $n \geqslant 2$) labelled with $\diamond$ (where $\diamond$ is $\wedge$ or $\vee$), and suppose that the inner rectangles are concrete diagrams $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n$. Let $D = D_1 \diamond D_2 \diamond \ldots \diamond D_n$ be an abstract diagram. If $ab(\hat{D}_i) = D_i$ for $i = 1, \ldots, n$, then $ab(\hat{D}) = D$ and we say that D is an *abstraction* of $\hat{D}$. If D is an abstraction of $\hat{D}$, we say that $\hat{D}$ is an *instantiation* of D.

3. *Semantics*

In this section we formalize the semantics of spider diagrams. The regions in a spider diagram represent sets, and the number of elements in the set represented by a region is greater than or equal to the number of spiders in that region. The number of elements in the set represented by a shaded region is less than or equal to the number of spiders touching that region. This allows us to place lower and, in the case of shaded regions, upper bounds on the cardinalities of the sets we are representing. Missing zones represent the empty set.

Definition 3.1.

1. A *set-assignment to contour labels* is a pair $m = (\mathbf{U}, \Psi)$, where $\mathbf{U}$ is a set and $\Psi : \mathcal{L} \longrightarrow \mathbb{P}\mathbf{U}$ is a function that maps contour labels to subsets of $\mathbf{U}$.

2. We extend Ψ to a *set-assignment to zones*, $\Psi : \mathcal{Z} \longrightarrow \mathbb{P}\mathbf{U}$. The set denoted by a zone, $z = (a, b)$, is defined to be the intersection of the sets denoted by the contour labels in a and the intersection of the complements of the sets denoted by the contour labels b:

$$\Psi(a, b) = \bigcap_{l \in a} \Psi(l) \cap \bigcap_{l \in b} \overline{\Psi(l)},$$

where $\overline{\Psi(l)} = \mathbf{U} - \Psi(l)$. We also define $\bigcap_{l \in \varnothing} \Psi(l) = \mathbf{U} = \bigcap_{l \in \varnothing} \overline{\Psi(l)}$.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

3. Finally, we extend Ψ to a *set-assignment to regions*, $\Psi\colon \mathcal{R} \longrightarrow \mathbb{P}\mathbf{U}$. The set denoted by a region, r, is the union of the sets denoted by the zones that r contains:

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

We also define $\Psi(\varnothing) = \bigcup_{z\in\varnothing} \Psi(z) = \varnothing$.

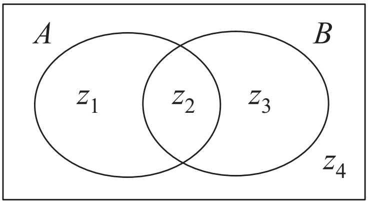

Figure 13: A Venn-2 diagram.

Example 3.1. Taking the universe to be the set of natural numbers, $\mathbb{N}$, a particular set-assignment is the pair $m = (\mathbb{N}, \Psi)$, where Ψ is the mapping $\Psi\colon \mathcal{L} \longrightarrow \mathbb{P}\mathbb{N}$ defined by

$$\Psi(A) = \{1, 2, 3\}, \qquad \Psi(B) = \{3, 4, 10\}$$

and for all $l \in \mathcal{L} - \{A, B\}$, $\Psi(l) = \varnothing$ is a set-assignment to contour labels. In Figure 13, the zone $z_1 = (\{A\}, \{B\})$ represents the set

$$\Psi(z_1) = \Psi(A) \cap \overline{\Psi(B)} = \{1, 2\}.$$

The region $\{z_1, z_2\}$ represents the set

$$\Psi(z_1) \cup \Psi(z_2) = \{1, 2\} \cup \{3\} = \{1, 2, 3\},$$

and so forth.

Note that the mapping Ψ (on regions) is well-behaved with respect to intersection, union and difference. For example, $\Psi(\{z_1\} \cup \{z_2\}) = \Psi(\{z_1\}) \cup \Psi(\{z_2\})$, $\Psi(\{z_1, z_2\} \cap \{z_1\}) = \Psi(\{z_1, z_2\}) \cap \Psi(\{z_1\})$ and $\Psi(\{z_1, z_2\} - \{z_1\}) = \Psi(\{z_1, z_2\}) - \Psi(\{z_1\})$.

Definition 3.1 introduces three functions, each denoted Ψ. Molina [31] showed that this overloading is well behaved, by establishing the following result. (Molina's result was for his SD2 diagrams, but the proof extends to spider diagrams.)

Lemma 3.1. *Let* $m = (\mathbf{U}, \Psi)$ *be a set-assignment to regions.*

1. *If* z_1 *and* z_2 *are distinct zones of a unitary diagram* d *then* $\Psi(z_1) \cap \Psi(z_2) = \varnothing$.
2. *Let* r_1 *and* r_2 *be regions of unitary diagram* d. *Then*
   (i) $\Psi(r_1 \cup r_2) = \Psi(r_1) \cup \Psi(r_2)$;
   (ii) $\Psi(r_1 \cap r_2) = \Psi(r_1) \cap \Psi(r_2)$;
   (iii) $\Psi(r_1 - r_2) = \Psi(r_1) - \Psi(r_2)$;
   (iv) *if* $r_1 \subseteq r_2$, *then* $\Psi(r_1) \subseteq \Psi(r_2)$. □

As in previous work on spider diagram systems [31], the semantics of spider diagrams are captured by a 'semantics predicate'. Our semantics predicate combines the semantics predicate for unitary SD2 diagrams (giving the interpretation of spiders and shading) with the plane tiling condition from ESD2 (which gives the interpretation of the underlying Euler diagram).

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

DEFINITION 3.2. Let D be a diagram, and let $m = (\mathbf{U}, \Psi)$ be a set-assignment to regions. We define the *semantics predicate*, denoted $P_D(m)$, of D. If $D = d$ ($\neq \perp$) is a unitary diagram, then $P_d(m)$ is the conjunction of the following three conditions.

1. *Distinct spiders condition.* The cardinality of the set denoted by a region r of a unitary diagram d is greater than or equal to the number of complete spiders in r:

$$\bigwedge_{r \in R(d)} |\Psi(r)| \geqslant |S(r, d)|.$$

2. *Shading condition.* The cardinality of the set denoted by a shaded region r of a unitary diagram d is less than or equal to the number of spiders touching r:

$$\bigwedge_{r \in R^*(d)} |\Psi(r)| \leqslant |T(r, d)|.$$

3. *Plane tiling condition.* All elements fall within sets denoted by the zones of d:

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

If $D = \perp$, then $P_D(m) = \perp$. If $D = D_1 \vee D_2$, then $P_{D_1}(m) \vee P_{D_2}(m)$. If $D = D_1 \wedge D_2$, then $P_{D_1}(m) \wedge P_{D_2}(m)$.

The plane tiling condition asserts that the union of the sets representing those zones present in a unitary diagram is the universal set. An alternative condition is that each of the sets represented by those zones missing from the diagram is empty. Recall that, for any unitary diagram d, a *missing zone* is an element of $\mathrm{VZ}(d) - Z(d)$. The following theorem, given in [**11**], formalizes this alternative semantic condition.

THEOREM 3.1. *Let* $(\mathbf{U}, \Psi)$ *be a set-assignment to regions, and let d be a unitary diagram. The plane tiling condition for d is equivalent to the following* missing zones condition.

$$\bigcup_{z \in \mathrm{VZ}(d) - Z(d)} \Psi(z) = \varnothing. \qquad \square$$

EXAMPLE 3.2. The pair $m = (\mathbb{N}, \Psi)$ where Ψ is the mapping $\Psi \colon \mathcal{L} \longrightarrow \mathbb{PN}$ defined by

$$\Psi(A) = \{1\}, \qquad \Psi(B) = \{1, 2, 3\}, \qquad \Psi(C) = \{3\}$$

and for all $l \in \mathcal{L} - \{A, B, C\}$, $\Psi(l) = \varnothing$ is a set-assignment to contour labels. In Figure 14, the zone z_1 represents the set $\{2\}$, since $\Psi(z_1) = \Psi(B) \cap \overline{\Psi(A)} \cap \overline{\Psi(C)}$. If the semantics predicate for d_1 is to be satisfied, we must have $|\Psi(z_1)| \leqslant 0$. This is false; hence d_1 fails the shading condition, and so $P_{d_1}(m)$ is false. Note that d_1 also fails the plane tiling condition since

$$\bigcup_{z \in Z(d_1)} \Psi(z) = \mathbb{N} - \{1\}.$$

Now consider the diagram d_2. With the same mapping $\Psi \colon \mathcal{L} \longrightarrow \mathbb{PN}$, the zone z_2 represents the set $\{1\}$. Since the zone z_2 contains and is touched by a single spider, it satisfies the distinct spiders condition and the shading condition. It is straightforward to verify that each of the other regions satisfies the distinct spiders condition, and since $\{z_2\}$ is the only shaded region it follows that the distinct spiders condition and the shading condition hold for d_2. To check the plane tiling condition, note that

$$\Psi(z_3) = \{2, 3\} \qquad \text{and} \qquad \Psi(z_4) = \mathbb{N} \cap (\mathbb{N} - \{1, 2, 3\}) \cap (\mathbb{N} - \{1\}) = \mathbb{N} - \{1, 2, 3\}.$$

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

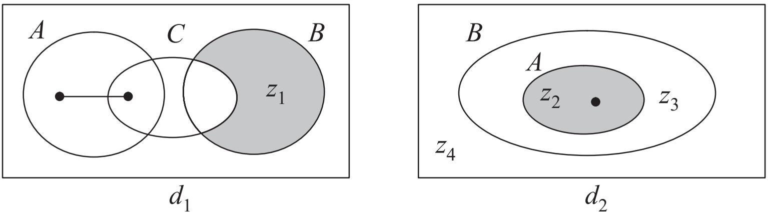

Figure 14: Two spider diagrams.

Thus

$$\Psi(\{z_2, z_3, z_4\}) = \Psi(\{z_2\}) \cup \Psi(\{z_3\}) \cup \Psi(\{z_4\}) = \{1\} \cup \{2, 3\} \cup (\mathbb{N} - \{1, 2, 3\}) = \mathbb{N},$$

and so the plane tiling condition holds. Hence $P_{d_2}(m)$ is true.

There is, in fact, another condition, the *containment condition*, which is equivalent to the plane tiling condition as shown by the following theorem.

THEOREM 3.2. *The plane tiling condition is equivalent to the following* containment condition, *which asserts that the set represented by each basic region is the same as that represented by its containing contour: for all $l \in L(d)$,*

$$\bigcup_{l \in c(z)} \Psi(z) = \Psi(l),$$

where $c(z)$ is the set of contour labels that contain z. □

DEFINITION 3.3. Let D be a spider diagram, and let $m = (\mathbf{U}, \Psi)$ be a set-assignment to regions. We say m is a *model* for D, denoted by $m \models D$, if and only if $P_D(m)$ is true. A spider diagram D is *satisfiable* if and only if it has a model.

Every unitary spider diagram ($\neq \bot$) is satisfiable. Given a non-false unitary diagram d, we follow the approach adopted by Molina [31] to construct a model for d, as follows. Take the universal set to be the set of spiders in d, $\mathbf{U} = S(d)$. For each spider $s \in S(d)$, choose a zone $f(s)$ in the habitat of s; this defines a 'choice function' $f : S(d) \longrightarrow Z(d)$ such that $f(s) \in \eta(s)$. Given the choice function, we can define a set-assignment to contour labels $\Psi : \mathcal{L} \longrightarrow \mathbb{P}S(d)$ by

$$\Psi(l) = \begin{cases} \{s \in S(d) : l \in c(f(s))\} & \text{if } l \in L(d), \\ \varnothing & \text{otherwise.} \end{cases}$$

The extension of Ψ to zones and regions satisfies the following conditions.

1. For any zone $z \in Z(d)$, $\Psi(z) = \{s \in S(d) : f(s) = z\}$.
2. For any region $r \in R(d)$, $\Psi(r) = \{s \in S(d) : f(s) \in r\}$.

It can be shown that the set-assignment $(S(d), \Psi)$ defined above is a model for d. (The proof, which we omit, is similar to that given in [31] for the SD2 system.)

THEOREM 3.3. *Every unitary spider diagram ($\neq \bot$) has a model.* □

DEFINITION 3.4. Let D_1 and D_2 be two spider diagrams. The diagram D_2 is a *logical consequence* of D_1, denoted $D_1 \vDash D_2$, if and only if every model for D_1 is also a model for D_2; that is,

$$D_1 \vDash D_2 \iff (\forall m \bullet (m \models D_1 \implies m \models D_2)).$$

4. *Diagrammatic reasoning*

In this section we introduce the reasoning rules for spider diagrams. Each rule transforms one spider diagram syntactically into another. The rules are defined on abstract diagrams, although we visualize their effect on concrete diagrams. For each of the main diagrammatic rules, we give an informal description and an illustration using concrete diagrams, followed by a formal definition using the abstract syntax.

4.1. *Rules of transformation of diagrams*

Many of the inference rules given here are generalizations of those given in [31] for the SD2 system. For each of the rules, we give an informal description, as well as the formal definition using the abstract syntax. However, because we are no longer forcing diagrams to be in conjunctive normal form, there are new rules analogous to rules in logic (Rules 6 – 13). The diagrammatic rules (Rules 1 – 5) given in this section preserve semantic information. Although this is not a requirement, information-preserving rules are useful when using tableaux [8]. Indeed, there are only two rules in our system that weaken information, both of which have analogies in logic: from D_1 we may deduce $D_1 \vee D_2$, and from $\perp$ we may deduce any diagram.

The first diagrammatic rule that we give allows contours to be introduced into a diagram, provided that no new semantic information is introduced. For example, in Figure 15, introducing contour B into d_1 to produce d_2 is invalid since d_2 includes the semantic information $A \cap B = \emptyset$, which is not represented by d_1. Similarly, introducing contour B into d_1 to produce d_3 is invalid since d_3 denotes $A - B \neq \emptyset$, which is stronger than $A \neq \emptyset$, which is denoted by d_1.

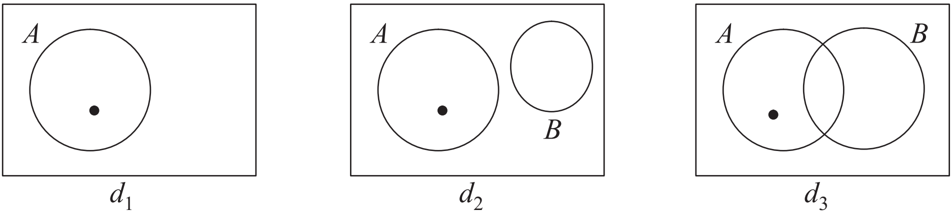

Figure 15: Invalid contour introduction.

Rule 1 (Introduction of a contour label). Let d $(\neq \perp)$ be a unitary diagram, and let d' be the diagram obtained from d after introducing a new contour label satisfying the following conditions:

1. Each zone splits into two zones.
2. Each shaded zone splits into two shaded zones.
3. Each foot of a spider is replaced with a connected pair of feet, one foot in each new zone.

Then we may replace d with d', and vice versa.

