



A normal form for spider diagrams of order

Aidan Delaney^{a,*}, Gem Stapleton^a, John Taylor^a, Simon Thompson^b

^a University of Brighton, Brighton BN2 4GJ, United Kingdom

^b School of Computing, University of Kent, CT2 7NF, United Kingdom



ARTICLE INFO

Article history:

Received 25 September 2014

Received in revised form

24 October 2014

Accepted 28 October 2014

This paper has been recommended for acceptance by S.-K. Chang

Available online 31 October 2014

Keywords:

Spider diagrams

Logic

Normal form

Euler diagrams

ABSTRACT

We develop a reasoning system for an Euler diagram based visual logic, called spider diagrams of order. We define a normal form for spider diagrams of order and provide an algorithm, based on the reasoning system, for producing diagrams in our normal form. Normal forms for visual logics have been shown to assist in proving completeness of associated reasoning systems. We wish to use the reasoning system to allow future direct comparison of spider diagrams of order and linear temporal logic.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Shin's rebirth of Peirce's α and β systems for reasoning [15] has produced a variety of Euler diagram based visual logics, for example [7,4,17,18]. Euler diagram based visual logics allow reasoning about sets, their elements and their relationships. Associated with visual logics are reasoning systems that embody equivalence between diagrams [2,10,19]. Spider Diagrams of Order (SDoO) and Second-Order Spider Diagrams [3] differ from the main body of work on Euler diagram based logics as elements of their token syntax were designed to be as expressive as star-free regular languages and regular languages respectively.

Weakly expressive language classes, such as regular languages and star-free regular languages, are used to formalise real-world temporal specifications [5,12]. Due to the real-world application there has been recent interest in incorporating temporal semantics in these diagrammatic logics [1,14]. In this paper we address the problem of adding temporal semantics to Euler diagrams by adding a

syntax and semantics for specifying order of the elements. Furthermore, we develop the first reasoning system for an Euler diagram based logic that includes an order relation. In demonstrating our reasoning system for spider diagrams of order we produce both a normal form and an algorithm to produce the normal form. Our algorithm also contributes to the recent interest in normal forms for Euler diagram based logics [9].

In Section 2 we define the syntax and semantics of spider diagrams of order. In Section 3 we present each of our reasoning rules. Thereafter, in Section 4 we present our normalisation algorithm by example. An implementation of our algorithm is available under an open-source license at <https://github.com/AidanDelaney/SpiderReasoning>.

2. Spider diagrams of order

The Euler diagram in Fig. 1(a) contains three labelled contours and six zones. A *zone* is defined to be a pair, (*in*, *out*), of disjoint subsets of the set of contour labels. The set *in* contains the labels of the contours that the zone is inside whereas *out* contains the labels of the contours that the zone is outside. The set of all zones is denoted \mathcal{Z} .

* Corresponding author.

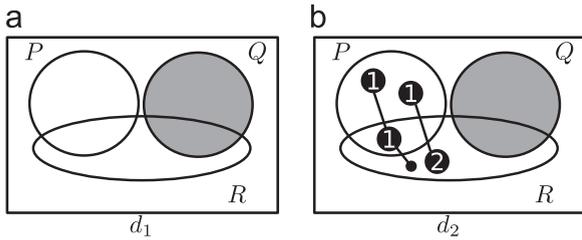


Fig. 1. An Euler and a spider diagram.

A region is a set of zones. As an example, there exists a zone inside the contour P but outside both contours Q and R denoted $(\{P\}, \{Q, R\})$. The zone inside the bounding box and outside all contours can be described by being inside \emptyset and outside $\{P, Q, R\}$. We note that there is no zone in the diagram corresponding to $(\{P, Q\}, \{R\})$ i.e. there is no zone inside contours labelled P and Q but outside the contour labelled R . Euler diagrams may be conjoined using the symbol \wedge , disjointed using the symbol \vee or negated using the symbol \neg .

A Spider Diagram of Order is an Euler diagram containing one or more graphs. The vertices of a graph are labelled with ‘•’ or an integer. A graph is restricted such that it is acyclic and may not have more than one vertex of a given label in a given zone. To maintain consistency with the literature we call graphs of this form spiders and term a vertex within a graph to be a spider foot. The diagram in Fig. 1(b) contains two spiders, one spider consisting of three feet labelled ‘1’, ‘1’ and ‘•’, and the other spider contains two feet labelled ‘1’ and ‘2’. In the following definition, as throughout the paper, we use \cup to mean set union, \cap to mean set intersection and $A - B$ to denote the set difference between A and B .

