On the Completeness and Expressiveness of Spider Diagram Systems

John Howse, Fernando Molina and John Taylor

School of Computing & Mathematical Sciences
University of Brighton, UK
{John.Howse, F.Molina, John.Taylor}@bton.ac.uk

Abstract. *Spider diagram systems provide a visual language that extends the popular and intuitive Venn diagrams and Euler circles. Designed to complement object-oriented modelling notations in the specification of large software systems they can be used to reason diagrammatically about sets, their cardinalities and their relationships with other sets. A set of reasoning rules for a spider diagram system is shown to be sound and complete. We discuss the extension of this result to diagrammatically richer notations and also consider their expressiveness. Finally, we show that for a rich enough system we can diagrammatically express the negation of any diagram.*

Keywords Diagrammatic reasoning, visual formalisms.

1. Introduction

Euler circles [2] is a graphical notation for representing relations between classes. This notation is based on the correspondence between the topological properties of enclosure, exclusion and intersection and the set-theoretic notions of subset, disjoint sets, and set intersection, respectively. Venn [14] modified this notation to illustrate all possible relations between classes by showing all possible intersections of contours and by introducing shading in a region to denote the empty set. However a disadvantage of this system is its inability to represent existential statements. Peirce [11] modified and extended the Venn system by introducing notation to represent empty and non-empty sets and disjunctive information. Recently, full formal semantics and inference rules have been developed for Venn-Peirce diagrams [13] and Euler diagrams [6]; see also [1, 5] for related work. Shin [13] proves soundness and completeness results for two systems of Venn-Peirce diagrams.

Spider diagrams [3, 7, 8, 9] emerged from work on constraint diagrams [4, 10] and extend the system of Venn-Peirce diagrams investigated by Shin. Constraint diagrams are a visual diagrammatic notation for expressing constraints such as invariants, preconditions and postconditions that can be used in conjunction with the Unified Modelling Language (UML) [12] for the development of object-oriented systems.

In [8] we considered a system of spider diagrams called SD2 that extended the diagrammatic rules and enhanced the semantics of the second Venn-Peirce system that Shin investigated (i.e., Venn-II, see [13] Chapter 4) to include upper and lower bounds for the cardinality of represented sets. In the proof of completeness of SD2 we opted for a strategy in which the diagram that results from combining a set of diagrams and the diagram that is the consequence of that set are expanded in a way similar to the disjunctive normal form in symbolic logic. This strategy extends to other similar systems, including the one considered in this paper. In this paper, we extend SD2 by introducing new notation and extending the inference rules to cover this notation. This extended system is shown to be sound and complete.

A discussion of the system SD2 is conducted in section 2, where the main syntax and semantics of the notation is introduced and soundness and completeness results are given. In section 3 we introduce new notation into the SD2 system, extend the inference rules described in section 2 to include the new notation and show that the extended system is sound and complete. We also enrich the system by providing additional results for reasoning with more expressive diagrams. In section 4, we show that we can express diagammatically the negation of every extended SD2 diagram. Section 5 states the conclusions of this paper and details related, ongoing and future work. Throughout this paper, for space reasons, we omit most proofs.

2. Spider Diagrams: SD2

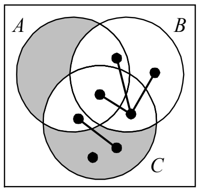
This section introduces the main syntax and semantics of SD2, a system of spider diagrams. For further details see [8].

2.1. Syntactic elements of unitary SD2 diagrams

A *contour* is a simple closed plane curve. A *boundary rectangle* properly contains all other contours. A *district* (or *basic region*) is the bounded subset of the plane enclosed by a contour or by the boundary rectangle. A *region* is defined, recursively, as follows: any district is a region; if r_1 and r_2 are regions, then the union, intersection, or difference, of r_1 and r_2 are regions provided these are non-empty. A *zone* (or *minimal region*) is a region having no other region contained within it; a zone is uniquely defined by the contours containing it and the contours not containing it. Contours and regions denote sets.

A *spider* is a tree with nodes (called *feet*) placed in different zones; the connecting edges (called *legs*) are straight lines. A spider *touches* a zone if one of its feet appears in that region. A spider may touch a zone at most once. A spider is said to *inhabit* the region which is the union of the zones it touches. For any spider s, the *habitat* of s, denoted $\eta(s)$, is the region inhabited by s. The set of complete spiders within region r is denoted by $S(r)$. The set of spiders touching region r is denoted by $T(r)$. A spider denotes the existence of an element in the set denoted by the habitat of the spider. Two distinct spiders denote distinct elements.