Example 4.1. In Figure 16, the diagrams d and d' are semantically equivalent. Each zone in d splits into two zones in d'. The spiders' feet in diagram d bifurcate to become two feet in diagram d'.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

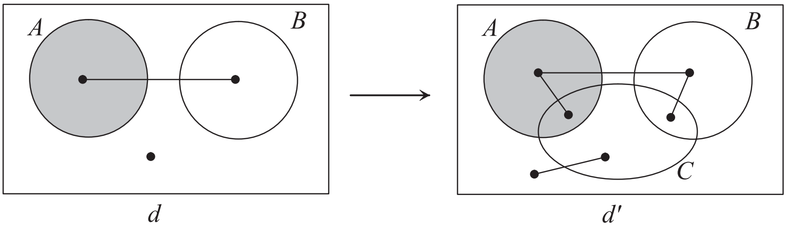

Figure 16: An application of Rule 1, introduction of contour label.

FORMAL DESCRIPTION. Let d ($\neq \perp$) be a unitary diagram, and let d' be a unitary diagram satisfying the following conditions.

1. $L(d') = L(d) \cup \{l^*\}$, where $l^* \notin L(d)$.
2. There exists a surjection $h\colon Z(d') \longrightarrow Z(d)$ defined by

$$h\big(a, L(d') - a\big) = \big(a - \{l^*\}, L(d') - (a \cup \{l^*\})\big),$$

which extends to the function $h\colon R(d') \longrightarrow R(d)$, where

$$h(r) = \bigcup_{z \in r} \{h(z)\} = \{h(z) : z \in r\}.$$

3. (i) The mapping $h : Z(d') \longrightarrow Z(d)$ is two-to-one: for every zone $z \in Z(d)$ there exist distinct zones $z_1, z_2 \in Z(d')$ such that $h(z_1) = z = h(z_2)$.
   (ii) The mapping $h : Z(d') \longrightarrow Z(d)$ preserves shading:

$$z \in Z^*(d') \Longleftrightarrow h(z) \in Z^*(d).$$

4. There exists a bijection $\sigma\colon S(d) \longrightarrow S(d')$ such that, for all $s \in S(d)$, $\eta(\sigma(s)) = \{z \in Z(d') : h(z) \in \eta(s)\}$.

Then d can be replaced by d' and vice versa.

The same semantic information can often be represented by syntactically different spider diagrams. As a simple example, each of the diagrams in Figure 17 represents $A \subseteq B$. In order to obtain a complete system, we need to be able to transform between the diagrams d and d' in Figure 17. Similarly, amongst the abstract diagrams representing $A = B$ are the following:

$$d_1 = \big\langle L, Z(d_1) = \big\{(\varnothing, \{A, B\}), (\{A, B\}, \varnothing)\big\}, Z^*(d_1) = \varnothing, \mathrm{SI}\big\rangle,$$
$$d_2 = \big\langle L, Z(d_2) = \big\{(\varnothing, \{A, B\}), (\{B\}, \{A\}), (\{A, B\}, \varnothing)\big\}, Z^*(d_2) = \big\{(\{B\}, \{A\})\big\}, \mathrm{SI}\big\rangle,$$
$$d_3 = \big\langle L, Z(d_3) = \big\{(\varnothing, \{A, B\}), (\{A\}, \{B\}), (\{B\}, \{A\}), (\{A, B\}, \varnothing)\big\}, Z^*(d_3) = \big\{(\{A\}, \{B\}), (\{B\}, \{A\})\big\}, \mathrm{SI}\big\rangle,$$

where $L = \{A, B\}$ and $\mathrm{SI} = \varnothing$.

Of these, d_1 does not have a concrete representation (it would require the contours labelled A and B to 'run along' one another). The diagrams d_2 and d_3 are represented by shading the zone $(\{B\}, \{A\})$ in the diagrams d and d' in Figure 17 respectively. Again, we need to be able to transform each of the diagrams d_1, d_2 and d_3 into each of the other two.

The next rule allows us to introduce a zone that is not already in the diagram, provided that it is shaded, or remove a shaded zone that is not part of the habitat of any spider

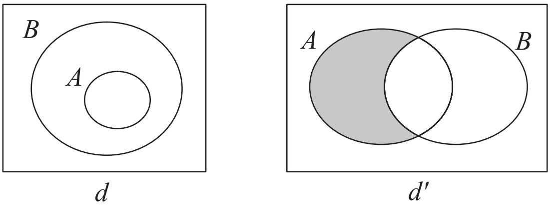

Figure 17: Two representations of $A \subseteq B$.

(provided that it is not the only zone contained within some contour). This will allow the diagram d in Figure 17 to be replaced by d'. Similarly, it allows us to transform between the abstract diagrams d_1, d_2 and d_3 above.

RULE 2 (INTRODUCTION OF A SHADED ZONE). Let d be a unitary diagram and let $z = (a, L(d) - a)$ (where $a \subseteq L(d)$) be a zone that is missing from d (that is, $z \notin Z(d)$). Let d' be a copy of d, except that d' contains z as an additional, shaded zone. Then d may be replaced by d', and vice versa.

This rule is similar to the *rule of weakening* given by Hammer in [**19**]. Hammer's rule loses semantic information since his system does not include shading. Here we are able to preserve all the semantic information. By the plane tiling condition, all elements in the universal set must occur in sets represented by zones in the diagram. It follows that missing zones represent the empty set – thus introducing a shaded missing zone does not alter the interpretation of the diagram.

FORMAL DESCRIPTION. Let d ($\neq \perp$) be a unitary diagram that is not in Venn form, and let $z^* \in \text{VZ}(d) - Z(d)$. Let d' be a unitary diagram such that

1. $L(d) = L(d')$;
2. $Z(d) \cup \{z^*\} = Z(d')$;
3. $Z^*(d) \cup \{z^*\} = Z^*(d')$;
4. $\text{SI}(d) = \text{SI}(d')$.

Then d can be replaced by d', and vice versa.

We should note that Rule 2 really only operates at the abstract level. If d' is formed from d by adding a shaded zone, according to the rule, then it is possible for both, exactly one, or neither, of the diagrams to have a well-formed concrete instantiation. This is illustrated in Figure 18. The diagram d_1 is formed from d by adding the shaded zone $(\{C\}, \{A, B\})$ and d_2 is formed from d_1 by adding a further shaded zone $(\{A, C\}, \{B\})$. The diagram d_1 does not have a well-formed concrete instantiation – the representation in Figure 18 is not a well-formed concrete diagram. However, adding the additional shaded zone to form d_2 produces a diagram that does have a well-formed concrete instantiation.

An *articulated spider* is one that has more than one foot; its habitat is a non-trivial union of regions. Semantically, an articulated spider denotes the existence of an element in the set represented by its habitat which is the disjoint union of the sets represented by each of the zones in the habitat. Thus an articulated spider represents disjunctive information, and we can reflect this at the syntactic level by replacing a diagram containing an articulated spider with a disjunction of 'simpler' diagrams.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

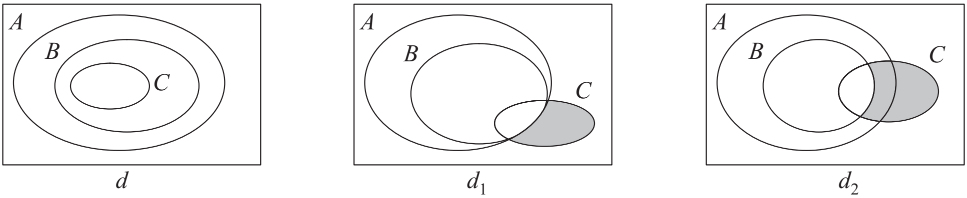

Figure 18: Adding shaded zones to concrete diagrams.

Rule 3 (Splitting spiders). Let d be a unitary diagram with a spider s^* touching every zone of two disjoint regions r_1 and r_2. Let d_1 and d_2 be unitary diagrams that are copies of d except that s^* is replaced in d_1 by a spider whose habitat is region r_1, and s^* is replaced in d_2 by a spider whose habitat is region r_2. Then d can be replaced by the diagram $d_1 \vee d_2$. The rule is reversible; that is, $d_1 \vee d_2$ can be replaced by d.

Example 4.2. Figure 19 illustrates an application of this rule. The spider s^* in d splits into two spiders, one in d_1, and the other in d_2. Intuitively, the element represented by the spider s^* belongs either to the set $U - (A \cup B)$ or to the set $A \cup B$.

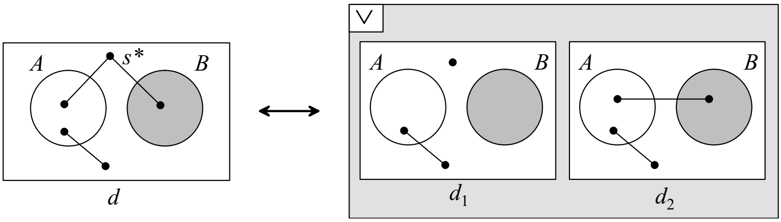

Figure 19: An application of Rule 3, splitting spiders.

Formal description. Let d be a unitary diagram, and let r, r_1 and r_2 be regions of d such that $r = r_1 \cup r_2$ and $r_1 \cap r_2 = \varnothing$. Let $s_n(r)$ be a spider in d (with habitat r). Let d_1 and d_2 be unitary diagrams such that

1. $Z(d) = Z(d_1) = Z(d_2)$;
2. $Z^*(d) = Z^*(d_1) = Z^*(d_2)$;
3. there exist spiders $s_1 \in S(d_1)$ and $s_2 \in S(d_2)$ such that

$$\eta(s_1) = r_1 \wedge \eta(s_2) = r_2$$

and

$$S(d) - \{s_n(r)\} = S(d_1) - \{s_1\} = S(d_2) - \{s_2\}.$$

Then d can be replaced by $d_1 \vee d_2$, and vice versa.

Suppose that a diagram d contains an unshaded region that is not touched by any spider. The semantics predicate gives no information about the cardinality of the set denoted by the region. The set is either empty (represented diagrammatically by shading) or non-empty (represented diagrammatically by a spider whose habitat is the region). This observation

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

forms the basis of our next rule, which does *not* require the region to be untouched by any spider.

RULE 4 (EXCLUDED MIDDLE). Let d ($\neq \perp$) be a unitary diagram with a non-shaded region $\overline{r}$. Let d_1 and d_2 be unitary diagrams that are copies of d except that d_1 contains an extra spider whose habitat is $\overline{r}$, and $\overline{r}$ is shaded in d_2. Then d can be replaced by the diagram $d_1 \vee d_2$. The rule is also reversible; that is, $d_1 \vee d_2$ can be replaced by d.

EXAMPLE 4.3. The diagram d in Figure 20 asserts that $A \cap B = \varnothing$ and $|A \cup B| \geqslant 1$, but asserts nothing about $|B|$. Each of the diagrams d_1 and d_2 asserts that $A \cap B = \varnothing$. However, d_1 also asserts that $|A \cup B| \geqslant 1$ and $|B| \leqslant 1$, whereas d_2 also asserts that $|A \cup B| \geqslant 2$ and $|B| \geqslant 1$. Now $A \cap B = \varnothing \wedge |A \cup B| \geqslant 1$ is equivalent to

$$(A \cap B = \varnothing) \wedge \big((|A \cup B| \geqslant 1 \wedge |B| \leqslant 1) \vee (|A \cup B| \geqslant 2 \wedge |B| \geqslant 1)\big).$$

It follows that we can replace d with $d_1 \vee d_2$, and vice versa.

Note that this example shows that the term ‘excluded middle’ to name the rule is being used somewhat loosely, since the semantic statements asserted by d_1 and d_2 are not mutually exclusive.

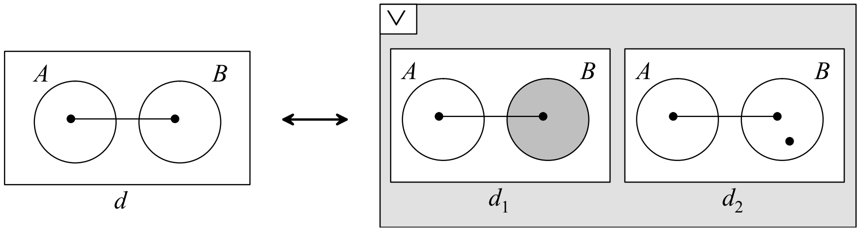

Figure 20: An application of Rule 4, excluded middle.

FORMAL DESCRIPTION. Let d be a unitary diagram containing a non-shaded region $\overline{r}$ (that is, $\overline{r} \cap Z^*(d) = \varnothing$). Let d_1 and d_2 be unitary diagrams such that

1. $Z(d) = Z(d_1) = Z(d_2)$;
2. $Z^*(d) = Z^*(d_1)$;
3. $Z^*(d) \cup \overline{r} = Z^*(d_2)$;
4. there exists $s^* \in S(d_1)$ such that $\eta(s^*) = \overline{r}$ and

$$S(d) = S(d_1) - \{s^*\} = S(d_2).$$

Then d can be replaced by $d_1 \vee d_2$, and vice versa.

The next rule in this section, called *combining*, replaces two unitary diagrams taken in conjunction by a single unitary diagram. In SD2 [31], the basic operation of combining was defined on a multi-diagram (recall that a multi-diagram is in conjunctive normal form). Unlike in the system SD2, the basic operation of combining diagrams will be performed on two unitary diagrams taken in conjunction: we will replace two unitary diagrams with a single, semantically equivalent, unitary diagram. In SD2, the basic operation of combining was defined for finitely many unitary diagrams taken in conjunction. As in the SD2 system, we combine unitary diagrams that have the same sets of zones, and that contain only spiders

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

whose habitats are single zones. The following example illustrates why the presence of spiders inhabiting regions that are not single zones is problematic when combining diagrams.

EXAMPLE 4.4. The diagram $d_1 \wedge d_2$ in Figure 21 contains two spiders whose habitats are not zones. A combined (unitary) diagram, semantically equivalent to $d_1 \wedge d_2$, would have the same set of zones as d_1 and d_2, but what spiders should it contain? We cannot deduce that spiders s_1 and s_2 denote distinct elements, since they are in different unitary diagrams and, in many set-assignments to regions, their habitats do not represent disjoint sets. Informally, each spider may represent the same element of $A - (B \cup C)$. Thus we cannot just place copies of s_1 and s_2 into a combined diagram.

Equally, we cannot deduce that s_1 and s_2 denote the same element for, in some set-assignments to regions, their habitats denote disjoint sets – informally, when $A-(B \cup C) = \varnothing$. Therefore we cannot just place one spider with habitat $\eta(s_1) \cup \eta(s_2)$ into a combined diagram.

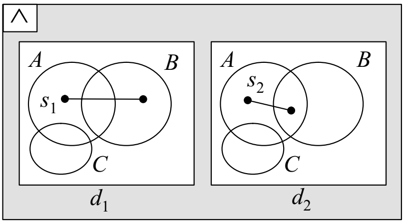

Figure 21: Spiders whose habitats are not zones.

DEFINITION 4.1. A spider diagram is an α-*diagram* if and only if the habitat of every spider is a single zone.

We now give two examples to motivate the definition of combining diagrams. In these examples, we derive results by working an the semantic level, although we will, of course, define combining diagrams purely syntactically.

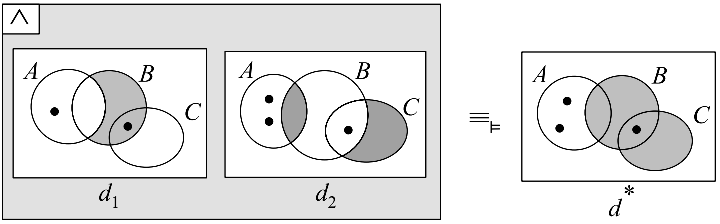

Figure 22: Combining unitary diagrams.