Definition 1. A spider foot is an element of the set $(\mathbb{Z}^+ \cup \{\bullet\}) \times \mathcal{Z}$ and the set of all feet is denoted \mathcal{F} . A spider foot $(k, z) \in \mathcal{F}$ has rank k where $k \in \mathbb{Z}^+ \cup \{\bullet\}$. The rank of a spider foot induces a relation $<$ on the feet, defined by $(k_1, z_1) < (k_2, z_2)$ if both $k_1, k_2 \in \mathbb{Z}^+$ and $k_1 < k_2$ hold or $k_1 = \bullet$ or $k_2 = \bullet$.

Whilst it may seem strange that $<$ as just defined is not a strict ordering (because \bullet is both less than and greater than all other feet) this choice of $<$ simplifies many definitions.

Definition 2. A spider, s , is a non-empty set of feet together with a positive natural number, that is $s \in \mathbb{Z}^+ \times (\mathbb{P}\mathcal{F} - \{\emptyset\})$, and the set of all spiders is denoted \mathcal{S} . The set p is the foot set of spider $s = (n, p)$. The habitat of a spider $s = (n, p)$ is the region $\eta(s) = \{z : (k, z) \in p\}$.

In the semantics predicate, we will define a spider to represent an element corresponding to only one of its feet in one of the constituent regions of its habitat.

Formally, the set of all contour labels is denoted \mathcal{C} .

Definition 3. A unitary spider diagram of order, d , is a quadruple $\langle \mathcal{C}(d), \mathcal{Z}(d), \text{ShZ}(d), \mathcal{S}(d) \rangle$ where:

- $\mathcal{C}(d) \subseteq \mathcal{C}$ is a finite set of contour labels,
- $\mathcal{Z}(d) \subseteq \{(in, C(d) - in) : in \in \mathcal{C}(d)\}$ is a set of zones,
- $\text{ShZ}(d) \subseteq \mathcal{Z}(d)$ is a set of shaded zones,

$\mathcal{S}(d) \subseteq \mathcal{S}$ is a finite set, called the spider identifiers, such that for all spiders $(n_1, p_1), (n_2, p_2)$ in $\mathcal{S}(d)$ if $p_1 = p_2$ then $n_1 = n_2$.

The set of spider identifiers, in effect, counts the number of spiders with a particular habitat. The set of spiders in d is defined to be

$$\mathcal{S}(d) = \{(i, p) : (n, p) \in \mathcal{S}(d) \wedge 1 \leq i \leq n\}.$$

The symbol \perp is also a unitary spider diagram. We define $\mathcal{C}(\perp) = \mathcal{Z}(\perp) = \text{ShZ}(\perp) = \mathcal{S}(\perp) = \emptyset$.

Spider diagrams of order may also be combined using the Boolean operations \wedge, \vee and \neg . In addition we allow the binary connective \triangleleft . A spider diagram of order that contains one of the \wedge, \vee, \neg or \triangleleft connectives is a compound diagram. Furthermore, a spider diagram of order containing either no spiders or containing spiders consisting of only single feet is an α -diagram. A zone can be considered to be missing from a spider diagram as presented in [10].

Definition 4. Given an Euler diagram, d , a zone (in, out) is said to be missing if it is in the set $\{(in, C(d) - in) : in \subseteq C(d)\} - \mathcal{Z}(d)$ with the set of such zones denoted $\text{MZ}(d)$. If d has no missing zones then d is in Venn form.

Spider diagrams of order have a model based semantics.

Definition 5. An interpretation is a triple $(U, <, \Psi)$ where U is a universal set and $\Psi : \mathcal{C} \rightarrow \mathbb{P}U$ is a function that assigns a subset of U to each contour label and $<$ is a strict total order on U . The function Ψ can be extended to interpret zones and sets of regions as follows:

1. each zone, $(in, out) \in \mathcal{Z}$, represents the set
$$\Psi(z) = \bigcap_{c \in in} \Psi(c) \cap \bigcap_{c \in out} (U - \Psi(c))$$
 and
2. each region, $r \in \mathbb{P}\mathcal{Z}$, represents the set which is the union of the sets represented by r 's constituent zones, that is
$$\Psi(r) = \bigcup_{z \in r} \Psi(z).$$

Definition 6. Let $I = (U, <, \Psi)$ be an interpretation and let $d (\neq \perp)$ be a unitary spider diagram. Then I is a model for d , denoted $I \models d$, if and only if the following conditions hold.