Every region is a union of zones. A region is *shaded* if each of its component zones is shaded. A shaded region denotes the empty set if it is not touched by any spider. A *unitary SD2 diagram* is a single boundary rectangle together with a finite collection of contours (all possible intersections of contours must occur, i.e., the underlying diagram is a Venn diagram), spiders and shaded regions. Each contour must be labelled and no two contours in the same unitary diagram can have the same label. The labelling of spiders is optional. For any unitary diagram D, we use $C = C(D)$, $Z = Z(D)$, $Z^* = Z^*(D)$, $R = R(D)$, $R^* = R^*(D)$, $L = L(D)$ and $S = S(D)$ to denote the sets of contours, zones, shaded zones, regions, shaded regions, contour labels and spiders of D, respectively.



D

Figure 1

The SD2 diagram D in Figure 1 can be interpreted as:

$$A-(B\cup C)=\varnothing \ \wedge\ 1\le|C-(A\cup B)|\le 2 \ \wedge\ 2\le|C-B| \ \wedge\ 1\le|B|.$$

2.2. Semantics of unitary SD2 diagrams

A *model* for a unitary SD2 diagram D is a pair $m = (\mathbf{U}, \Psi)$ where $\mathbf{U}$ is a set and $\Psi : C \rightarrow \mathit{Set}\,\mathbf{U}$, where $\mathit{Set}\,\mathbf{U}$ denotes the power set of $\mathbf{U}$, is a function mapping contours to subsets of $\mathbf{U}$. The boundary rectangle U is interpreted as $\mathbf{U}$, $\Psi(U) = \mathbf{U}$.

The intuitive interpretation of a zone is the intersection of the sets denoted by those contours containing it and the complements of the sets denoted by those contours not containing it. We extend the domain of Ψ to interpret regions as subsets of $\mathbf{U}$. First define $\Psi : Z \rightarrow \mathit{Set}\,\mathbf{U}$ by

$$\Psi(z) = \bigcap_{c\in C^+(z)} \Psi(c) \cap \bigcap_{c\in C^-(z)} \overline{\Psi(c)}$$

where $C^+(z)$ is the set of contours containing the zone z, $C^-(z)$ is the set of contours not containing z and $\overline{\Psi(c)} = \mathbf{U} - \Psi(c)$, the *complement* of $\Psi(c)$. Since any region is a union of zones, we may define $\Psi : R \rightarrow \mathit{Set}\,\mathbf{U}$ by

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

where, for any region r, $Z(r)$ is the set of zones contained in r.

The semantics predicate $P_D(m)$ of a unitary diagram D is the conjunction of the following two conditions:

Distinct Spiders Condition: The cardinality of the set denoted by region r of unitary diagram D is greater than or equal to the number of complete spiders in r:

$$\bigwedge_{r\in R} |\Psi(r)| \ge |S(r)|$$

Shading Condition: The cardinality of the set denoted by a shaded region r of unitary diagram D is less than or equal to the number of spiders touching r:

$$\bigwedge_{r \in R^*} |\Psi(r)| \leq |T(r)|$$

2.3. Compound diagrams and multi-diagrams

Given two unitary diagrams D_1 and D_2, we can *connect* D_1 and D_2 with a straight line to produce a diagram $D = D_1–D_2$. If a diagram has more than one rectangle, then it is a *compound* diagram. The 'connection operation' is commutative, $D_1–D_2 = D_2–D_1$. Hence, if a diagram has n unitary components, then these components can be placed in any order.

The semantics predicate of a compound diagram D is the disjunction of the semantics predicates of its component unitary diagrams; the boundary rectangles of the component unitary diagrams are interpreted as the same set **U**.

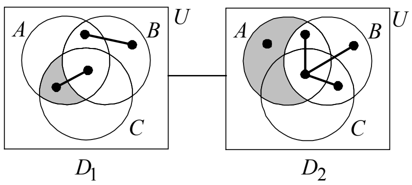


Figure 2

The compound diagram $D = D_1–D_2$ in Figure 2 asserts that:

$$(\exists x, y \bullet\ x \in A \cap C\ \wedge\ y \in B - C\ \wedge\ |A \cap C - B| \leq 1)\ \vee$$
$$(\exists x, y \bullet\ x \in B\ \wedge\ y \in A - (B \cup C)\ \wedge\ |A - B| = 1).$$

A spider *multi-diagram* is a finite collection Δ of spider diagrams. The semantics predicate of a multi-diagram is the conjunction of the semantics predicates of the individual diagrams; the boundary rectangles of all diagrams are interpreted as the same set **U**. Contours with the same labels in different individual diagrams of a multi-diagram Δ are interpreted as the same set.