EXAMPLE 4.5. In Figure 22, $d_1 \wedge d_2$ is semantically equivalent to d^*. From $|A-(B\cup C)| \geqslant 1$ (asserted by d_1) and $|A-(B\cup C)| \geqslant 2$ (asserted by d_2), we deduce that $|A-(B\cup C)| \geqslant 2$, which is asserted by d^*. Similarly, from d_1 we have $|(B \cap C) - A| = 1$ (since the zone $(\{B, C\}, \{A\})$ is shaded and contains a single spider), and from d_2 we have $|(B\cap C)-A| \geqslant 1$. Hence, from $d_1 \wedge d_2$ we have $|(B \cap C) - A| = 1$, which is also asserted by d^*.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

The diagram d_1 also represents $B - (A \cup C) = \varnothing$, whereas d_2 provides no information about this set. Therefore $d_1 \wedge d_2$ represents $B - (A \cup C) = \varnothing$, and this is also asserted by d^*. In fact, $d_1 \wedge d_2$ is semantically equivalent to d^*.

EXAMPLE 4.6. In Figure 23, the two unitary components of $d_1 \wedge d_2$ represent conflicting information. The diagram d_1 asserts that $A - (B \cup C) = \varnothing$, whereas the diagram d_2 asserts that $|A - (B \cup C)| \geqslant 1$. Thus $d_1 \wedge d_2$ has no model.

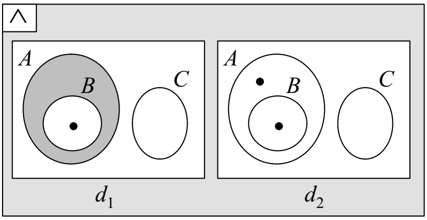

Figure 23: An unsatisfiable diagram.

The operation of combining unitary α-diagrams will be performed on diagrams with the same sets of zones. Combining unitary diagrams produces a unitary diagram that is semantically equivalent to the original unitary diagrams when their semantic information is taken in conjunction.

DEFINITION 4.2 (COMBINING UNITARY α-DIAGRAMS WITH THE SAME ZONE SETS). Let d_0 and d_1 be unitary α-diagrams such that $Z(d_0) = Z(d_1)$ or $d_0 = \perp$ or $d_1 = \perp$. Then their *combination*

$$d^* = d_0 * d_1$$

is a unitary α-diagram, defined as follows.

1. If $d_0 = \perp$ or $d_1 = \perp$, then $d^* = \perp$.

2. If there is a zone that is shaded in one diagram but contains more spiders in the other diagram, then $d_0 \wedge d_1$ is unsatisfiable and $d^* = \perp$. More precisely, if there exists $z \in Z(d_i)$, $i = 0, 1$, such that

$$z \in Z^*(d_j)$$

and $S(\{z\}, d_i) - S(\{z\}, d_j) \neq \varnothing$, where $j = 1 - i$, then $d^* = \perp$.

3. Otherwise d^* is a unitary α-diagram such that the following statements hold.

(i) The set of zones of the combined diagram is the union of the zone sets of the original diagrams:

$$Z(d^*) = Z(d_0) \cup Z(d_1).$$

(ii) Shaded zones in the combined diagram are shaded in one (or both) of the original diagrams:

$$Z^*(d^*) = Z^*(d_0) \cup Z^*(d_1).$$

(iii) The number of spiders in any zone in the combined diagram is the maximum number of spiders inhabiting that zone in the original diagrams:

$$\forall z \in Z(d^*) \bullet S(\{z\}, d^*) = S(\{z\}, d_0) \cup S(\{z\}, d_1).$$

RULE 5 (COMBINING). Let d_0 and d_1 be unitary α-diagrams satisfying $Z(d_0) = Z(d_1)$ or $d_0 = \bot$ or $d_1 = \bot$. Then $d_0 \wedge d_1$ may be replaced by $d_0 * d_1$, and vice versa.

We now introduce a collection of rules, each of which has an (obvious) analogy in logic.

RULE 6 (CONNECTING A DIAGRAM). Let D_1 and D_2 be two spider diagrams. Then D_1 can be replaced by $D_1 \vee D_2$.

RULE 7 (INCONSISTENCY). The diagram $\bot$ may be replaced by any diagram.

RULE 8 (IDEMPOTENCY OF $\vee$). Let D a be spider diagram. Then D may be replaced by $D \vee D$, and vice versa.

RULE 9 (IDEMPOTENCY OF $\wedge$). Let D a be spider diagram. Then D may be replaced by $D \wedge D$, and vice versa.

RULE 10 (DISTRIBUTIVITY OF $\vee$). Let D_1, D_2 and D_3 be spider diagrams. Then $D_1 \vee (D_2 \wedge D_3)$ may be replaced by $(D_1 \vee D_2) \wedge (D_1 \vee D_3)$, and vice versa.

RULE 11 (DISTRIBUTIVITY OF $\wedge$). Let D_1, D_2 and D_3 be spider diagrams. Then $D_1 \wedge (D_2 \vee D_3)$ may be replaced by $(D_1 \wedge D_2) \vee (D_1 \wedge D_3)$, and vice versa.

RULE 12 (SIMPLIFICATION OF $\vee$). Let D_1, D_2 and D_3 be spider diagrams. If diagram D_2 can be transformed into diagram D_3 by one of the first 11 transformation rules, then $D_1 \vee D_2$ may be replaced by $D_1 \vee D_3$.

RULE 13 (SIMPLIFICATION OF $\wedge$). Let D_1, D_2 and D_3 be spider diagrams. If diagram D_2 can be transformed into diagram D_3 by one of the first 11 transformation rules, then $D_1 \wedge D_2$ may be replaced by $D_1 \wedge D_3$.

4.2. *Derived reasoning rules*

The reasoning rules introduced in the previous section produce, as we shall see, a sound and complete system. However, there are a number of additional, intuitive rules that may be derived from them, and which aid the reasoning process. In this section we introduce some of these 'derived reasoning rules'. In practice, users of any reasoning system may choose to introduce other derived rules (as theorems) to suit their particular purposes.

DEFINITION 4.3. Let D and D' be two spider diagrams. We write $D \Vdash D'$ if and only if D can be transformed to D' by a single application of one of the first 13 rules of transformation. We say that D' is *obtainable* from D, denoted $D \vdash D'$, if and only if there is a sequence of diagrams $\langle D^1, D^2, \ldots, D^m \rangle$ such that $D^1 = D$, $D^m = D'$ and $D^k \Vdash D^{k+1}$ for each k (where $1 \leqslant k < m$).

Two spider diagrams D and D' are *syntactically equivalent*, denoted $D \equiv_{\vdash} D'$, if and only if $D \vdash D'$ and $D' \vdash D$.

DEFINITION 4.4. Let ρ be a (finite) set of reasoning rules, and let r be a reasoning rule such that $r \notin \rho$. The rule r is *derived from* ρ if and only if, for all spider diagrams D_1 and D_2, if D_1 can be transformed into D_2 by a single application of r, then there exists a sequence of diagrams $\langle D_1 = D^1, D^2, \ldots, D^m = D_2 \rangle$ such that for each k $(1 \leqslant k < m)$ D^k is transformed into D^{k+1} by a single application of a reasoning rule in ρ.

We will say, informally, that a rule is *derived* if it is derived from the (set of the) first 13 reasoning rules, in the sense of Definition 4.4. Our first derived rule allows erasure of shading in any region, and amounts to 'throwing away' the information on the upper bounds of the cardinalities of the sets represented by those regions affected.

RULE 14 (ERASURE OF SHADING). We may erase shading from any region. Let d be a unitary diagram with shaded region r, and let d' be a copy of d except that r is not shaded. Then d may be replaced by d'.

This rule is derived from Rule 6 (connecting a diagram) and Rule 4 (excluded middle). The proof strategy (to show that Rule 14 is a derived rule) is illustrated in the following example.

EXAMPLE 4.7. To the diagram d in Figure 24, we connect diagram d_1 using Rule 6. Then Rule 4 (excluded middle) is applied, obtaining diagram d'.

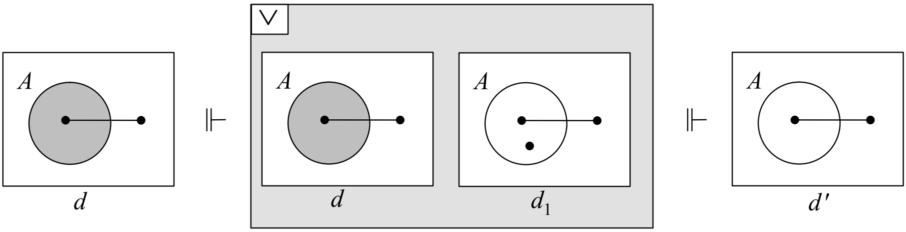

Figure 24: Erasure of shading.

FORMAL DESCRIPTION. Let d be a unitary diagram that contains a shaded region r^*. Let d' be a unitary diagram such that

1. $Z(d) = Z(d')$;
2. $Z^*(d) - r^* = Z^*(d')$;
3. $\mathrm{SI}(d) = \mathrm{SI}(d')$.

Then d can be replaced by d'.

A second derived rule allows the extension of a spider's habitat: we call this 'adding feet to a spider'. The validity of the rule is intuitively obvious: it allows us to replace the containing set for the element denoted by a spider by any superset.

RULE 15 (ADDING FEET TO A SPIDER). Let d be a unitary diagram with a spider s^*, whose habitat does not include the zone z of d. Let d' be a copy of d, except that the spider s^* has an extra foot placed in the zone z. Then d may be replaced by d'.

The proof strategy (to show that Rule 15 is a derived rule) is illustrated by the following example.

EXAMPLE 4.8. In Figure 25, we add a foot to the spider s_1 in the diagram d into the zone $(\emptyset, \{A, B\})$. First we use Rule 6 to connect the diagram d_1, which is a copy of d except that s_1 has been replaced by a new spider in the zone $(\emptyset, \{A, B\})$. Then Rule 3 (splitting spiders) is applied to obtain the diagram d', where the habitat of s_1 has been extended into the new zone $(\emptyset, \{A, B\})$.

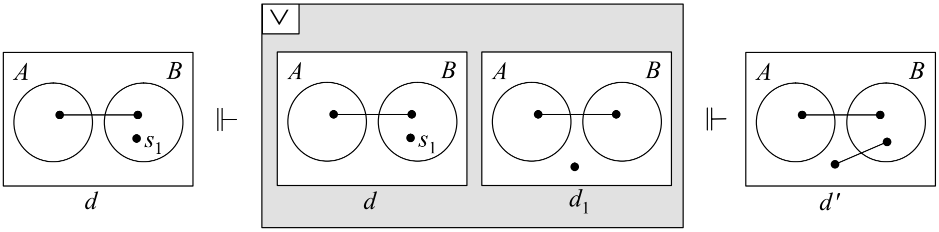

Figure 25: Adding feet to a spider.

FORMAL DESCRIPTION. Let d be a unitary diagram that contains a spider $s_n(r)$ whose habitat r does not contain a zone z of d. Let d' be a unitary diagram such that

1. $Z(d) = Z(d')$;
2. $Z^*(d) = Z^*(d')$;
3. there exists a spider $s^* \in S(d')$ such that $\eta(s^*) = r \cup \{z\}$ and $S(d') - \{s^*\} = S(d') - \{s_n(r)\}$.

Then d can be replaced by d'.

RULE 16 (ERASURE OF A SPIDER). We may erase a complete spider whose habitat is a non-shaded region. Let d be a unitary diagram with a spider s whose habitat is a non-shaded region. Let d' be a copy of d, except that s has been removed. We may replace d with d'.

The proof strategy (to show that Rule 16 is a derived rule) is illustrated by the following example.

EXAMPLE 4.9. In Figure 26, we erase the spider s. First, we use Rule 6 to connect a diagram d_1, which is a copy of d except that the spider s is missing and its former habitat is shaded. The diagram d' is then obtained by an application of Rule 4, excluded middle.

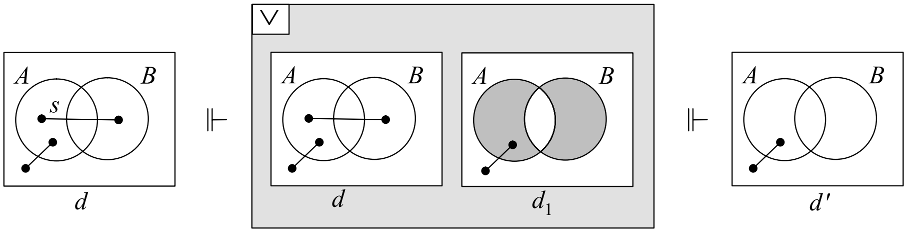

Figure 26: Erasing a spider.

FORMAL DESCRIPTION. Let d be a unitary diagram such that there exists $(n, r) \in \mathrm{SI}(d)$ with $r \cap Z^*(d) = \emptyset$. Let d' be a unitary diagram such that

1. $Z(d) = Z(d')$;
2. $Z^*(d) = Z^*(d')$; and
3. $S(d) - \{s_n(r)\} = S(d')$.

Then d can be replaced by d'.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

Erasing a contour from a concrete diagram may result in a diagram that is not well formed. There are two potential problems that may arise because, in erasing a contour, some pairs of zones will 'coalesce' to form single zones. If one zone of such a pair is shaded and the other is unshaded, then the new 'coalesced' zone is partially shaded. Similarly, if both zones of a pair contain a foot of the same spider, then, in the resulting diagram, the spider has two feet in the new zone. These problems are illustrated in the middle diagram of Figure 27.

RULE 17 (ERASURE OF A CONTOUR LABEL). Let d be a unitary diagram containing at least one contour label, and let d' be the diagram obtained from d after erasing a contour label as follows.

1. Any shading remaining in only part of a zone is also erased.
2. If a spider has feet in two zones that combine to form a single zone with the erasure of the contour label, then these feet are replaced by a single foot connected to the rest of the spider.

Then we may replace d with d'.

EXAMPLE 4.10. In Figure 27, erasing the contour C from d leaves the region inside A partially shaded and the spider s_1 having two feet in the new zone $(\varnothing, \{A, B\})$. To ensure that diagram d' is well formed, the partial shading in the new region $(\{A\}, \{B\})$ is erased, and the two feet of s_1 are combined to form a single foot. Applying this rule loses all information about the set represented by the contour with label C. For example, from diagram d we can deduce that $|A - C| = 1$, but from diagram d' we can infer (about the cardinality of A) only that $|A| \geqslant 1$.

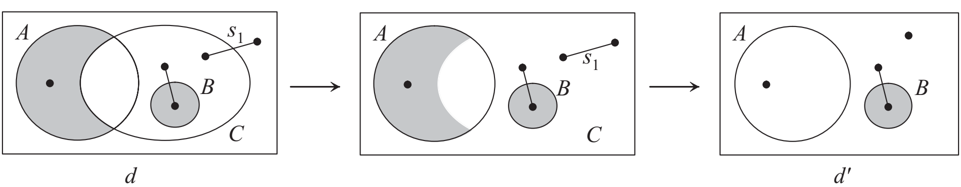

Figure 27: An application of Rule 2, erasure of a contour label.

FORMAL DESCRIPTION. Let d be a unitary diagram and let $l^* \in L(d)$. Let d' be a unitary diagram satisfying the following conditions.

1. $L(d') = L(d) - \{l^*\}$.
2. There exists a surjection $h\colon Z(d) \longrightarrow Z(d')$ defined by

$$h(a, L(d) - a) = \big(a - \{l^*\}, L(d) - (a \cup \{l^*\})\big).$$

3. (i) For all unshaded zones z of d, the zone $h(z)$ is unshaded in d'.