1. The missing zones condition: All of the missing zones represent the empty set, that is, $\bigcup_{z \in \text{MZ}(d)} \Psi(z) = \emptyset$.
2. The spider mapping condition: There exists an injective function, $\varphi : \mathcal{S}(d) \rightarrow U$ and a function $f : \mathcal{S}(d) \rightarrow \mathcal{F}$, called a valid pair, such that the following conditions hold:
 - (a) The selected foot condition: Each spider s must map, under f , to a spider foot in its foot set:
$$\forall (n, p) \in \mathcal{S}(d) f(n, p) \in p.$$
 - (b) The spiders' location condition: All spiders represent elements in the sets represented by the zone in

which the selected foot, under f , is placed:

$$\forall s \in S(d) f(s) = (k, z) \Rightarrow \varphi(s) \in \Psi(z).$$

(c) *The shading condition:* Shaded regions represent a subset of elements denoted by spiders:

$$\forall z \in ShZ(d) \Psi(z) \subseteq im(\varphi).$$

(d) *The order condition:* The ordering information provided by the selected spider feet agrees with that provided by the strict order relation.

That is,

$$\forall s_1, s_2 \in S(d) (\varphi(s_1) < \varphi(s_2) \Rightarrow f(s_1) < f(s_2)).$$

If $d = \perp$ then no interpretation is a model for d .

The conjunction of conditions 1 and 2 above is the *semantics predicate* for spider diagrams of order. The semantics of the connectives \wedge , \vee and \neg extend in the obvious manner, however the semantics of \circ requires some explanation.

The \circ operation allows the specification of an order between unitary diagrams. In order to define the semantics of compound spider diagrams of order involving \circ we present the definition of ordered sum of interpretations [8].

Definition 7. The *ordered sum* of two interpretations $m_1 = (U_1, <_1, \Psi_1)$ and $m_2 = (U_2, <_2, \Psi_2)$, denoted $m_1 + m_2$, where U_1 and U_2 are disjoint, is the interpretation $m = (U, <, \Psi)$ such that

- $U = U_1 \cup U_2$,
- $< = <_1 \cup <_2 \cup \{(a, b) : a \in U_1 \wedge b \in U_2\}$,
- $\Psi(c) = \Psi_1(c) \cup \Psi_2(c)$ for all $c \in C$.

Given an interpretation, I , and a diagram, $D_1 \circ D_2$, I models $D_1 \circ D_2$ if there exist models m_1 and m_2 for D_1 and D_2 respectively and $I = m_1 + m_2$. We now define when two diagrams are semantically equivalent.

Definition 8. Let D_1 and D_2 be spider diagrams of order. If the model set for D_1 is exactly that of D_2 then D_1 and D_2 are *semantically equivalent*, denoted $D_1 \equiv D_2$.

Having defined the syntax and semantics of spider diagrams of order, we now define the rules of our sound reasoning system.

3. Reasoning rules

We introduce seven reasoning rules for spider diagrams of order. These rules are subsequently used to produce diagrams in normal form; providing the basis of our reasoning system. The rules are:

1. introduction of a contour label,
2. introduction of a missing zone,
3. splitting spiders,
4. separate rank and bounds,

5. factor lowest spiders,
6. drop spider-foot rank, and
7. rule of replacement.

The rules of replacement, introduction of a contour label, introduction of a missing zone and splitting spiders rule are generalised from [19], whereas the other three rules are completely new. For each rule we present a statement of the rule, a formal definition of the rule and an example of the use of the rule. All of the reasoning rules presented in this section produce semantically equivalent diagrams. Therefore, each of the rules defines its own inverse, and if D_2 is the consequence of applying a rule to D_1 then D_1 is the consequence of applying the inverse of the rule to D_2 . The reasoning rules are presented here without proof of soundness. Full proofs of soundness of each reasoning rule can be found in [6].

The following rule describes how to introduce a missing contour into a diagram producing a semantically equivalent diagram.