In [8] we describe how to compare regions across diagrams. This formalizes the intuitively clear notion of 'corresponding regions' in different diagrams. For example, in figure 3, the region $z = z_1 \cup z_2$ in D corresponds to the zone z' in D' since both represent the set $B - A$.

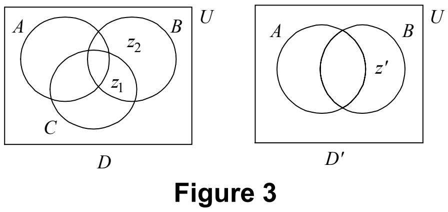


Figure 3

2.4. Compliance and Consistency

A model $m = (\mathbf{U}, \Psi)$ *complies* with diagram D if it satisfies its semantic predicate $P_D(m)$. We write $m \vDash D$. That is, $m \vDash D \Leftrightarrow P_D(m)$. Similarly, a model m *complies* with multi-diagram Δ if it satisfies its semantic predicate $P_\Delta(m)$. That is, $m \vDash \Delta \Leftrightarrow P_\Delta(m)$. A diagram is *consistent* if and only if it has a compliant model. All SD2 diagrams are consistent. However, there exist inconsistent SD2 *multi*-diagrams [8].

2.5. Rules of transformation for SD2

In [8], we introduced rules that allow us to obtain one unitary diagram from a given unitary diagram by removing, adding or modifying diagrammatic elements. These rules are summarised below; they are based on the rules given by Shin in [13], which developed earlier work of Peirce [11].

Rule 1: Erasure of shading. We may erase the shading in an entire zone.

Rule 2: Erasure of a spider. We may erase a complete spider on any non-shaded region.

Rule 3: Erasure of a contour. We may erase a contour. When a contour is erased:

- any shading remaining in only a part of a zone should also be erased.
- if a spider has feet in two regions which combine to form a single zone with the erasure of the contour, then these feet are replaced with a single foot connected to the rest of the spider.

Rule 4: Spreading the feet of a spider. If a diagram has a spider s, then we may draw a node in any non-shaded zone z that does not contain a foot of s and connect it to s.

Rule 5: Introduction of a contour. A new contour may be drawn interior to the bounding rectangle observing the partial-overlapping rule: each zone splits into two zones with the introduction of the new contour. Each foot of a spider is replaced with a connected pair of feet, one in each new zone. Shaded zones become corresponding shaded regions.

Rule 6: Splitting spiders. If a unitary diagram D has a spider s whose habitat is formed by n zones, then we may replace D with a connection of n unitary diagrams $D_1 - \ldots - D_n$ where each foot of the spider s touches a different corresponding zone in each diagram D_i.

Rule 7: Rule of excluded middle. If a unitary diagram D has a non-shaded zone z where $|S(z)| = n$, then we may replace D with $D_1 – D_2$, where D_1 and D_2 are unitary and one of the corresponding zones of z is shaded with $|S(z)| = n$ and the other is not shaded with $|S(z)| = n + 1$.

Rule 8: The rule of connecting a diagram. For a given diagram D, we may connect any diagram D' to D.

Rule 9: The rule of construction. Given a diagram $D_1 - \ldots - D_n$, we may transform it into D if each $D_1, \ldots, D_n$ may be transformed into D by a sequence of the first eight transformation rules.

2.6. Consistency of a multi-diagram and combining diagrams

Definition: An α diagram is a diagram in which no spider's legs appear; that is, the habitat of any spider is a zone.

Any SD2 diagram D can be transformed into an α diagram by repeated application of rule 6, splitting spiders.

Two unitary α diagrams with the same contour set are consistent if and only if for all zones

(i) corresponding shaded zones contain the same number of spiders;

(ii) when a shaded zone in one diagram corresponds to a non-shaded zone in the other, the shaded zone contains at least as many spiders as the non-shaded zone.

Two diagrams D^1 and D^2 are consistent if they can be transformed into α diagrams with the same number of contours D^{1a} and D^{2b} (by rules 5 and 6) and there exist unitary components D_i^{1a} of D^{1a} and D_j^{2b} of D^{2b} such that D_i^{1a} and D_j^{2b} are consistent. See [8] for details.

Intuitively, the diagrammatic conditions (i) and (ii) would prevent the case in which two corresponding zones denote two sets whose cardinalities are inconsistent; this is the only case in which a pair of unitary α diagrams can be inconsistent.