   (ii) For all shaded zones z of d, if $h(z)$ is unshaded in d' then $h(z) = h(z^*)$ for some unshaded zone z^* of d.
4. There exists a bijection $\sigma : S(d) \longrightarrow S(d')$ such that for all $s \in S(d)$,

$$\eta(\sigma(s)) = \{h(z)\colon z \in \eta(s)\} = h(\eta(s)).$$

Then d can be replaced with d'.

The proof strategy (to show that Rule 17 is a derived rule) is illustrated by the following example.

EXAMPLE 4.11. In Figure 28, we show how to erase the contour C from d (as in Example 4.10). The idea is to modify the diagram d so that (the converse of) Rule 1, introduction of a contour label, may be applied. First, we use Rule 2 to introduce the shaded zone $(\{B\}, \{A, C\})$; then we add various feet to the spiders, using Rule 15 several times. Next we remove the shading from the zone $(\{A\}, \{B, C\})$ using Rule 14. We can then obtain the diagram d' by an application of Rule 1.

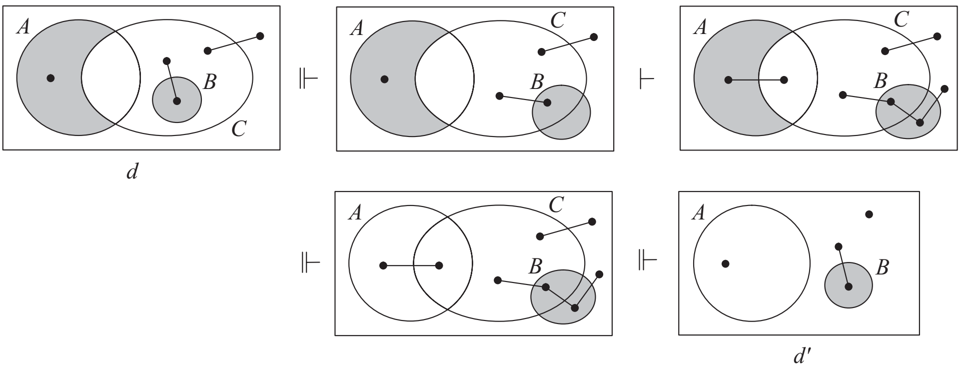

Figure 28: Erasing a contour label.

Since each of the rules introduced in this section may be derived from the first 13 rules given in the previous section, we have the following theorem.

THEOREM 4.1. *Let D and D' be spider diagrams. D' is obtainable from D using all 17 rules if and only if $D \vdash D'$.* □

The following two lemmas are simple extensions of the simplification rules, 12 and 13.

LEMMA 4.1. *Let D_1 and D_2 be spider diagrams such that $D_1 \vdash D_2$, and let D be a spider diagram. Then $D \vee D_1 \vdash D \vee D_2$.* □

LEMMA 4.2. *Let D_1 and D_2 be spider diagrams such that $D_1 \vdash D_2$, and let D be a spider diagram. Then $D \wedge D_1 \vdash D \wedge D_2$.* □

COROLLARY 4.1 (RULE OF REPLACEMENT). *Let D_1, D_2, D_3 and D_4 be spider diagrams such that $D_1 \vdash D_2$ and $D_3 \vdash D_4$. Let $\diamond$ be $\vee$ or $\wedge$. Then $D_1 \diamond D_3 \vdash D_2 \diamond D_4$.* □

Let $\{D_1, D_2, \ldots, D_n\}$ be a set of spider diagrams. Then we write

$$\bigvee_{1 \leqslant i \leqslant n} D_i$$

for $D_1 \vee D_2 \vee \ldots \vee D_n$ and

$$\bigwedge_{1 \leqslant i \leqslant n} D_i$$

for $D_1 \wedge D_2 \wedge \ldots \wedge D_n$.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

THEOREM 4.2 (RULE OF CONSTRUCTION). *For $i = 1, \ldots, n$ let D_i be a spider diagram, and let D be a spider diagram. If $D_i \vdash D$ for $i = 1, \ldots, n$ then*

$$\Big(\bigvee_{1 \leqslant i \leqslant n} D_i \Big) \vdash D. \qquad \square$$

5. *Soundness*

An essential aspect of any reasoning system is, of course, the soundness of the system. In this section we start by considering the validity of each reasoning rule. After proving the validity of each transformation rule, it follows by induction that the system is sound. The omitted proofs and further details for the sketched proofs can be found in [38].

DEFINITION 5.1. 1. A transformation rule r is *valid* if, for all diagrams D and D', if D' is obtained from D by a single application of r $(D \Vdash D')$ then $D \vDash D'$.

2. The diagrams D_1 and D_2 are *semantically equivalent*, denoted $D_1 \equiv_{\vDash} D_2$, if and only if $D_1 \vDash D_2$ and $D_2 \vDash D_1$.

The general strategy of the proof of validity for each rule is to assume that a set-assignment to regions satisfies 'the diagram we start with', and to show that it satisfies the diagram that results after applying the rule concerned.

LEMMA 5.1. *Rule 1, introduction of a contour label, is valid.* □

LEMMA 5.2. *Rule 2, introduction of a shaded zone, is valid.*

Sketch of the proof. The strategy is to show that the introduced zone, z^*, represents the empty set in any set-assignment to regions that is a model for d_1 or for d_2. It is then straightforward to show that $d_1 \equiv_{\vDash} d_2$. □

To prove the validity of Rule 3, splitting spiders, we need the result that states that if a set-assignment to regions satisfies a unitary diagram d and $d_1 \vee d_2$ is a diagram obtained from d by applying the splitting spiders rule, then the distinct spiders condition holds for one of the disjuncts d_1 and d_2, or the shading condition holds for the other disjunct.

LEMMA 5.3. *Let d be a unitary diagram, and suppose that there are $n > 0$ spiders with habitat the region $r \in R(d)$. Further, suppose that there are regions r_1 and r_2 such that $r = r_1 \cup r_2$ and $r_1 \cap r_2 = \varnothing$.*
Let d_1 and d_2 be unitary diagrams such that

1. $Z(d) = Z(d_1) = Z(d_2)$;
2. $Z^*(d) = Z^*(d_1) = Z^*(d_2)$;
3. *there exist spiders $s_1 \in S(d_1)$ and $s_2 \in S(d_2)$ such that $\eta(s_1) = r_1$ and $\eta(s_2) = r_2$ and*

$$S(d) - \{s_n(r)\} = S(d_1) - \{s_1\} = S(d_2) - \{s_2\},$$

(where $s_n(r)$ denotes the 'nth spider in r').

Let $m = (\mathbf{U}, \Psi)$ be a set-assignment to regions. If $m \models d$, then

$$\Big(\bigwedge_{r \in R(d_1)} |\Psi(r)| \geqslant |S(r, d_1)| \vee \bigwedge_{r \in R^*(d_2)} |\Psi(r)| \leqslant |T(r, d_2)| \Big) \wedge \Big(\bigwedge_{r \in R(d_2)} |\Psi(r)| \geqslant |S(r, d_2)| \vee \bigwedge_{r \in R^*(d_1)} |\Psi(r)| \leqslant |T(r, d_1)| \Big). \qquad \square$$

LEMMA 5.4. *Rule 3, splitting spiders, is valid.*

Sketch of proof. Let $m = (\mathbf{U}, \Psi)$ be a set-assignment to regions. It is easy to show that $P_{d_1}(m) \Longrightarrow P_d(m)$ and $P_{d_2}(m) \Longrightarrow P_d(m)$. Hence $(P_{d_1}(m) \vee P_{d_2}(m)) \Longrightarrow P_d(m)$. To show that $P_d(m) \Longrightarrow (P_{d_1}(m) \vee P_{d_2}(m))$, show that

$$\begin{aligned} P_d(m) \Longrightarrow \bigcup_{z \in Z(d_1)} \Psi(z) = \mathbf{U} \wedge \bigcup_{z \in Z(d_2)} \Psi(z) = \mathbf{U} \\ \wedge \Big(\bigwedge_{r \in R(d_1)} |\Psi(r)| \geqslant |S(r, d_1)| \vee \bigwedge_{r \in R(d_2)} |\Psi(r)| \geqslant |S(r, d_2)| \Big) \\ \wedge \Big(\bigwedge_{r \in R^*(d_1)} |\Psi(r)| \leqslant |T(r, d_1)| \vee \bigwedge_{r \in R^*(d_2)} |\Psi(r)| \leqslant |T(r, d_2)| \Big). \end{aligned}$$

Then use Lemma 5.3 to give $P_d(m) \Longrightarrow (P_{d_1}(m) \vee P_{d_2}(m))$. □

Part of the proof of the validity of Rule 4 (excluded middle) requires us to show that, if $d_1 \vee d_2$ is a diagram obtained from a unitary diagram d by an application of the excluded middle rule, and the distinct spiders condition for d_1 fails, then the shading condition for d_2 holds. In order to do this, the following lemma is required. This states that if a region r_1 represents a set whose cardinality is the same as that of $|S(r_1, d)|$ and the distinct spiders condition holds for d, then any subregion r_2 of r_1 represents a set whose cardinality is at most the number of spiders that inhabit r_1 and touch r_2.

LEMMA 5.5. *Let d be a unitary diagram, and let $m = (\mathbf{U}, \Psi)$ be a set-assignment to regions. Let $r_1 \in R(d)$ be a region such that $|\Psi(r_1)| = |S(r_1, d)|$. If the distinct spiders condition for d holds, then, for each subregion r_2 of r_1,*

$$|\Psi(r_2)| \leqslant |S(r_1, d) \cap T(r_2, d)|.$$ □

LEMMA 5.6. *Rule 4, excluded middle, is valid.*

Proof. It is straightforward to show that $d_1 \vDash d$ and $d_2 \vDash d$, and so $d_1 \vee d_2 \vDash d$. It is obvious that the plane tiling condition for d implies that the plane tiling condition holds for d_1 and for d_2. Moreover, it is also obvious that the distinct spiders condition for d implies that the distinct spiders condition holds for d_2, and that the shading condition for d implies that the shading condition holds for d_1.

Suppose that $m \models d$, and that the distinct spiders condition for d_1 is false. We show that the shading condition for d_2 is true. Since the distinct spiders condition for d_1 is false, there exists a region r_1 such that

$$|\Psi(r_1)| < |S(r_1, d_1)| \wedge |\Psi(r_1)| \geqslant |S(r_1, d)|.$$

Since $|S(r_1, d_1)|$ can be at most one bigger than $|S(r_1, d)|$, it follows that

$$|\Psi(r_1)| = |S(r_1, d)|.$$

For any region $r_3 \in R^*(d_2)$, we wish to show that $|\Psi(r_3)| \leqslant |T(r_3, d_2)|$. Now $r_3 = (r_3 \cap r_1) \cup (r_3 - r_1)$ and, since $r_3 \cap r_1 \subseteq r_1$,

$$|\Psi(r_3 \cap r_1)| \leqslant |S(r_1, d) \cap T(r_3 \cap r_1, d)|$$

by Lemma 5.5. Furthermore, if $r_3 - r_1 \in R^*(d)$, then, by the shading condition for d,

$$|\Psi(r_3 - r_1)| \leqslant |T(r_3 - r_1, d)|.$$

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

Alternatively, $r_3 - r_1 = \varnothing$, and it is trivial that

$$|\Psi(r_3 - r_1)| \leqslant |T(r_3 - r_1, d)|.$$

Thus

$$\begin{aligned} |\Psi(r_3)| &= |\Psi(r_3 \cap r_1)| + |\Psi(r_3 - r_1)| \\ &\leqslant |S(r_1, d) \cap T(r_3 \cap r_1, d)| + |T(r_3 - r_1, d)|. \end{aligned} \tag{1}$$

Now

$$S(r_1, d) \cap T(r_3 \cap r_1, d) \subseteq T(r_3, d)$$

and

$$T(r_3 - r_1, d) \subseteq T(r_3, d).$$

Also,

$$S(r_1, d) \cap T(r_3 \cap r_1, d) \cap T(r_3 - r_1, d) = \varnothing.$$

Thus

$$|S(r_1, d) \cap T(r_3 \cap r_1, d)| + |T(r_3 - r_1, d)| \leqslant |T(r_3, d)|.$$

Using equation (1), we deduce that

$$|\Psi(r_3)| \leqslant |T(r_3, d)| = |T(r_3, d_2)|,$$

as required. Therefore the shading condition holds for d_2. Hence $P_d(m) \iff (P_{d_1}(m) \vee P_{d_2}(m))$ and $d \equiv_{\models} d_1 \vee d_2$; that is, Rule 4 is valid. □

For Rules 6 to 13, the proofs of validity are trivial.

Lemma 5.7. *Rules 6 to 13 are valid.* □

It now follows by induction that the reasoning system that uses only the first 13 reasoning rules (other than combining) is valid. More precisely, if D_1 and D_2 are spider diagrams such that there exists a sequence of diagrams $\langle D^1, D^2, \ldots, D^m \rangle$, where $D_1 = D^1$, $D_2 = D^m$ and D^k can be transformed into D^{k+1} $(1 \leqslant k < m)$ using a single application of one of the first 13 transformation rules (other than combining), then $D_1 \models D_2$. The following result is an immediate consequence.

Corollary 5.1. *Rules 14 to 17 (the derived reasoning rules) are valid.* □

Finally, we consider the validity of the rule for combining diagrams.

Lemma 5.8. *Rule 5, the rule for combining diagrams, is valid.*

Sketch of the proof. There are three cases, corresponding to those in the definition of the combined diagram.

1. If either d_0 or d_1 is $\perp$, then $d_0 * d_1 = \perp$, and the result is obvious.

2. Assume that there is a zone, z, that is shaded in one of d_0 and d_1, d_j say, and contains more spiders in d_{1-j} then $d_0 * d_1 = \perp$. In this case, for any set-assignment to regions $m = (U, \Psi)$, if $m \models d_j$ then $\Psi(z) = |S(\{z\}, d_j)|$, and if $m \models d_{1-j}$ then $\Psi(z) > |S(\{z\}, d_j)|$. Therefore $d_0 \wedge d_1$ is unsatisfiable and $d_0 \wedge d_1 \equiv_{\models} d_0 * d_1$.

3. For the final case, it can be shown that $d_0 * d_1 \vdash d_0$ and $d_0 * d_1 \vdash d_1$, using the rules *erasure of shading* and *erasure of a spider*. By the validity of these rules, $d_0 * d_1 \vDash d_0 \wedge d_1$. To show that $d_0 \wedge d_1 \vDash d_0 * d_1$, the strategy is to consider each zone of $d_0 * d_1$ in turn, and to show that the distinct spiders condition and the shading condition hold for that zone in $d_0 * d_1$. Noting that $d_0 * d_1$ is an α-diagram, we see that it then follows that the distinct spiders condition and the shading condition hold for $d_0 * d_1$. It is trivial that the plane tiling condition holds for $d_0 * d_1$. □

The soundness of the system now follows by induction.

THEOREM 5.1 (SOUNDNESS). *Let D_1 and D_2 be spider diagrams. If $D_1 \vdash D_2$, then $D_1 \vDash D_2$.* □

6. *Completeness*

In this section we show that our system of spider diagrams is complete. The strategy that we adopt is a simplified version of that used in the proof of completeness for the SD2 system, given by Molina [31]. Given $D_1 \vDash D_2$, the strategy is to convert both D_1 and D_2 into disjunctions of unitary α-diagrams, all with the same label set, giving D_1' and D_2' respectively, using only rules that preserve information. The excluded middle rule is then applied to D_1', giving D_1'', until enough spiders and sufficient shading are present to allow erasure rules to be applied to D_1'' until all the unitary parts on the left-hand side appear on the right-hand side. The completeness result will then follow. We start by proving a completeness result for unitary α-diagrams, all with the same zone set.