Rule 1 (Introduction of a contour label). Let d be a unitary spider diagram of order and let d' be a unitary spider diagram of order obtained from d by introduction of a contour label as follows.

- The new contour has a label that is not present in d .
- The contour introduced in d' splits each zone z of d into two zones z_1 and z_2 and both z_1 and z_2 are shaded where z is shaded.
- Each unordered foot of a spider in zone z of d is replaced in d' by a pair of unordered spider feet in z_1 and z_2 .
- Each ordered spider foot in zone z is similarly replaced in d' by a pair of ranked feet of the same rank in z_1 and z_2 .

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

Example 1. Let d be the diagram in Fig. 2(a). Let d' be the diagram in Fig. 2(b) where each zone in d has been split by the introduced contour R in d' . Each spider foot of d has been replaced by a pair of spider feet in d' such that one foot of the pair is the original foot and the other foot is extended into new zone created by the partition of the original zone containing the original foot.

Formal description of rule 1. Let d be a unitary spider diagram of order such that $d \neq \perp$. Let $l \in C - C(d)$ and let d' be the diagram where

$$C(d') = C(d) \cup \{l\},$$

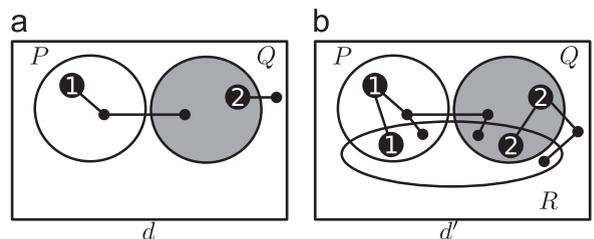


Fig. 2. The introduction of a contour label rule.

$$\begin{aligned} Z(d') &= \{(in \cup \{l\}, out) : (in, out) \in Z(d)\} \\ &\cup \{(in, out \cup \{l\}) : (in, out) \in Z(d)\}, \\ ShZ(d') &= \{(in \cup \{l\}, out) : (in, out) \in ShZ(d)\} \\ &\cup \{(in, out \cup \{l\}) : (in, out) \in ShZ(d)\}, \\ SI(d') &= \{(n, p') : \exists (n, p) \in SI(d) \wedge p' \\ &= \{(k, (in \cup \{l\}, out)) : (k, (in, out)) \in p\} \\ &\cup \{(k, (in, out \cup \{l\})) : (k, (in, out)) \in p\}\}. \end{aligned}$$

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

The add contour rule is sound as the resultant diagram is semantically equivalent to the original, as we now state.

Theorem 1. Let d be a unitary spider diagram of order such that $d \neq \perp$. Let $l \in C - C(d)$. Let d' be a spider diagram of order such that l is introduced to d resulting in d' by rule 1, introduction of a contour. Then $d \equiv_{\models} d'$.

Given an arbitrary diagram D and the introduction of a contour label rule we may introduce all contours in C producing a diagram containing all contours. The introduction of a missing zone rule, when coupled with the introduction of a contour label rule, allows us to produce diagrams in Venn-form containing all contours.

Rule 2 (Introduction of a missing zone). Let d be a unitary spider diagram of order with missing zone z and let d' be a copy of d where z is added to d' and z is shaded. Then d can be replaced by d' and vice versa.

Example 2. Let d be the unitary diagram in Fig. 3(a). The zone $z = (\{P, Q\}, \{\})$ is missing from d . Let d' be the diagram in Fig. 3(b). The zone z has been added as a shaded zone to d' .

Formal description of rule 2. Let $d (\neq \perp)$ be a unitary spider diagram of order. Let $z \in MZ(d)$. Then d' is a unitary spider diagram of order where

$$\begin{aligned} C(d') &= C(d), \\ Z(d') &= Z(d) \cup \{z\}, \\ ShZ(d') &= ShZ(d) \cup \{z\}, \\ SI(d') &= SI(d). \end{aligned}$$

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

The following establishes the soundness of the introduction of a missing zone rule.

Theorem 2. Let $d (\neq \perp)$ be a unitary spider diagram of order. Let $z \in MZ(d)$. Let d' be the diagram obtained by applying rule 2 introduction of a missing zone to d . Then $d \equiv_{\models} d'$.