Given two consistent diagrams, D^1 and D^2, we can combine them to produce a diagram $D = D^1 * D^2$, losing no semantic information in the process. Given D^1 and D^2, first transform them into α diagrams D^{1a} and D^{2b} with the same number of contours (using rules 5 and 6). Then the combined diagram $D = D^1 * D^2$ is the compound diagram formed by combining each component D_i^{1a} of D^{1a} with each component D_j^{2b} of D^{2b}; where two components are inconsistent, we do not obtain a corresponding component in D. In forming $D_i^{1a} * D_j^{2b}$, the number of spiders in a zone z is equal to the maximum of the numbers of spiders in the corresponding zones of D_i^{1a} and D_j^{2b}, and z is shaded if and only if at least one of the corresponding zones in D_i^{1a} or D_j^{2b} is shaded – see [8] for details.

The associativity of $*$ allows us to define the combination of the components of a multi-diagram $\Delta = \{D^1, D^2, \ldots, D^n\}$ unambiguously as $D^* = D^1 * D^2 * \ldots * D^n$. If Δ is inconsistent, the result will be no diagram; D^* is only defined when Δ is consistent. A test for the consistency of Δ is to try to evaluate D^*. Note that there exist inconsistent multi-diagrams each of whose proper subsets are consistent.

Rule 10: The rule of inconsistency. Given an inconsistent multi-diagram Δ, we may replace Δ with any multi-diagram.

Rule 11: The rule of combining diagrams. A consistent multi-diagram $\Delta = \{D^1, D^2, \ldots, D^n\}$ may be replaced by the combined diagram $D^* = D^1 * D^2 * \ldots * D^n$.

2.7. Soundness and completeness

D' is a consequence of D, denoted by $D \vDash D'$, if every compliant model for D is also a compliant model for D'. A rule is *valid* if, whenever a diagram D' is obtained from a diagram D by a single application of the rule then $D \vDash D'$. We write $\Delta \vdash D'$ to denote that diagram D' is obtained from multi-diagram Δ by applying a sequence of transformations. We write $D \vdash D'$ to mean $\{D\} \vdash D'$, etc.

For space reasons, we omit the proofs of the validity of rules 1 to 11. These rules are similar to those of the Venn-II system given in [13] and the proofs are fairly straightforward. It can be noted that rules 5, introduction of a contour, 6, splitting spiders, 7, excluded middle, and 11, combining diagrams do not lose any semantic information; this is useful for the proof completeness.

Theorem 1 Soundness Theorem Let Δ be a multi-diagram and D' a diagram.

Then $\Delta \vdash D' \Rightarrow \Delta \vDash D'$.

The result follows by induction from the validity of the rules. To prove completeness we show that if diagram D' is a consequence of multi-diagram Δ, then Δ can be transformed into D' by a finite sequence of applications of the rules. That is, $\Delta \vDash D' \Rightarrow \Delta \vdash D'$. The proof of the completeness theorem, together with an explanation of the strategy of the proof, can be found in [8].

Theorem 2 Completeness Theorem Let Δ be a multi-diagram and let D' be a diagram. Then $\Delta \vDash D' \Rightarrow \Delta \vdash D'$.

3. Extending the notation

In this section we introduce new notation into the SD2 system, extend the transformation rules to include the new notation and show that the extended system is sound and complete.

In fact the new syntactic elements we introduce do not increase the formal expressiveness of the system as a whole. However, they do increase the expressive power of unitary diagrams so that information can be represented more compactly and naturally using the extended notation.

3.1. Extending the notation

The notion of a *strand* was introduced into spider diagrams (see [3]) to provide a means for denoting that spiders may represent the same element should they occur in the same zone. A *strand* is a wavy line connecting two nodes, from different spiders, placed in the same zone. The *web* of spiders s and t, written $\zeta(s, t)$, is the union of zones z having the property that there is a sequence of spiders $s = s_0, s_1, s_2, \dots, s_n = t$ such that, for $i = 0, \dots, n-1$, s_i and s_{i+1} are connected by a strand in z. Two spiders with a non-empty web are referred to as *friends*. Two spiders s and t may (but not necessarily must) denote the same element if that element is in the set denoted by the web of s and t. In Figure 4, it is possible that if the elements denoted by s and t happen to be in $A \cap B$ then they may be the same element.

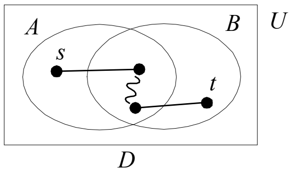


Figure 4

A *Schrödinger spider* is a spider each of whose feet is represented by a small open circle. A Schrödinger spider denotes a set whose size is zero or one: rather like Schrödinger's cat, one is not sure whether the element represented by a Schrödinger spider exists or not. Because of this, a Schrödinger spider in a non-shaded region does not assert anything; however, in shaded regions they are useful for specifying bounds for the cardinality of the set denoted by the region. They are also useful in representing the negation of a diagram (see next section). The set of Schrödinger spiders in diagram D is denoted by $S^* = S^*(D)$. We also let $T^*(r)$ denote the set of Schrödinger spiders touching region r and $S^*(r)$ denote the set of complete Schrödinger spiders in r. From Figure 5, we can deduce $|A - B| \leq 1$, $1 \leq |A \cap B| \leq 2$, $1 \leq |A| \leq 2$ and $|B - A| \leq 1$.