6.1. *Completeness for unitary α-diagrams*

We show that if $d_1 \vDash d_2$, for α-diagrams d_1 and d_2 with the same zone sets, then we can erase shading and spiders from d_1 to give d_2, and therefore $d_1 \vdash d_2$.

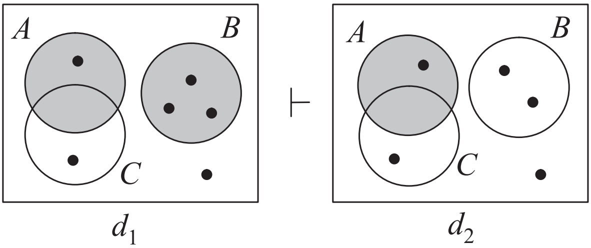

Figure 29: Two α-diagrams.

EXAMPLE 6.1. The diagrams d_1 and d_2 in Figure 29 satisfy the following conditions.

1. Every shaded zone in d_2 is shaded in d_1, and contains the same number of spiders in both diagrams.

2. Every zone in d_2 contains the same number of spiders, or fewer, than in d_1.

Under these conditions, the diagram d_2 can be obtained from d_1 by applying Rules 14 (erasure of shading) and 16 (erasure of a spider). The properties 1 and 2 above relate to properties 3(i) and 3(ii) in Theorem 6.1 below.

The transformation from d_1 to d_2 is illustrated in Figure 30. The zone $(\{B\}, \{A, C\})$ is shaded in d_1, but not in d_2. Remove this shading using Rule 14, giving d_3. Next delete the extra spider inhabiting the zone $(\{B\}, \{A, C\})$.

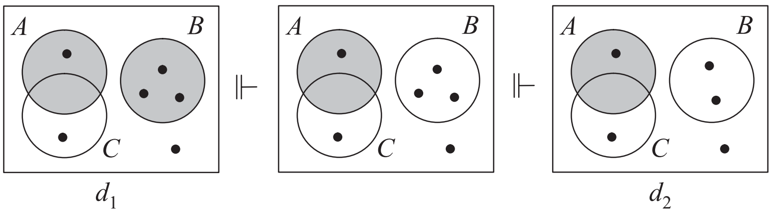

Figure 30: Transformation of α-diagrams.

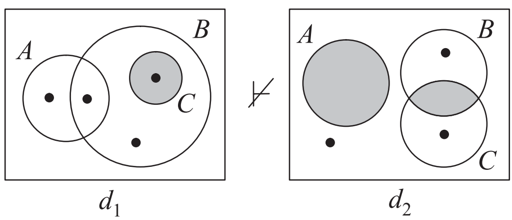

Figure 31: Two α-diagrams.

EXAMPLE 6.2. The diagram d_2 in Figure 31 cannot be obtained from d_1, for two reasons.

1. The zone $(\{A\}, \{B, C\})$ is shaded in d_2, but not shaded in d_1. There is a model for d_1 that will cause the shading condition for d_2 to fail.

2. The zone $(\emptyset, \{A, B, C\})$ contains a single spider in d_2, but is untouched in d_1. Again, we can deduce that there is a model for d_1 that does not satisfy d_2, this time causing the distinct spiders condition for d_2 to fail.

Hence $d_1 \nVdash d_2$.

The following theorem gives syntactic conditions on unitary α-diagrams, equivalent to semantic and syntactic entailment. The theorem forms the heart of the proof of completeness. Our theorem is a modified form of the corresponding result in [**31**], to take account of the fact the our spider diagrams are now based on Euler, rather than Venn, diagrams.

THEOREM 6.1. *Let d_1 $(\neq \perp)$ and d_2 $(\neq \perp)$ be two unitary α-diagrams. If $Z(d_1) = Z(d_2)$, then the following three statements are equivalent.*

1. $d_1 \vdash d_2$.
2. $d_1 \vDash d_2$.
3. (i) *Every zone that is shaded in d_2 is shaded in d_1, and contains the same number of spiders in both diagrams:*

$$Z^*(d_2) \subseteq Z^*(d_1) \wedge \forall z \in Z^*(d_2) \bullet S(\{z\}, d_2) = S(\{z\}, d_1).$$

(ii) *Every zone in d_2 contains at most the same number of spiders as in d_1:*

$$\forall z \in Z(d_2) \bullet S(\{z\}, d_2) \subseteq S(\{z\}, d_1).$$

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

Proof. (a) $1 \Longrightarrow 2$. This follows by soundness.

(b) $2 \Longrightarrow 3$. We prove the contrapositive.
That is, we prove that $(\neg 3(a) \vee \neg 3(b)) \Longrightarrow \neg(d_1 \vDash d_2)$, which is equivalent to

(b$_1$) $\neg 3(a) \Longrightarrow \neg(d_1 \vDash d_2)$ and
(b$_2$) $\neg 3(b) \Longrightarrow \neg(d_1 \vDash d_2)$.

(b$_1$) Suppose that there exists a shaded zone, z, in d_2 such that

$$z \notin Z^*(d_1) \vee S(\{z\}, d_2) \neq S(\{z\}, d_1).$$

Suppose first that $z \notin Z^*(d_1)$. Let $m = (\mathbf{U}, \Psi)$ be a model for d_1 such that $|\Psi(z)| > |S(\{z\}, d_2)|$. Since z is shaded in d_2 and d_2 is an α-diagram, m does not satisfy d_2 (the shading condition fails).

Alternatively, suppose that $S(\{z\}, d_2) \neq S(\{z\}, d_1)$. Let $m = (\mathbf{U}, \Psi)$ be a model for d_1 such that

$$|\Psi(z)| = |S(\{z\}, d_1)|,$$

for example the model constructed in Theorem 3.3. Then the shading condition or the distinct spiders condition fails for d_2, since it is not the case that

$$|S(\{z\}, d_2)| \leqslant |\Psi(z)| \leqslant |T(\{z\}, d_2)| = |S(\{z\}, d_2)|.$$

So in either case there is a model for d_1 such that $m \nvDash d_2$, so $\neg 3(a) \Longrightarrow \neg(d_1 \vDash d_2)$.

(b$_2$) $\neg 3(b) \Longrightarrow \neg(d_1 \vDash d_2)$. Suppose that there exists a zone, z, in $Z(d_2)$ such that $S(\{z\}, d_2) \nsubseteq S(\{z\}, d_1)$. Then $|S(\{z\}, d_2)| > |S(\{z\}, d_1)|$. Let $m = (\mathbf{U}, \Psi)$ be a model for d_1 such that $|\Psi(z)| = |S(\{z\}, d_1)|$. Then $|\Psi(z)| < |S(\{z\}, d_2)|$ and the distinct spiders condition fails for d_2, so $\neg(m \vDash d_2)$. Thus $\neg 3(b) \Longrightarrow \neg(d_1 \vDash d_2)$.

Hence $2 \Longrightarrow 3$.

(c) $3 \Longrightarrow 1$. Erase the shading from the region $Z^*(d_1) - Z^*(d_2)$ in d_1 by applying Rule 14 (erasure of shading), yielding d_3. The diagram d_3 satisfies $Z(d_3) = Z(d_2)$ and $Z^*(d_3) = Z^*(d_2)$. Any shaded zone in d_2 contains the same number of spiders in d_1, by 3(i), and therefore in d_3. Any zone that is not shaded in d_2 contains at most the same number of spiders in d_2 as in d_1, by 3(ii), and therefore in d_3. Thus we can apply Rule 16 (erasure of a spider) repeatedly to d_3, removing all extra spiders and no others, yielding d_2. Thus $d_1 \vdash d_2$. Hence $3 \Longrightarrow 1$.

Therefore all three statements are equivalent. □

Hence for unitary α-diagrams with the same zone sets, the system is complete.

6.2. *Associated contour diagrams*

Recall that we are aiming to replace a diagram with a disjunction of unitary α-diagrams, each with the same zone set. To do so, the combining rule must be used. When combining two unitary diagrams, we require the sets of zones to be identical. Thus we may need to introduce contours into each diagram. It is convenient to define, for a unitary diagram d and set of contours $L \supseteq L(d)$, the diagram d^L obtained from d by introducing the contours in $L - L(d)$. The following definition also extends this notion to compound diagrams (where $L(D_1 \diamond D_2)$, $\diamond = \wedge$ or $\vee$, is defined to be $L(D_1) \cup L(D_2)$).

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

DEFINITION 6.1. Let L be a finite subset of $\mathcal{L}$.

1. Let $d \neq \perp$ be a unitary diagram such that $L \supseteq L(d)$. A *contour diagram associated with* d, denoted d^L, is any unitary diagram such that $L = L(d^L)$ and $d \equiv_\vdash d^L$. If $d = \perp$, then $d^L = \perp$.

2. Let $D = D_1 \diamond D_2$, where $\diamond$ is $\wedge$ or $\vee$, be any spider diagram not in $\mathcal{D}_0$ such that $L \supseteq L(D)$. A *contour diagram associated with* D is $D^L = D_1^L \diamond D_2^L$, where D_1^L and D_2^L are contour diagrams associated with D_1 and D_2 respectively.

EXAMPLE 6.3. Associated contour diagrams need not be unique. Consider the diagram d (with $L(d) = \{A, B\}$) in Figure 32 and $L = \{A, B, C\}$. Two associated contour diagrams $d^L = d_1$ and $d^L = d_2$ are shown. The diagram d_1 is obtained by introducing C directly into d, whereas the diagram d_2 is obtained by first converting d to Venn form, and then introducing C.

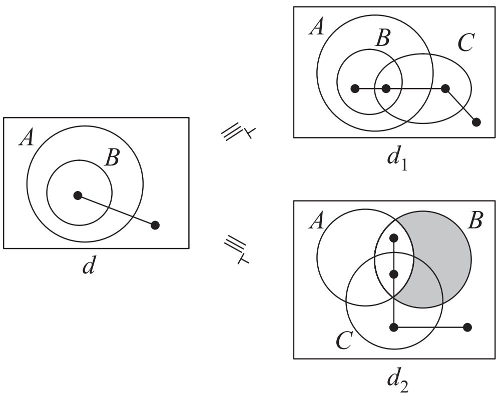

Figure 32: A diagram with two associated contour diagrams.

THEOREM 6.2. *Let L be a finite subset of $\mathcal{L}$, and let D be any spider diagram such that $L \supseteq L(D)$. There exists a contour diagram D^L, associated with D, and any such contour diagram is syntactically equivalent to D, $D \equiv_\vdash D^L$.*

Sketch of the proof. The proof is achieved by induction on the depth of D in the inductive construction. □

6.3. *Associated zone diagrams*

The next stage in our completeness proof strategy is to introduce zones until all unitary parts have the same zone sets. To formalize this, we now define associated zone diagrams. In order to do so, we first define the set of *unitary components* of a spider diagram.

DEFINITION 6.2. Let D be a spider diagram. The set of *unitary components of* D, denoted $\text{comp}(D)$, is defined as follows.

1. If $D \in \mathcal{D}_0$ then $\text{comp}(D) = \{D\}$.

2. If $D = D_1 \diamond D_2$ for some $D_1, D_2 \in \mathcal{D}$, $\diamond \in \{\vee, \wedge\}$ then $\text{comp}(D) = \text{comp}(D_1) \cup \text{comp}(D_2)$.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

DEFINITION 6.3. Let Z be a finite subset of $\mathcal{Z}$ such that

$$\forall (a_1, b_1), (a_2, b_2) \in Z \bullet a_1 \cup b_1 = a_2 \cup b_2.$$

Given Z, for any unitary diagram d ($\neq \perp$) such that $Z \supseteq Z(d)$, the *zone diagram associated with* d, denoted d^Z, is the unitary diagram defined as follows.

1. The zones of d^Z are those of d, together with those in Z: $Z(d^Z) = Z(d) \cup Z = Z$.
2. The shaded zones of d^Z are those of d, together with those in Z that are not in d: $Z^*(d^Z) = Z^*(d) \cup (Z - Z(d))$.
3. The spiders match: $\mathrm{SI}(d^Z) = \mathrm{SI}(d)$.

If $d = \perp$, then the *zone diagram associated with* d is $\perp$.

Let $D = D_1 \diamond D_2$ ($\diamond \in \{\vee, \wedge\}$) be any diagram such that, for each d in comp(D) $Z \supseteq Z(d)$ or $d = \perp$. Given Z, the *zone diagram associated with* D is $D^Z = D_1^Z \diamond D_2^Z$ where D_1^Z and D_2^Z are the zone diagrams associated with D_1 and D_2 respectively.

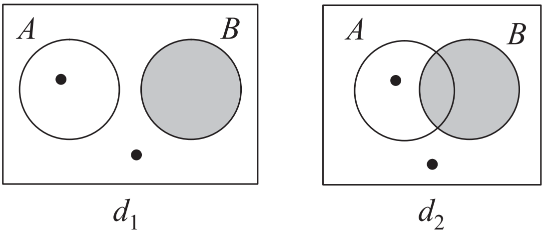

Figure 33: A diagram with an associated zone diagram.

EXAMPLE 6.4. The zone set $Z = \{(\emptyset, \{A, B\}), (\{A\}, \{B\}), (\{B\}, \{A\}), (\{A, B\}, \emptyset)\}$ is a finite subset of $\mathcal{Z}$. Given Z, the diagram d_1 in Figure 33 has associated zone diagram d_2. The diagram d_2 can be obtained from d_1 by introducing zone $(\{A, B\}, \emptyset)$ to d_1.

Given a zone set, the associated zone diagrams are unique.

THEOREM 6.3. *Let Z be a finite subset of $\mathcal{Z}$ such that*

$$\forall (a_1, b_1), (a_2, b_2) \in Z \bullet a_1 \cup b_1 = a_2 \cup b_2.$$

Let D be any diagram such that for each d in comp(D), *$Z \supseteq Z(d)$ or $d = \perp$. Given Z, let D^Z be the zone diagram associated with D. Then D is syntactically equivalent to D^Z.*

Sketch of the proof. Suppose that D is a unitary diagram. If $D = \perp$, then $D^Z = \perp$ and $D \equiv_\vdash D^Z$. Alternatively, $D \neq \perp$. Apply Rule 2 (introduction of a shaded zone) to D, introducing all zones in $Z - Z(D)$ to give D^Z. Then $D \equiv_\vdash D^Z$. The result then follows by induction on the depth of D in the inductive construction. □

6.4. *Associated α-diagrams*

We are aiming to replace any spider diagram by a syntactically equivalent disjunction of unitary α-diagrams. To do so, we must apply the combining rule, which operates on unitary α-diagrams with the same zone sets. So far, we have seen that any diagram can be replaced by another diagram where each of the unitary parts have the same zone sets, or are $\perp$. Thus

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

the next step that we take is to replace each unitary part by a disjunction of α-diagrams. Any spider diagram that is not an α-diagram may be transformed into an α-diagram by splitting all the spiders.

Example 6.5. The diagram d in Figure 34 is syntactically equivalent to the α-diagram $d_1 \vee d_2$. This equivalence is obtained by a single application of Rule 3, splitting spiders.