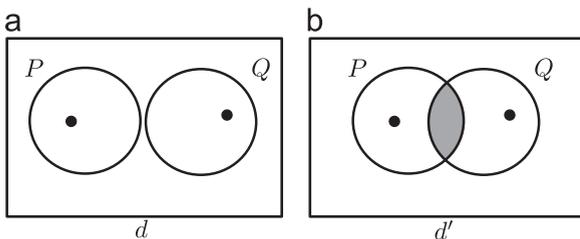


Fig. 3. The introduction of a missing zone rule.

The *splitting spiders* rule allows us to represent the disjunctive information held within a unitary diagram as a disjunction of unitary diagrams. By repeated application we generate a disjunction of α -diagrams.

Rule 3 (Splitting spiders). Let $d (\neq \perp)$ be a unitary spider diagram of order containing a spider s with foot set p where $|p| > 1$. Let d_1 and d_2 be copies of d and let $\{p_1, p_2\}$ be a partition of p . Then s is replaced in d_1 with s_1 where the foot set of s_1 is p_1 . Similarly, s is replaced in d_2 with s_2 where the foot set of s_2 is p_2 . Then d can be replaced by the diagram $d_1 \vee d_2$ and vice versa.

Example 3. Let d_1 be the diagram in Fig. 4(a). A single application of the splitting spiders rule may result in the diagram $d_2 \vee d_3$ in Fig. 4(b). A further application of the splitting spiders rule to d_3 produces a disjunction of α -spider diagrams of order.

In order to formally describe the split spiders rule we require the following definition, which allows us to remove spiders from, and add spiders to, a unitary diagram.

Definition 9. Let d be a unitary spider diagram of order. Let p be a foot set such that $\{z : (k, z) \in p\} \subseteq Z(d)$. Let d' be a unitary spider diagram of order that contains the same set of contours, set of zones and set of shaded zones as d . We may remove a spider with foot set p from d , denoted $d \ominus p$ to give d' such that, d' is identical to d except that:

$$\begin{aligned} SI(d') &= SI(d) - \{(n, p) : (n, p) \in SI(d)\} \\ &\cup \{(n - 1, p) : (n, p) \in SI(d) \wedge n > 1\}. \end{aligned}$$

Alternatively, we may add a spider with foot set p to d , denoted $d \oplus p$ to give d' such that, d' is identical to d except that:

$$\begin{aligned} SI(d') &= SI(d) - \{(n, p) : (n, p) \in SI(d)\} \\ &\cup \{(n + 1, p) : (n, p) \in SI(d)\} \\ &\cup \{(1, p) : (n, p) \notin SI(d)\}. \end{aligned}$$

Formal description of rule 3. Let d be a spider diagram of order containing a spider $s = (n, p)$ with $|p| > 1$ and let $\{p_1, p_2\}$ be a partition of p . Let d_1 and d_2 be unitary diagrams such that:

- $d_1 = (d \ominus p) \oplus p_1$, and
- $d_2 = (d \ominus p) \oplus p_2$.

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

Theorem 3. Let d be a unitary spider diagram of order and let $d_1 \vee d_2$ be the result of the application of rule 3 splitting spiders to d . Then $d \equiv_{\models} d_1 \vee d_2$.

When given an arbitrary diagram D , we may use the introduction of a contour label, introduction of a missing zone and splitting spiders rules, to produce a diagram D_α where each unitary component is in Venn-form, contains all contours in C and is an α -diagram. We now introduce a series of three rules which, when given D_α , produce a diagram that contains no ranked feet. The first of our three

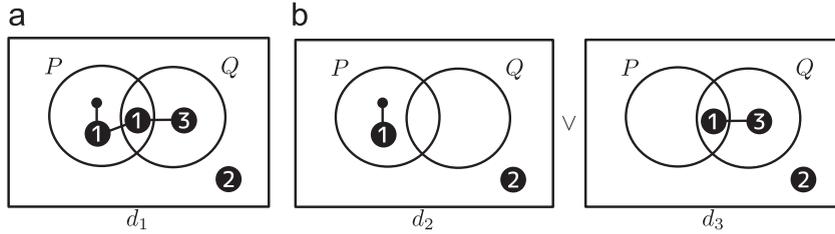


Fig. 4. The splitting spiders rule.