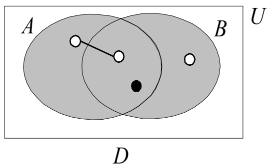


Figure 5

SD2 is based on Venn diagrams; that is, all possible intersection of districts must occur. In general, spider diagrams are based on Euler diagrams, in which information regarding set containment and disjointness is given visually (in terms of enclosure and exclusion). A spider diagram based on a Venn diagram is said to be in *Venn form*; otherwise, it is in *Euler form*. Figure 6 shows a spider diagram in Euler form.

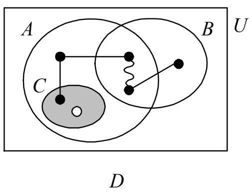


Figure 6

Extending SD2 by including strands and Schrödinger spiders and basing it on Euler diagrams we can express the system's semantics as the conjunction of the following conditions.

Spiders Condition: A non-Schrödinger spider denotes the existence of an element in the set denoted by its habitat and the elements denoted by two distinct non-Schrödinger spiders are distinct unless they fall within the set denoted by the zone in the spiders' web:

$$\exists x_1,...,\exists x_n \bullet \Big[\bigwedge_{i=1...n} x_i \in \Psi(\eta(s_i)) \wedge \bigwedge_{\substack{i,j=1...n \\ i\neq j}} (x_i = x_j \Rightarrow \bigvee_{z\in\zeta(s_i,s_j)} x_i, x_j \in \Psi(z)) \Big]$$

where $S = \{s_1, ..., s_n\}$.

Plane Tiling Condition: All elements fall within sets denoted by zones:

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

Shading Condition: The set denoted by a shaded region contains no elements other than those denoted by spiders (including Schrödinger spiders):

$$\bigwedge_{r\in R^*} |\Psi(r)| \leq |T(r)| + |T^*(r)|$$

3.2. Rules of transformation for extended notation

We can adapt and extend the rules of transformation given in section 2.6 to include rules involving the extended notation.

Adapted Rule 6: Splitting spiders. If a unitary diagram D has a spider s formed by two or more feet then we may remove a leg of this spider and replace D with a connection of two unitary diagrams D_1–D_2, each containing a different component of the split spider. If splitting a spider disconnects any component of the 'strand graph' in a zone, then the components so formed should be reconnected using one or more strands to restore the original component. The rule is reversible: if a compound diagrams contains diagram D_1 and D_2 as just described, then D_1 and D_2 can be replaced by diagram D.

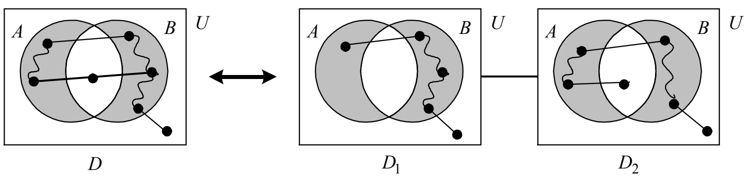


Figure 7

Rule 12: The rule of strand equivalence. A unitary α diagram D (that is, a unitary diagram in which each spider is single-footed) containing a strand in a zone can be replaced by a pair of connected unitary diagrams D_1 and D_2 which are copies of D. In D_1 the strand is deleted and in D_2 the two spiders connected by the strand are deleted and replaced by a single-footed spider in the zone originally containing the strand. Again, the rule is reversible.

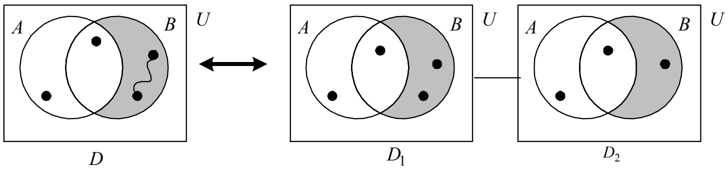


Figure 8

Rule 13: The rule of Schrödinger spider equivalence. A unitary diagram D containing a Schrödinger spider strand can be replaced by a pair of connected unitary diagrams D_1 and D_2 which are copies of D. In D_1 the Schrödinger spider is deleted and in D_2 the Schrödinger spider is deleted and replaced by non-Schrödinger spider. Again, the rule is reversible.