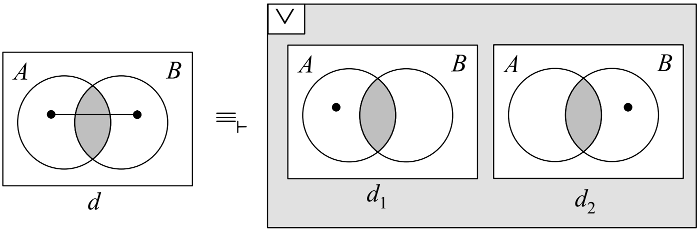

Figure 34: Syntactically equivalent diagrams.

We wish to define, for any spider diagram D, its 'associated α-diagram' which is an α-diagram that is syntactically equivalent to D. To describe the construction of such an α-diagram, we introduce the notion of an α-subdiagram of a unitary diagram. Informally, we can obtain an α-subdiagram of a diagram d by erasing, for each spider in d, all of its feet except one.

Definition 6.4. Let d be a unitary diagram and d' a unitary α-diagram. If $d \neq \perp$, $Z(d) = Z(d')$, $Z^*(d) = Z^*(d')$ and there exists a bijection $\sigma : S(d) \longrightarrow S(d')$ such that, for all spiders s in d,

$$\eta_{d'}(\sigma(s)) \subseteq \eta_d(s),$$

then d' is an *α-subdiagram of d*, denoted $d' \sqsubseteq_\alpha d$. If $d = \perp$, then the only α-subdiagram of d is $\perp$.

Example 6.6. In Figure 35, the diagram d' is an α-subdiagram of d. Furthermore, by applying Rule 15 (adding feet to a spider) twice, it is clear that d is obtainable from d'.

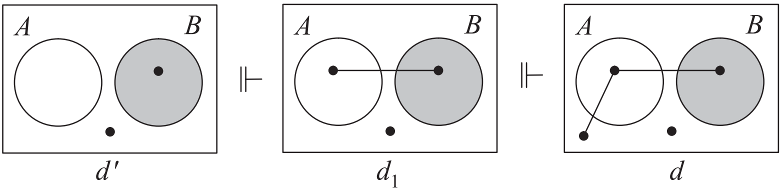

Figure 35: An α-subdiagram of d transformed into d.

Lemma 6.1. *Let d be a unitary diagram, and let d' be a unitary α-diagram. If $d' \sqsubseteq_\alpha d$, then $d' \vdash d$.* □

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

DEFINITION 6.5. Let D be a spider diagram.

1. If D is a unitary diagram, let $\mathcal{D}_\alpha^D = \{d' \in \mathcal{D}_0 : d' \sqsubseteq_\alpha D\}$ be the set of all α-subdiagrams of D. The *α-diagram associated with D*, ${}^\alpha D$, is the disjunction of all of its α-subdiagrams:

$$ {}^\alpha D = \bigvee_{d' \in \mathcal{D}_\alpha^D} d'. $$

2. If $D = D_1 \diamond D_2$ for some $D_1, D_2 \in \mathcal{D}$ and $\diamond$ is $\wedge$ or $\vee$, then the *α-diagram associated with D*, ${}^\alpha D$, is

$$ {}^\alpha D = {}^\alpha D_1 \diamond {}^\alpha D_2, $$

where ${}^\alpha D_1$ and ${}^\alpha D_2$ are the α-diagrams associated with D_1 and D_2 respectively.

EXAMPLE 6.7. In Figure 36, the diagram d has associated α-diagram ${}^\alpha D = d_1 \vee d_2 \vee d_4$ which is syntactically equivalent to d. This equivalence is established by splitting the spiders in d, and then using idempotency to remove the repeated diagram.

EXAMPLE 6.8. Similarly, the compound diagram D in Figure 37 has the associated α-diagram ${}^\alpha D$.

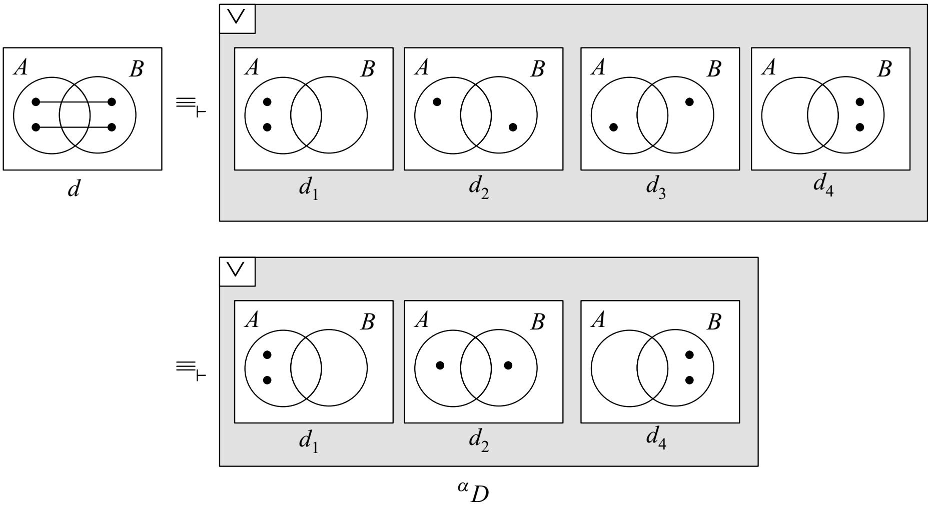

Figure 36: A unitary diagram with its associated α-diagram.

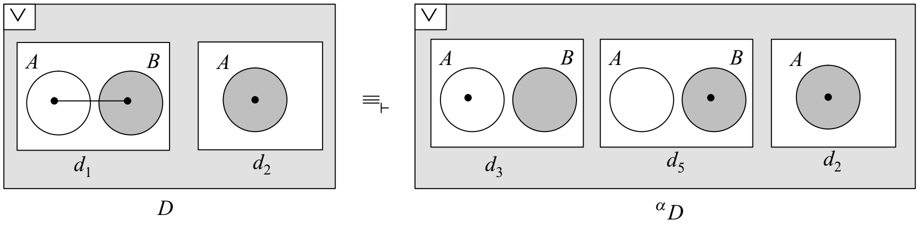

Figure 37: A diagram with its associated α-diagram.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

We wish to show that any diagram is syntactically equivalent to its associated α-diagram.

LEMMA 6.2. *Let d be a unitary diagram and D the α-diagram obtained from d by applying Rules 3 (splitting spiders) and 12 (simplification of $\vee$) repeatedly to d. Then the α-diagram associated with d is*

$$^{\alpha}d = \bigvee_{d' \in \mathrm{comp}(D)} d'. \qquad \square$$

THEOREM 6.4. *Let D be a spider diagram, and let $^{\alpha}D$ be the α-diagram associated with D. Then $D \equiv_{\vdash} {}^{\alpha}D$.* $\square$

6.5. *Combining α-diagrams*

The next stage in our completeness proof strategy is to remove all the conjuncts; for this we use the combining rule.

DEFINITION 6.6. Let D be an α-diagram such that each pair of unitary components, $d_1, d_2 \in \mathrm{comp}(D)$, satisfies $Z(d_1) = Z(d_2)$ or $d_1 = \bot$ or $d_2 = \bot$. The *combined diagram associated with D*, denoted D^*, is defined as follows.

1. If D is a unitary diagram, then $D^* = D$.
2. If $D = D_1 \vee D_2$, for some D_1 and D_2, then

$$D^* = D_1^* \vee D_2^*,$$

where D_1^* and D_2^* are combined diagrams associated with D_1 and D_2 respectively.

3. If $D = D_1 \wedge D_2$, then

$$D^* = \bigvee_{\substack{d_i \in \mathrm{comp}(D_1^*) \\ d_j \in \mathrm{comp}(D_2^*)}} d_i * d_j.$$

We now prove that D is syntactically equivalent to D^*.

THEOREM 6.5. *Let D be an α-diagram such that each pair of unitary components $d_1, d_2 \in \mathrm{comp}(D)$ satisfies $Z(d_1) = Z(d_2)$, or $d_1 = \bot$ or $d_2 = \bot$. Then $D \equiv_{\vdash} D^*$.*

Proof. The proof is achieved by induction on the depth of D in the inductive construction. If $D \in \mathcal{D}_0$, then $D = D^*$ and the base case holds.

For any α-diagram $D \in \mathcal{D}_n$ such that each pair of unitary components $d_1, d_2 \in \mathrm{comp}(D)$ satisfies $Z(d_1) = Z(d_2)$ or $d_1 = \bot$ or $d_2 = \bot$, assume that it is the case that $D \equiv_{\vdash} D^*$, where D^* is the combined diagram associated with D. Let $D_1 \in \mathcal{D}_{n+1} - \mathcal{D}_n$ be an α-diagram such that each pair of unitary components $d_1, d_2 \in \mathrm{comp}(D_1)$ satisfies $Z(d_1) = Z(d_2)$ or $d_1 = \bot$ or $d_2 = \bot$. Then $D_1 = D_2 \diamond D_3$ for some $\diamond \in \{\vee, \wedge\}$ and $D_2, D_3 \in \mathcal{D}_n$ for which each pair of unitary components $d_1, d_2 \in \mathrm{comp}(D_2) \cup \mathrm{comp}(D_3)$ satisfies $Z(d_1) = Z(d_2)$ or $d_1 = \bot$ or $d_2 = \bot$. Let D_1^* be the combined diagram associated with D_1.

Suppose firstly that $\diamond = \vee$. Then $D_1^* = D_2^* \vee D_3^*$, where D_2^* and D_3^* are the combined diagrams associated with D_2 and D_3 respectively. By assumption, $D_2 \equiv_{\vdash} D_2^*$ and $D_3 \equiv_{\vdash} D_3^*$. Therefore we have

$$\begin{aligned} D_1 = &\ D_2 \vee D_3 & \\ \equiv_{\vdash} &\ D_2^* \vee D_3 & \text{by Corollary 4.1 (rule of replacement)} \\ \equiv_{\vdash} &\ D_2^* \vee D_3^* & \text{by Corollary 4.1 (rule of replacement)} \\ = &\ D_1^*. & \end{aligned}$$

Hence $D_1 \equiv_{\vdash} D_1^*$.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

Alternatively, $\diamond = \wedge$. Then

$$D^* = \bigvee_{\substack{d_i \in \text{comp}(D_1^*) \\ d_j \in \text{comp}(D_2^*)}} d_i * d_j.$$

Since D_2^* and D_3^* are combined diagrams, it must be true that

$$D_2^* = \bigvee_{1 \leqslant i \leqslant n} d_{2,i} \qquad \text{and} \qquad D_3^* = \bigvee_{1 \leqslant j \leqslant m} d_{3,j},$$

where each $d_{2,i}$ $(1 \leqslant i \leqslant n)$ and $d_{3,j}$ $(1 \leqslant j \leqslant m)$ is a unitary diagram. Using Rules 10 (distributivity of $\vee$) and 12 (simplification of $\vee$), we can show that

$$D_2^* \wedge D_3^* = \Bigg(\bigvee_{1 \leqslant i \leqslant n} d_{2,i} \Bigg) \wedge \Bigg(\bigvee_{1 \leqslant j \leqslant m} d_{3,j} \Bigg) \equiv_{\vdash} \bigvee_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}} (d_{2,i} \wedge d_{3,j}).$$

Now, for each $d_{2,i}$ and $d_{3,j}$, we have $d_{2,i} * d_{3,j} \equiv_{\vdash} d_{2,i} \wedge d_{3,j}$, by Rule 5 (combining). Therefore

$$D_2^* \wedge D_3^* \equiv_{\vdash} \bigvee_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}} (d_{2,i} * d_{3,j}) = D_1^* \qquad \text{by Rule 12 (simplification of } \vee).$$

We have shown that

$$\begin{aligned} D_1 = {} & D_2 \wedge D_3 \\ \equiv_{\vdash} {} & D_2^* \wedge D_3^* \qquad \text{by Corollary 4.1 (rule of replacement)} \\ \equiv_{\vdash} {} & D_1^*. \end{aligned}$$

Hence $D_1 \equiv_{\vdash} D_1^*$. □

6.6. *Extended diagrams*

Suppose that we have a unitary α-diagram d and a disjunction of unitary α-diagrams, D, such that $Z(d) = Z(d_i)$ or $d_i = \bot$ for each $d_i \in \text{comp}(D)$. We are aiming to find an α-diagram which we call the *extended diagram associated with d in the context of D*, namely $\text{ext}(d, D)$, that is syntactically equivalent to d. The diagram $\text{ext}(d, D)$ will have the property that, if a unitary component of $\text{ext}(d, D)$, say ${}^e d$, satisfies ${}^e d \vDash D$, then there exists a unitary component of D, say d_i, satisfying ${}^e d \vDash d_i$.

EXAMPLE 6.9. In Figure 38, the diagram D is a semantic consequence of d but no unitary component of D is semantically entailed by d; that is, $d \nvDash d_1$, $d \nvDash d_2$ and $d \nvDash d_3$. The diagram $\text{ext}(d, D)$ can be obtained from d (and vice versa) by applying Rules 4 (excluded middle) and 12 (simplification of $\wedge$). The spiders and shading introduced to d to obtain $\text{ext}(d, D)$ are determined by D. For example, consider the outside zone $(\varnothing, \{A\})$. This zone is shaded and contains two spiders in d_3, and no other unitary component of D contains more than two spiders in this zone. In $\text{ext}(d, D)$, this zone contains one, two or three spiders in any unitary component. The process of constructing $\text{ext}(d, D)$ will be described in Definitions 6.7 and 6.8 below.

Note that we have

$$d_1' \vDash d_1, \quad d_2' \vDash d_2, \quad d_3' \vDash d_1, \quad d_3' \vDash d_3, \quad d_4' \vDash d_1, \quad d_5' \vDash d_2, \quad d_6' \vDash d_2 \quad \text{and} \quad d_6' \vDash d_3$$

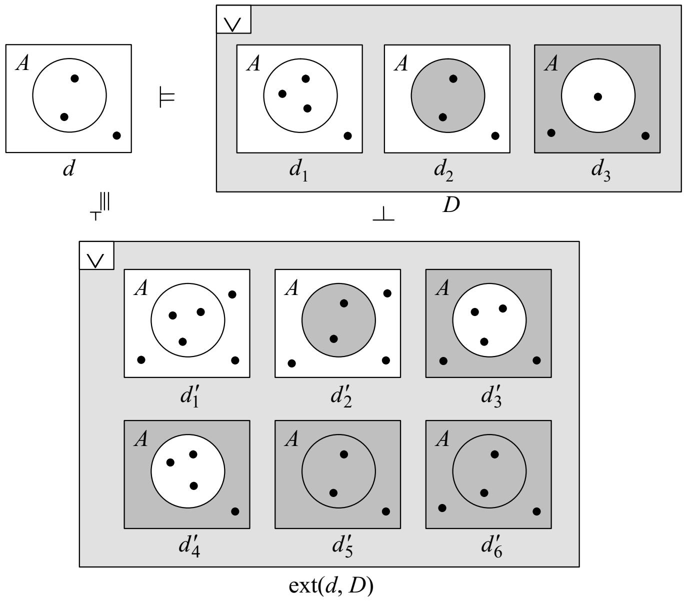

Figure 38: An α-diagram and an extended diagram.

so, for each unitary component d'_i of $\mathrm{ext}(d, D)$, there exists a unitary component d_j of D such that $d'_i \vDash d_j$. By Theorem 6.1, $d'_i \vdash d_j$. Thus

$$d'_1 \vee d'_3 \vee d'_4 \vdash d_1 \qquad \text{and} \qquad d'_2 \vee d'_5 \vee d'_6 \vdash d_2,$$

by the rule of construction. Therefore,

$$\mathrm{ext}(d, D) = d'_1 \vee d'_3 \vee d'_4 \vee d'_2 \vee d'_5 \vee d'_6 \vdash d_1 \vee d_2,$$

by the rule of replacement. By Rule 6 (connecting a diagram), $d_1 \vee d_2 \vdash D$, and by transitivity, $\mathrm{ext}(d, D) \vdash D$. Therefore $d \vdash D$, since $d \equiv_{\vdash} \mathrm{ext}(d, D)$.