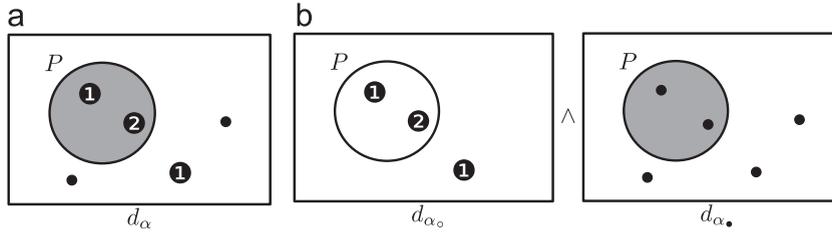


Fig. 5. The separate rank and bounds rule.

rules isolates the order information from the bounds information provided by unranked spiders and shading.

Rule 4 (*Separate rank and bounds*). Let d be a unitary α -spider diagram of order. The diagram d can be decomposed into the conjunction of d_1 and d_2 where d_1 contains the ordered spider feet from d and no shading. Furthermore, d_2 contains an unranked spider foot for each spider foot in d and each shaded zone in d is also shaded in d_2 . Then d can be replaced by diagram $d_1 \wedge d_2$ and vice versa.

Example 4. Let d be the diagram in Fig. 5(a). A single application of rule 4 separate rank and bounds produces $d_{\alpha_o} \wedge d_{\alpha}$ in Fig. 5(b). The ordered spider feet are separated from the bounds information as the diagram d_1 contains only the order information, provided by ordered spider feet in d . The diagram d_2 contains the bounds information provided by both the shading and spiders in d . For each of the spiders in d there exists an unranked spider in d_2 with the same habitat.

Formal description of rule 4. Let $d (\neq \perp)$ be a unitary α -diagram of order in Venn-form containing all contours in \mathcal{C} . Let d_1 and d_2 be diagrams such that

$$\begin{aligned} C(d_1) &= C(d_2) = C(d), \\ Z(d_1) &= Z(d_2) = Z(d), \\ ShZ(d_1) &= \emptyset, \\ ShZ(d_2) &= ShZ(d), \end{aligned}$$

and

$$SI(d_1) = \{(n, \{(k, z)\}) : (n, \{(k, z)\}) \in SI(d) \wedge k \neq \bullet\},$$

$$SI(d_2) = orderErase(SI(d)),$$

where

$$orderErase(SI(d))$$

$$= \left\{ (n, \{(\bullet, z)\}) : n > 0 \wedge n = \sum_{(m, \{(k, z)\}) \in SI(d)} m \right\}.$$

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

Our proof of the soundness of Rule 4 proceeds by first presenting a series of lemmas. Each lemma corresponds to a step demonstrated in Fig. 6. The diagram d is semantically equivalent to the diagram $d \wedge d_2$ by Lemma 1 below. Lemma 2 will show that $d \wedge d_2$ is equivalent to $d_3 \wedge d_2$; i.e. we may drop shading from d without changing the meaning of the diagram $d \wedge d_2$. Finally, Lemma 3 will show that

$$d_3 \wedge d_2 \equiv d_4 \wedge d_2 \equiv d_5 \wedge d_2$$

i.e. we may remove all unranked spiders from d_3 , one at a time, without changing the meaning of the diagram. We observe that, in general, shading has no effect on the order information contained in a unitary diagram. We now show that the first step in this process holds.

Lemma 1. Let d be a unitary α -diagram of order in Venn-form containing all contours in \mathcal{C} . Let d_2 be a diagram where

$$\begin{aligned} C(d_2) &= C(d) \\ Z(d_2) &= C(d) \\ ShZ(d_2) &= ShZ(d) \\ SI(d_2) &= orderErase(SI(d)). \end{aligned}$$

Then $d \equiv d \wedge d_2$.

Returning to Fig. 6, Lemma 1 shows that d is semantically equivalent to $d \wedge d_2$. We now show that $d \wedge d_2$ is semantically equivalent to $d_3 \wedge d_2$. Here, d_3 is obtained from d by removing the shading.

Lemma 2. Let $d \wedge d_2$ where $d \neq \perp$ and $d_2 \neq \perp$ be a spider diagram of order where d is a unitary α -diagram of order in Venn form containing all contours in \mathcal{C} and where

$$\begin{aligned} C(d_2) &= C(d) \\ Z(d_2) &= C(d) \\ ShZ(d_2) &= ShZ(d) \\ SI(d_2) &= orderErase(SI(d)). \end{aligned}$$

Let d_3 be a copy of d where $ShZ(d_3) = \{\}$. Then $d \wedge d_2 \equiv d_3 \wedge d_2$.