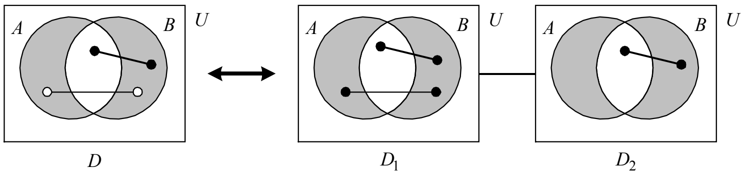


Figure 9

Rule 14: The rule of equivalence of Venn and Euler forms. For a given unitary diagram in Euler form there is an equivalent unitary diagram in Venn form.

Figure 10 shows equivalent diagrams in Euler (left) and Venn form.

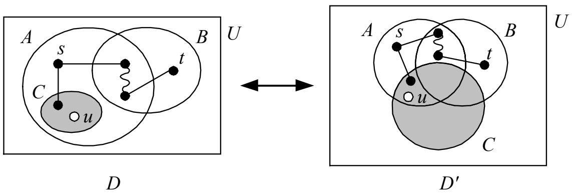


Figure 10

3.3. Soundness and completeness of extended system

The validity of the rules stated above is given in Theorem 3.

Theorem 3 If D' is derived from D by a sequence of applications of adapted rule 6 or rule 12 or rule 13 or rule 14, then $D \vDash D'$ and $D' \vDash D$.

The proof is omitted for space reasons. The soundness of the extended SD2 system is derived by induction from the validity of each of the rules.

Theorem 4 Completeness Theorem for extended SD2 system. Let Δ be an extended SD2 multi-diagram and let D' be an extended SD2 diagram. Then $\Delta \vDash D' \Rightarrow \Delta \vdash D'$.

Proof. Assume that $\Delta \vDash D'$. We apply rule 14 to each component of each diagram in Δ, so that each diagram is in Venn form. Next, we apply adapted rule 6 repeatedly to each diagram until all the spiders are single footed. We then apply rule 12 repeatedly until we have removed all strands and then rule 13 repeatedly to remove all Schrödinger spiders. The resulting multi-diagram Δ^{SD2} is a set of SD2 diagrams. We apply the same strategy to D' to produce D'^{SD2} an SD2 diagram. By Theorem 3 $\Delta^{SD2} \vDash \Delta$, and $D' \vDash D'^{SD2}$. Hence by transitivity, $\Delta^{SD2} \vDash D'^{SD2}$. So, by Theorem 2 (completeness of SD2), $\Delta^{SD2} \vdash D'^{SD2}$. For each diagram D^{SD2} in Δ^{SD2} there is a diagram D in Δ such that $D \vdash D^{SD2}$, so we have, $\Delta \vdash D'^{SD2}$. Each of the rules adapted rule 6, 12, 13, 14 is reversible, so $D'^{SD2} \vdash D'$. Hence, by transitivity, $\Delta \vdash D'$.

3.4. Derived reasoning results

In this section we enrich the system by providing additional results for reasoning with diagrams containing strands or Schrödinger spiders. Several of the results are extensions of the given rules to include diagrams with strands. Each lemma is illustrated in a figure immediately following the lemma. Their proofs are omitted.

Lemma 1 (Rule 2): Erasure of a spider. We may erase a complete spider on any non-shaded region and any strand connected to it. If removing a spider disconnects any component of the 'strand graph' in a zone, then the components so formed should be reconnected using one or more strands to restore the original component.

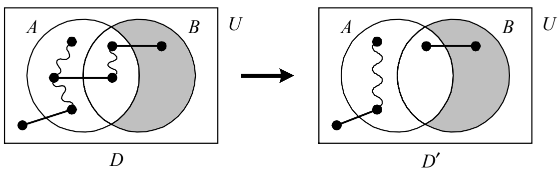


Figure 11

Lemma 2 (Rule 5): Introduction of a contour. A new contour may be drawn interior to the bounding rectangle observing the partial-overlapping rule: each zone splits into two zones with the introduction of the new contour. Each foot of a spider is replaced with a connected pair of feet, one in each new zone. Likewise, each strand bifurcates and becomes a pair of strands, one in each new zone. Shaded zones become corresponding shaded regions.