In general, the diagram $\mathrm{ext}(d, D)$ will be constructed by taking copies of d and adding shading and spiders, as specified below. The unitary components of $\mathrm{ext}(d, D)$ are called *extended unitary components associated with d*, which we now define.

Definition 6.7. Let d ($\neq \bot$) be a unitary α-diagram, and let D be an α-diagram. Then, given D, a unitary α-diagram ${}^e d$ is an *extended unitary component associated with d*, denoted by $d \sqsubseteq_e^D {}^e d$, if and only if the following six conditions are satisfied.

1. The diagrams d and ${}^e d$ have the same zones: $Z(d) = Z({}^e d)$.
2. All shading in d occurs in ${}^e d$: $Z^*(d) \subseteq Z^*({}^e d)$.
3. All spiders in d occur in ${}^e d$: $S(d) \subseteq S({}^e d)$.
4. If zone z is shaded in d, then the spiders match in d and ${}^e d$: $\forall z \in Z^*(d) \bullet S(\{z\}, d) = S(\{z\}, {}^e d)$.
5. If zone z is not shaded in d but is shaded in some unitary component of D, and the number, m say, of spiders that z contains in d is at most the number that z contains in any unitary component of D in which z is shaded, then the following statements hold.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

(i) If z is shaded in ${}^e d$, then z contains at most m spiders in ${}^e d$; and
(ii) if z is not shaded in ${}^e d$, then z contains $m+1$ spiders in ${}^e d$.

More formally, we have:

$$\forall z \in Z(d) - Z^*(d) \bullet \left(z \in \bigcup_{d_i \in \text{comp}(D)} Z^*(d_i) \ \wedge \ S(\{z\}, d) \subseteq \bigcup_{\substack{d_i \in \text{comp}(D) \\ z \in Z^*(d_i)}} S(\{z\}, d_i) \right)$$
$$\Longrightarrow \left(\left(z \in Z^*({}^e d) \wedge S(\{z\}, {}^e d) \subseteq \bigcup_{\substack{d_i \in \text{comp}(D) \\ z \in Z^*(d_i)}} S(\{z\}, d_i) \right) \right.$$
$$\left. \vee \left(z \in Z({}^e d) - Z^*({}^e d) \wedge |S(\{z\}, {}^e d)| = \left| \bigcup_{\substack{d_i \in \text{comp}(D) \\ z \in Z^*(d_i)}} S(\{z\}, d_i) \right| + 1 \right) \right).$$

6. If an unshaded zone z in d is not shaded in any unitary component of D, or if z contains more spiders in d than any shaded occurrence of z in D, then z is not shaded in ${}^e d$ and z contains the same number of spiders in ${}^e d$ as in d. More formally, we have:

$$\forall z \in Z(d) - Z^*(d) \bullet \left(z \notin \bigcup_{d_i \in \text{comp}(D)} Z^*(d_i) \vee S(\{z\}, d) \supset \bigcup_{\substack{d_i \in \text{comp}(D) \\ z \in Z^*(d_i)}} S(\{z\}, d_i) \right)$$
$$\Longrightarrow \left(z \in Z({}^e d) - Z^*({}^e d) \wedge S(\{z\}, {}^e d) = S(\{z\}, d) \right).$$

If $d = \bot$, then the *extended unitary component associated with* d is $\bot$.

DEFINITION 6.8. Let d be a unitary α-diagram, and let D be a disjunction of unitary α-diagrams such that $Z(d) = Z(d_i)$ or $d_i = \bot$ for each $d_i \in \text{comp}(D)$. Given D, let $\mathcal{D}_e^d$ be the set of all extended unitary components associated with d:

$$\mathcal{D}_e^d = \left\{ d' \in \mathcal{D}_0 : d \sqsubseteq_e^D d' \right\}.$$

Then the diagram

$$\text{ext}(d, D) = \bigvee_{d' \in \mathcal{D}_e^d} d'$$

is the *extended diagram associated with* d *in the context of* D.

EXAMPLE 6.10. In Figure 38, each d_i' $(i = 1, \ldots, 6)$ is an extended unitary component associated with d, given D. Indeed, all such extended components ${}^e d$ are present, so $\text{ext}(d, D)$ is the extended diagram associated with d in the context of D.

THEOREM 6.6. *Let d be a unitary α-diagram, and let D be a disjunction of unitary α-diagrams such that $Z(d) = Z(d_i)$ or $d_i = \bot$ for each $d_i \in$ comp(D). Then d is syntactically equivalent to* ext(d, D)*, the extended diagram associated with d in the context of D:*

$$d \equiv_\vdash \text{ext}(d, D).$$

Sketch of the proof. The proof follows by repeated application of Rules 4 (excluded middle) and 12 (simplification of $\vee$) to d in the case where $d \neq \bot$. When $d = \bot$, the result follows immediately. □

6.7. *Completeness theorem*

The next result is the final prerequisite for our proof of completeness.

THEOREM 6.7. *Let d be a unitary α-diagram, and let D be a disjunction of unitary α-diagrams such that $Z(d) = Z(d_i)$ or $d_i = \bot$ for each $d_i \in \text{comp}(D)$. Given D, let ${}^e d \in \mathcal{D}_e^d$. If ${}^e d \vDash D$, then there exists a unitary component of D, say d_i, such that ${}^e d \vDash d_i$:*

$$ {}^e d \vDash D \Longrightarrow \exists d_i \in \text{comp}(D) \bullet {}^e d \vDash d_i . $$

Proof. The proof is shown by contradiction. Assume that ${}^e d \vDash D$, but that there is no $d_i \in \text{comp}(D)$ for which ${}^e d \vDash d_i$. By Theorem 6.1, for each d_i one of the following alternative statements holds:

(i) $\exists z \in Z^*(d_i) \bullet z \notin Z^*({}^e d) \vee S(\{z\}, d_i) \neq S(\{z\}, {}^e d)$ or

(ii) $\exists z \in Z(d_i) \bullet S(\{z\}, {}^e d) \subset S(\{z\}, d_i)$.

Let $m = (\mathbf{U}, \Psi)$ be a model for ${}^e d$ with the property that, for each zone in $Z({}^e d)$, $|\Psi(z)| = |S(\{z\}, {}^e d)|$. We show that m does not satisfy any d_i in D, and therefore ${}^e d \nvDash D$, giving a contradiction.

Suppose that statement (i) is true. Firstly, consider the case where $z \notin Z^*({}^e d)$; then $z \in Z({}^e d) - Z^*({}^e d)$. So $z \in Z(d) - Z^*(d)$ and, by the definition of ${}^e d$,

$$ S(\{z\}, {}^e d) \supset \bigcup_{\substack{d_j \in \text{comp}(D) \\ z \in Z^*(d_j)}} S(\{z\}, d_j) \supseteq S(\{z\}, d_i). $$

Now $|\Psi(z)| = |S(\{z\}, {}^e d)| > |S(\{z\}, d_i)| = |T(\{z\}, d_i)|$, and it follows that the shading condition fails for d_i.

Consider now the case where $S(\{z\}, d_i) \neq S(\{z\}, {}^e d)$. Then either the shading condition fails for d_i (as above), or the distinct spiders condition fails for d_i. Hence $m \nvDash d_i$.

Suppose that statement (ii) is true. That is,

$$ \exists z \in Z(d_i) \bullet S(\{z\}, {}^e d) \subset S(\{z\}, d_i). $$

Then $|\Psi(z)| = |S(\{z\}, {}^e d)| < |S(\{z\}, d_i)|$, and thus the distinct spiders condition fails for d_i. Hence $m \nvDash d_i$.

Thus m does not satisfy any unitary component of D, and by the definition of the semantics predicate for D, m does not satisfy D. This contradicts the assumption that ${}^e d \vDash D$. Hence if ${}^e d \vDash D$, then there exists $d_i \in \text{comp}(D)$ such that ${}^e d \vDash d_i$. □

THEOREM 6.8 (COMPLETENESS). *Let D_1 and D_2 be spider diagrams. If $D_1 \vDash D_2$, then $D_1 \vdash D_2$.*

Proof. Let D_1 and D_2 be spider diagrams such that $D_1 \vDash D_2$. Given $L = L(D_1) \cup L(D_2)$, let D_1^L and D_2^L be the contour diagrams associated with D_1 and D_2 respectively. Given

$$ Z = \bigcup_{d \in \text{comp}(D_1^L) \cup \text{comp}(D_2^L)} Z(d), $$

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

let D_1^Z and D_2^Z be zone diagrams associated with D_1^L and D_2^L respectively. Let ${}^{\alpha}D_1^Z$ and ${}^{\alpha}D_2^Z$ be α-diagrams associated with D_1^L and D_2^L respectively. Let D_1^* and D_2^* be combined diagrams associated with ${}^{\alpha}D_1^Z$ and ${}^{\alpha}D_2^Z$ respectively. Then

$$\begin{aligned} D_1 &\equiv_{\vdash} D_1^L && \text{by Theorem 6.2} \\ &\equiv_{\vdash} D_1^Z && \text{by Theorem 6.3} \\ &\equiv_{\vdash} {}^{\alpha}D_1^Z && \text{by Theorem 6.4} \\ &\equiv_{\vdash} D_1^* && \text{by Theorem 6.5.} \end{aligned}$$

Hence, by transitivity,

$$D_1 \equiv_{\vdash} D_1^*. \tag{2}$$

Similarly,

$$D_2 \equiv_{\vdash} D_2^*. \tag{3}$$

By the soundness theorem,

$$D_1^* \vDash D_1 \wedge D_2 \vDash D_2^*,$$

and again by transitivity,

$$D_1^* \vDash D_2^*.$$

Now D_1^* and D_2^* are disjunctions of unitary diagrams, and so

$$D_1^* = \bigvee_{1 \leqslant i \leqslant n} d_{1,i} \qquad \text{and} \qquad D_2^* = \bigvee_{1 \leqslant j \leqslant l} d_{2,j},$$

where each $d_{1,i}$ $(1 \leqslant i \leqslant n)$ and $d_{2,j}$ $(1 \leqslant j \leqslant l)$ is a unitary α-diagram. The semantics predicate for D_1^* is

$$\bigvee_{1 \leqslant i \leqslant n} P_{d_{1,i}}(m),$$

where $m = (\mathbf{U}, \Psi)$ is a set-assignment to regions. Therefore, for all $d_{1,i}$ $(1 \leqslant i \leqslant n)$ we have

$$d_{1,i} \vDash D_1^* \tag{4}$$

Let $\text{ext}(d_{1,i}, D_2^*)$ be the extended diagram associated with $d_{1,i}$ in the context of D_2^*. By Theorem 6.6, we have

$$d_{1,i} \equiv_{\vdash} \text{ext}(d_{1,i}, D_2^*),$$

and by the soundness theorem,

$$d_{1,i} \equiv_{\vDash} \text{ext}(d_{1,i}, D_2^*).$$

Now we have $\text{ext}(d_{1,i}, D_2^*) \vDash d_{1,i}$, $d_{1,i} \vDash D_1^*$ and $D_1^* \vDash D_2^*$, and so by transitivity

$$\text{ext}(d_{1,i}, D_2^*) \vDash D_2^*.$$

Now

$$\text{ext}(d_{1,i}, D_2^*) = \bigvee_{1 \leqslant p \leqslant k} {}^{e}d_{1,p}$$

for some unitary diagrams ${}^{e}d_{1,p}$ $(1 \leqslant p \leqslant k)$.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

Similarly to (4), we deduce that for all $d_{1,p}$ $(1 \leqslant p \leqslant k)$, we have

$$^{e}d_{1,p} \vDash D_2^*.$$

Therefore, by Theorem 6.7,

$$\forall\, {}^{e}d_{1,p}(1 \leqslant p \leqslant k)\, \exists d_{2,j}(1 \leqslant j \leqslant m) \bullet {}^{e}d_{1,p} \vDash d_{2,j}.$$

By Theorem 6.1,

$$\forall\, {}^{e}d_{1,p}(1 \leqslant p \leqslant k)\, \exists d_{2,j}(1 \leqslant j \leqslant m) \bullet {}^{e}d_{1,p} \vdash d_{2,j}.$$

By Rule 6 (connecting a diagram), for all ${}^{e}d_{i,p}$ $(1 \leqslant p \leqslant k)$ we have

$$^{e}d_{1,p} \vdash D_2^*.$$

Hence, by Theorem 4.2 (rule of construction), for all $d_{1,i}$ $(1 \leqslant i \leqslant n)$, we have

$$\mathrm{ext}(d_{1,i}, D_2^*) \vdash D_2^*.$$

Since, by Theorem 6.6, $d_{1,i} \vdash \mathrm{ext}(d_{1,i}, D_2^*)$, by transitivity we obtain for all $d_{1,i}$ $(1 \leqslant i \leqslant n)$:

$$d_{1,i} \vdash D_2^*.$$

By Theorem 4.2 (rule of construction), we have

$$D_1^* \vdash D_2^* \tag{5}$$

Now

$$\begin{aligned} D_1 &\vdash D_1^* && \text{by (2)} \\ &\vdash D_2^* && \text{by (5)} \\ &\vdash D_2 && \text{by (3).} \end{aligned}$$

Finally, transitivity gives $D_1 \vdash D_2$. Hence

$$D_1 \vDash D_2 \Longrightarrow D_1 \vdash D_2,$$

as required. □

We have seen that the spider diagram system is both sound and complete. An immediate consequence of the completeness proof strategy is that the system is also decidable.

THEOREM 6.9 (DECIDABILITY). *Let D_1 and D_2 be spider diagrams. There is an algorithm that determines whether $D_1 \vdash D_2$.*

Sketch of the proof. Given D_1 and D_2, apply the completeness proof algorithm to both D_1 and D_2, giving D_1^* and D_2^* as above. For each unitary part, d_1, of D_1^*, obtain $\mathrm{ext}(d_1, D_2^*)$. Then either each unitary part of $\mathrm{ext}(d_1, D_2^*)$ syntactically entails a unitary part of D_2^*, or there exists a unitary part of $\mathrm{ext}(d_1, D_2^*)$ that does not syntactically entail a unitary part of D_2^*. In the latter case, we can deduce that $D_1 \nvdash D_2$. □

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

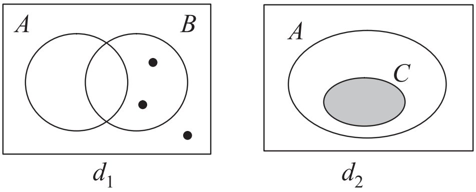

Figure 39: From diagrams to sentences.

7. *Expressiveness*

A natural question to ask is: "What is the formal expressiveness of spider diagrams?" Shin showed that her Venn II system is equivalent in expressive power to first-order monadic logic without equality. Spider diagrams are more expressive.