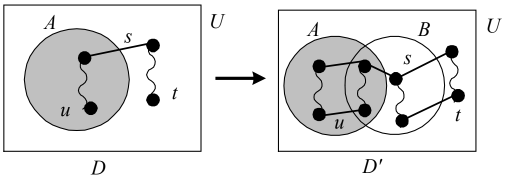


Figure 12

Lemma 3 (Rule 7): Rule of excluded middle. If a unitary diagram D has a non-shaded zone z, then we may replace D with D_1–D_2. D_1 is copy of D where zone z is shaded and D_2 is copy of D where zone z contains an extra single-footed non-Schrödinger spider.

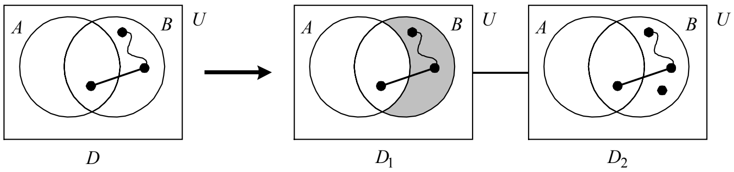


Figure 13

Lemma 4: Elimination of a strand I. Let D be a unitary diagram containing a single-footed non-Schrödinger spider s in an unshaded zone z connected by a strand to a non-Schrödinger spider t whose habitat is unshaded. Then D is equivalent to a diagram D' in which the spider t has been deleted.

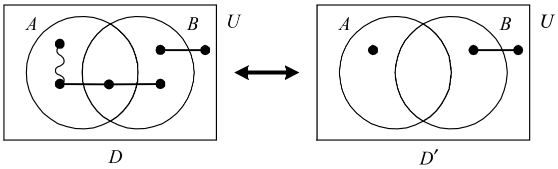


Figure 14

Lemma 5: Elimination of a strand II. Let D be a unitary diagram containing a single-footed non-Schrödinger spider s in an unshaded zone z connected by a strand to a non-Schrödinger spider t. Then D is equivalent to a diagram D' in which the part of t lying in an unshaded region is deleted and that part of t lying in a shaded region is replaced by a Schrödinger spider.

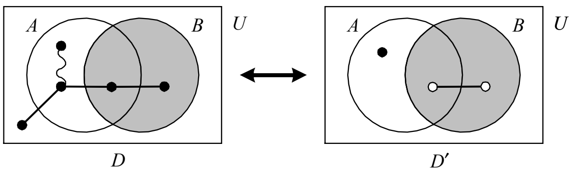
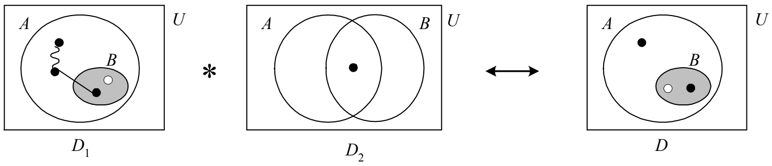
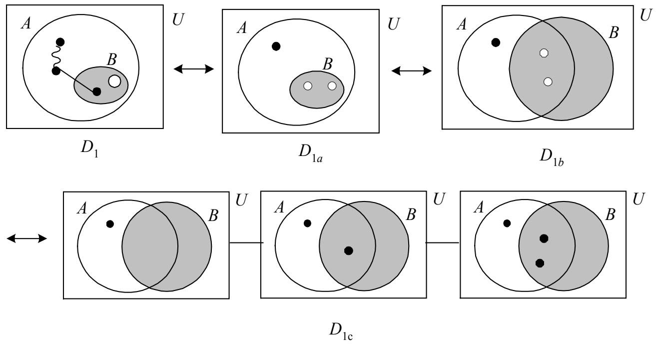


Figure 15

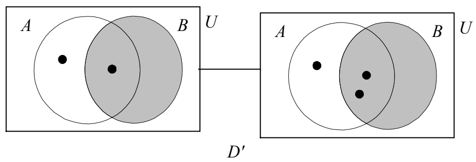
Example. In this example we give a diagrammatic proof that the combination of D_1 and D_2 is equivalent to D.



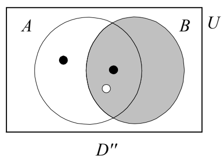
First, transform D_1 into its equivalent α SD2 diagram by lemma 5, elimination of a strand, rule 14, Venn-Euler equivalence, and rule 13, Schrödinger spider equivalence, as follows



Combining D_{1c} and D_2 by rule 11 we obtain D'.



Again, by rule 13, Schrödinger spider equivalence, we obtain a unitary diagram D''.



Finally, since we can transform the Euler form D into the Venn form D'' above, we can complete proof and obtain D from D''.

It is worth noting that, given an SD2 diagram in Euler form, the transformation to its corresponding Venn form is algorithmic. However, in general transforming from Venn to Euler forms is not mechanical.