THEOREM 7.1 (EXPRESSIVENESS). *The language of spider diagrams is equivalent in expressive power to first-order monadic logic with equality.*

Proof. This result is proved formally in [42]. The mapping from diagrams to sentences is reasonably straightforward. □

EXAMPLE 7.1. The diagram d_1 in Figure 39 contains three spiders, one outside both A and B and the other two inside B and outside A. It is expressively equivalent to the sentence

$$\exists x_1 \, (\neg A(x_1) \wedge \neg B(x_1)) \wedge \exists x_1 \exists x_2 \, (B(x_1) \wedge B(x_2) \wedge \neg A(x_1) \wedge \neg A(x_2) \wedge x_1 \neq x_2).$$

The diagram d_2 expresses the information that no elements can be in C and not in A (due to the missing zone), and no elements can be in both A and C (due to the shading), and is expressively equivalent to the sentence

$$\forall x_1 \, \neg(C(x_1) \wedge \neg A(x_1)) \wedge \forall x_1 \, \neg(A(x_1) \wedge C(x_1)).$$

The diagram $d_1 \vee d_2$ is expressively equivalent to the disjunction of the sentences given for d_1 and d_2.

The mapping from sentences to diagrams is more challenging. Shin's approach for the Venn II system does not extend to spider diagrams. She converts a sentence to prenex normal form, and then syntactically manipulates this to remove nested quantifiers; a diagram can then be drawn for each of the simple parts of the resulting formula. This approach does not extend to the case where equality is allowed because '$=$' is a dyadic predicate, and so nesting of quantifiers cannot necessarily be removed. We take a different approach, based on a classic result of Dreben and Goldfarb [1, pp. 209–210]. To establish the existence of a diagram that is expressively equivalent to a sentence, we consider models for that sentence. In [42], it is shown that for every sentence S, there exists a finite set of models that can be used to classify all the models for S. Each classifying model has a finite domain, and can be used to construct a diagram. The disjunction of all such diagrams is expressively equivalent to S. The idea is illustrated in the following example.

EXAMPLE 7.2. Let S be the sentence $\exists x_1 A(x_1) \vee \forall x_1 A(x_1)$. There are four classifying models for S that give rise to the diagrams d_1, d_2, d_3 and d_4 in Figure 40. The diagram $d_1 \vee d_2 \vee d_3 \vee d_4$ is expressively equivalent to S. This is not the 'natural' diagram that one would associate with S.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

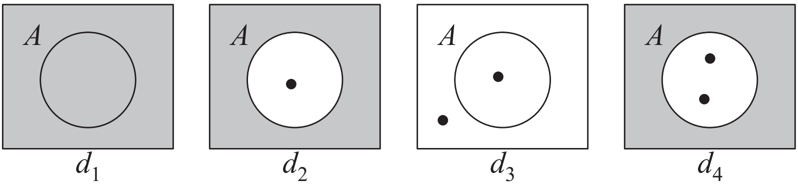

Figure 40: Constructing diagrams from models.

8. *Conclusion and further work*

Building on the traditions established by Euler, Venn and Peirce, and on more recent seminal work by Sun-Joo Shin, we have developed a sound and complete reasoning system that is purely diagrammatic. The spider diagram system presented in this paper is expressively equivalent to first-order monadic logic with equality, making it more expressive than the diagrammatic systems developed by Shin [37] and Hammer [19].

This work is part of an ongoing project to develop formal visual notations and associated tool support. The effective use of diagrammatic notations for practical applications requires computer-aided support tools. Currently, tools have been developed to generate concrete diagrams from an abstract description [9], and for laying-out the resulting diagrams aesthetically [15, 32, 35]. An automated theorem-prover has been developed for spider diagrams [10], together with heuristics for generating readable proofs [13, 14].

The main intended application area for this work is the modelling and specification of software systems. The focus of further work is on various systems of *constraint diagrams* [4, 6, 7, 5, 16, 18, 28, 39, 40], which extend spider diagrams by incorporating (explicit) universal quantification and relational navigation. All of the spider diagram reasoning rules given in this paper extend to some restricted fragments of the constraint diagram notation, presented in [38, 39, 40]. Along with further reasoning rules, these restricted fragments are shown to be sound and complete. The strategy for proving completeness in our spider diagram system extends to these restricted systems, although the details of the proof become considerably more complex.

The syntax and semantics of the full constraint-diagram language have been formalized [7]. Work is ongoing to develop formal reasoning rules for the full constraint-diagram notation [3].

Constraint diagrams were designed to complement the visual notations that comprise the Unified Modeling Language (UML) [33], which are used in modelling software systems. Constraint diagrams provide a notation for expressing logical constraints, such as invariants and operation pre-conditions and post-conditions, which, in UML, are expressed in the Object Constraint Language (OCL) [45]. The OCL is essentially a stylized text-based version of first-order predicate logic, so constraint diagrams provide an alternative language that is more in keeping with the other diagrammatic notations within the UML. There is ample informal evidence – the strong take-up of the mainly diagrammatic UML as the standard for software modelling, and the relatively poor take-up of traditional formal methods by the software industry – that software engineers prefer diagrammatic notations to traditional mathematical notations.

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press

References

1. B. Dreben and D. Goldfarb, *The decision problem. Solvable classes of quantificational formulas* (Addison Wesley, 1979). 190
2. L. Euler, *Lettres a une princesse d'Allemagne sur divers sujets de physique et de philosophie*, vol. 2 (1761) Letters 102–108. 145
3. A. Fish and J. Flower, 'Investigating reasoning with constraint diagrams', *Proc. Visual Languages and Formal Methods* 2004, Electron. Notes Theor. Comput. Sci. 127 (Elsevier, 2005) 53–69. 191
4. A. Fish and J. Howse, 'Computing reading trees for constraint diagrams', *Proc. AGTIVE* 2003 (Applications of Graph Transformations with Industrial Relevance, Charlottesville, Virginia) (Springer, 2003) 260–274. 191
5. A. Fish and J. Howse, 'Towards a default reading for constraint diagrams', *Proc. Diagrams* 2004 (International Conference on the Theory and Application of Diagrams, Cambridge), Lecture Notes in Artificial Intelligence 2980 (Springer, 2004) 51–65. 191
6. A. Fish, J. Flower and J. Howse, 'A reading algorithm for constraint diagrams', *Proc. IEEE Symposium on Human Centric Computing Languages and Environments* (Auckland, New Zealand, October 2003) (IEEE Computer Society Press, 2003) 161–168. 191
7. A. Fish, J. Flower and J. Howse, 'The semantics of augmented constraint diagrams', *J. of Visual Languages and Computing*, to appear 2005. 191
8. M. Fitting, *First order logic and automated theorem proving* (Springer, 1996). 161
9. J. Flower and J. Howse, 'Generating Euler diagrams', *Proc. Diagrams* 2002 (International Conference on the Theory and Application of Diagrams), Lecture Notes in Artificial Intelligence 2317 (Springer, 2002) 61–75. 152, 154, 155, 156, 191
10. J. Flower and G. Stapleton, 'Automated theorem proving with spider diagrams', *Proc. CATS* 2004 (Computing: The Australasian Theory Symposium, Dunedin, New Zealand), Electron. Notes Theor. Comput. Sci. 91 (Science Direct, 2004) 116–132. 191
11. J. Flower, J. Howse and J. Taylor, 'Nesting in Euler Diagrams: syntax, semantics and construction', *J. Software and Systems Modeling* 3 (2004) 55–67. 159
12. J. Flower, J. Howse, J. Taylor and S. Kent, 'A visual framework for modelling with heterogeneous notations', *Proc. HCC* 2002 (IEEE Symposium on Human Centric Computing Languages and Environments, Arlington) (IEEE Computer Society Press, 2002) 71–73. 156
13. J. Flower, J. Masthoff and G. Stapleton, 'Generating readable proofs: a heuristic approach to theorem proving with spider diagrams', *Proc. Diagrams* 2004 (International Conference on the Theory and Application of Diagrams, Cambridge), Lecture Notes in Artificial Intelligence 2980 (Springer, 2004) 166–181. 191
14. J. Flower, J. Masthoff and G. Stapleton, 'Generating proofs with spider diagrams using heuristics', *Proc. VLC* 2004 (10th International Conference on Distributed Multimedia Systems, International Workshop on Visual Languages and Computing, San Francisco) (Knowledge Systems Institute, 2004) 279–285. 191
15. J. Flower, P. Rodgers and P. Mutton, 'Layout metrics for Euler diagrams', *Proc. IV* 2003 (7th International Conference on Information Visualisation, London) (IEEE Computer Society Press, 2003) 272–280. 191

16. J. GIL, J. HOWSE and S. KENT, 'Constraint diagrams: a step beyond UML', *Proc. TOOLS USA* 1999 (Santa Barbara, California) (IEEE Computer Society Press, 1999) 453–463. 191

17. J. GIL, J. HOWSE and S. KENT, 'Formalising spider diagrams', *Proc. VL* 1999 (IEEE Symposium on Visual Languages) (IEEE Computer Society Press, 1999) 130–137. 146

18. J. GIL, J. HOWSE and S. KENT, 'Towards a formalization of constraint diagrams', *Proc. HCC* 2001 (IEEE Symposium on Human-Centric Computing, Stresa) (IEEE Computer Society Press, 2001) 72–79. 191

19. E. HAMMER, *Logic and visual information* (CSLI Publ., 1995). 146, 152, 163, 191

20. D. HAREL, 'On visual formalisms', *Diagrammatic reasoning* (ed. J. Glasgow, N. H. Narayan and B. Chandrasekaran, MIT Press, 1998) 235–271.

21. J. HOWSE, F. MOLINA, S.-J. SHIN and J. TAYLOR, 'Type-syntax and token-syntax in diagrammatic systems', *Proc. FOIS* 2001 (2nd International Conference on Formal Ontology in Information Systems, Maine, USA) (ACM Press, 2001) 174–185. 147, 152

22. J. HOWSE, F. MOLINA, S.-J. SHIN and J. TAYLOR, 'On diagram tokens and types', *Proc. Diagrams* 2002 (International Conference on the Theory and Application of Diagrams), Lecture Notes in Artificial Intelligence 2317 (Springer, 2002) 76–90. 147

23. J. HOWSE, F. MOLINA and J. TAYLOR, 'SD2: a sound and complete diagrammatic reasoning system', *Proc. ASC* 2000 (Artificial Intelligence and Soft Computing, Banff) (IASTED/ACTA Press, 2000) 402–408. 147, 149, 151, 152

24. J. HOWSE, F. MOLINA and J. TAYLOR, 'On the completeness and expressiveness of spider diagram systems', *Proc. Diagrams* 2000 (International Conference on the Theory and Application of Diagrams), Lecture Notes in Artificial Intelligence 1889 (Springer, 2000) 26–41. 151, 152

25. J. HOWSE, F. MOLINA and J. TAYLOR, 'A sound and complete diagrammatic reasoning system', *Proc. VL* 2000 (IEEE Symposium on Visual Languages, Seattle) (IEEE Computer Society Press, 2000) 127–136. 147, 149, 151, 152

26. J. HOWSE, F. MOLINA, J. TAYLOR and S. KENT, 'Reasoning with spider diagrams', *Proc. VL* 1999 (IEEE Symposium on Visual Languages, Tokyo, Sept. 1999) (IEEE Computer Society Press) 138–147.

27. J. HOWSE, F. MOLINA, J. TAYLOR, S. KENT and J. GIL, 'Spider diagrams: a diagrammatic reasoning system', *J. Visual Languages and Computing* 12 (2001) 299–324.

28. S. KENT, 'Constraint diagrams: visualising invariants in object oriented models', *Proc. OOPSLA*97, ACM SIGPLAN Notices 32 (1997) 327–341. 146, 191

29. O. LEMON, 'Comparing the efficacy of visual languages', *Words, proofs and diagrams* (ed. D. Barker-Plummer, D. I. Beaver, J. van Benthem and P. Scotto di Luzio, CSLI Publications, 2002) 47–69. 152

30. R. LULL, *Ars magma* (Lyons, 1517). 145

31. F. MOLINA, 'Reasoning with extended Venn–Peirce diagrammatic systems', PhD Thesis, University of Brighton, 2001. 147, 149, 151, 152, 158, 160, 161, 165, 176, 177

32. P. MUTTON, P. RODGERS and J. FLOWER, 'Drawing graphs in Euler Diagrams', *Proc. Diagrams* 2004 (International Conference on the Theory and Application of Diagrams, Cambridge, March 2004), Lecture Notes in Artificial Intelligence 2980 (ed. A. Blackwell, *et al.*, Springer, 2004) 66–81. 191

33. OBJECT MANAGEMENT GROUP, 'Unified Modeling Language specification', available at www.omg.org. 146, 191

34. C. PEIRCE, *Collected papers*, vol. 4 (Harvard Univ. Press, 1933). 146

35. P. RODGERS, P. MUTTON and J. FLOWER, 'Dynamic Euler diagram drawing', *Proc. VLHCC* 2004 (Visual Languages and Human Centric Computing, Rome) (IEEE Computer Society Press, 2004) 147–156. 191

36. P. SCOTTO DI LUZIO, 'Patching up a logic of Venn diagrams', *Proc.* 6*th CSLI Workshop on Logic, Language and Computation* (CSLI Publications, Stanford, 2000). 147

37. S.-J. SHIN, *The logical status of diagrams* (Cambridge Univ. Press, 1994). 145, 146, 147, 151, 152, 191

38. G. STAPLETON, 'Reasoning with constraint diagrams', PhD Thesis, University of Brighton, August 2004. 147, 173, 191

39. G. STAPLETON, J. HOWSE and J. TAYLOR, 'A constraint diagram reasoning system', *Proc. VLC* 2003 (Distributed Multimedia Systems, International Conference on Visual Languages and Computing, Florida International University, Miami) (Knowledge Systems Institute, 2003) 263–270. 191

40. G. STAPLETON, J. HOWSE and J. TAYLOR, 'A decidable constraint diagram reasoning system', *J. Logic Comput.*, to appear 2005. 191

41. G. STAPLETON, J. HOWSE, J. TAYLOR and S. THOMPSON, 'What can spider diagrams say?' *Proc. Diagrams* 2004 (International Conference on the Theory and Application of Diagrams, Cambridge, March 2004), Lecture Notes in Artificial Intelligence 2980 (ed. A. Blackwell, *et al.*, Springer, 2004) 112–127. 147

42. G. STAPLETON, S. THOMPSON, J. HOWSE and J. TAYLOR, 'The expressiveness of spider diagrams', *J. Logic Comput.* 14 (2004) 857–880. 190

43. J. VENN, 'On the diagrammatic and mechanical representation of propositions and reasonings', *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science* 9 (1880) 1–18. 145

44. A. VERROUST and M.-L. VIAUD, 'Ensuring the drawability of extended Euler diagrams for up to 8 sets', *Proc. Diagrams* 2004 (International Conference on the Theory and Application of Diagrams, Cambridge, March 2004), Lecture Notes in Artificial Intelligence 2980 (ed. A. Blackwell, *et al.*, Springer, 2004) 128–141. 152, 156

45. J. WARMER and A. KLEPPE, *The Object Constraint Language: precise modeling with UML* (Addison-Wesley, 1998). 147, 191

John Howse John.Howse@brighton.ac.uk
Gem Stapleton G.E.Stapleton@brighton.ac.uk
John Taylor John.Taylor@brighton.ac.uk

http://www.brighton.ac.uk/cmis/research/vmg/

School of Computing, Mathematical and Information Sciences
University of Brighton
Watts Building
Brighton BN2 4GJ
United Kingdom

https://doi.org/10.1112/S1461157000000942 Published online by Cambridge University Press