4. Negation

One of the important properties of the SD2 system is that it is syntactically rich enough to express the negation of any diagram D in a reasonably natural manner. We describe the construction of the negation of D, a diagram which may include Schrödinger spiders, in several stages as follows.

(i) D is an α unitary diagram with n zones which are shaded or contain spiders. The negation of D gives a (compound) diagram with m components ($m \geq n$). Any non-shaded zone z with p spiders gives a unitary component where its corresponding zone z_1 is shaded and contains $p - 1$ Schrödinger spiders. Any shaded zone z with q Schrödinger spiders and r ($r \geq 1$) non-Schrödinger spiders gives two unitary components where its corresponding zones z_1 and z_2 contain $q + r + 1$ non-Schrödinger spiders and $r - 1$ Schrödinger spiders respectively and z_2 is shaded. (When $r = 0$ we obtain a single unitary component whose corresponding zone z_1 contains $q + 1$ non-Schrödinger spiders). If D is an α unitary diagram where no zone is shaded or contains spiders, its negation is any inconsistent multi-diagram.

(ii) D is a compound diagram with n α unitary components. The negation of D gives a multi-diagram formed by n (compound) diagrams being each member of the collection the negation of each α unitary component as in case (i)

(iii) D is any (compound) diagram. We transform it into its α diagram D^{α} and negate D^{α} as in (ii).

(iv) D is any multi-diagram. The negation of D is equivalent to negate D^*, the result of combining the components of D, as in (ii).

Figure 16 illustrates the negation of an α unitary diagram. We use the diagrammatic notation $\overline{D}$ to denote the negation of D.

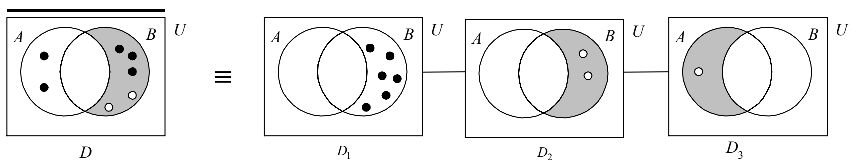


Figure 16

5. Conclusion and related work

We have extended the syntax and inference rules of the system of spider diagrams we call SD2 and have shown that this extended system is sound and complete. We have given a number of derived reasoning rules to aid reasoning in the extended notation and shown that we can syntactically give the inverse of any diagram in this system. This extended system contains most of the syntactic elements of spider diagrams given in [3].

Our longer term aim is to prove similar results for constraint diagrams, and to provide the necessary mathematical underpinning for the development of software tools to aid the diagrammatic reasoning process.

References

[1] Allwein, G, Barwise, J (1996) *Logical Reasoning with Diagrams*, OUP.

[2] Euler, L (1761) *Lettres a Une Princesse d'Allemagne*. Vol. 2, Letters No. 102-108.

[3] Gil, Y., Howse, J., Kent, S. (1999) Formalizing Spider Diagrams, *Proceedings of IEEE Symposium on Visual Languages* (VL99), IEEE Computer Society Press.

[4] Gil, Y., Howse, J., Kent, S. (1999) Constraint Diagrams: a step beyond UML, *Proceedings of TOOLS USA 1999*, IEEE Computer Society Press.

[5] Glasgow, J, Narayanan, N, Chandrasekaran, B (1995) *Diagrammatic Reasoning*, MIT Press.

[6] Hammer, E.M. (1995) *Logic and Visual Information*, CSLI Publications.

[7] Howse, J., Molina, F., Taylor, J., (2000) A Sound and Complete Diagrammatic Reasoning System, accepted for ASC 2000, IASTED Conference on Artificial Intelligence and Soft Computing.

[8] Howse, J., Molina, F., Taylor, J., (2000) SD2: A Sound and Complete Diagrammatic Reasoning System, accepted for VL2000, IEEE Symposium on Visual Languages.

[9] Howse, J., Molina, F., Taylor, J., Kent, S. (1999) Reasoning with Spider Diagrams, *Proceedings of IEEE Symposium on Visual Languages* (VL99), IEEE Computer Society Press.

[10] Kent, S. (1997) Constraint Diagrams: Visualising Invariants in Object Oriented Models. *Proceedings of OOPSLA 97*

[11] Peirce, C (1933) *Collected Papers*. Vol. 4. Harvard University Press.

[12] Rumbaugh, J., Jacobson, I., Booch, G. (1999) *Unified Modeling Language Reference Manual.* Addison-Wesley.

[13] Shin, S-J (1994) *The Logical Status of Diagrams*. CUP.

[14] Venn, J (1880) On the Diagrammatic and Mechanical Representation of Propositions and Reasonings, *Phil. Mag.* 123.