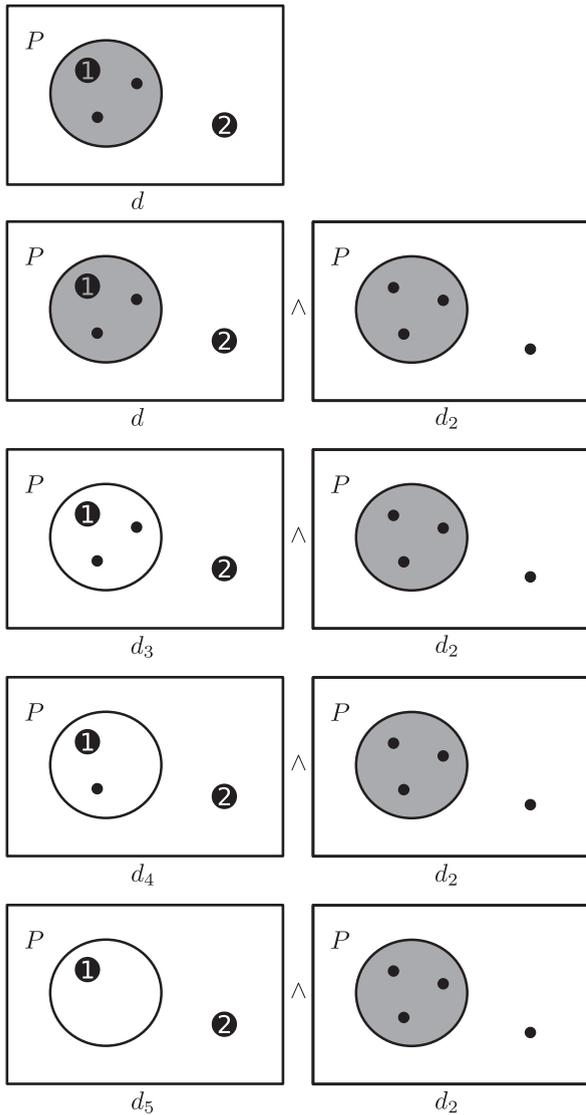


Fig. 6. The separating the rank and bounds rule.

We now show that where rank information has been separated from bounds information we can remove unranked spiders. This is illustrated in Fig. 6, where $d_3 \wedge d_2$ becomes $d_4 \wedge d_2$, by removing an unranked spider.

Lemma 3. Let $d_3 \wedge d_2$ be a spider diagram of order where

- d_3 is a unitary α -spider diagram of order in Venn form containing all contours from C and $ShZ(d_3) = \{\}$,
- d_2 is a unitary α -spider diagram of order in Venn form containing all contours from C containing only unranked spider feet, and
- there exists a habitat preserving injective function $\pi: S(d_3) \rightarrow S(d_2)$.

Let d_4 be a copy of d_3 where one of the unranked spiders $s = (n, \{(\bullet, z)\})$ in d_4 is removed; i.e. $S(d_4) = S(d_3) \ominus \{(\bullet, z)\}$. Then $d_3 \wedge d_2 \equiv d_4 \wedge d_2$.

Having demonstrated that we can remove a single unranked spider, we can repeatedly remove such spiders from diagrams like d_3 in Fig. 6 until no unranked spiders remain. We use this observation in the proof of the next theorem.

Theorem 4. Let d be a unitary α -diagram of order where $d_1 \wedge d_2$ is the result of applying rule 4 separate rank and bounds rule to d . Then $d \equiv d_1 \wedge d_2$.

Given a diagram d that contains only rank information (such a diagram is generated by an application of separate rank and bounds), the factor lowest spiders rule allows us to factor the different ranks out into a product of diagrams, where each unitary component of the product contains spiders of the same rank. We will subsequently show that ranked spider feet may be substituted by unranked spider feet given a unitary diagram containing only spiders of the same rank.

Rule 5 (Factor lowest spiders). Let d be a unitary α -diagram containing only spiders whose feet are ranked and containing no shaded zones. Then d may be replaced by $d_k \wedge d'$ where d_k contains those spider feet of lowest rank in d and d' contains all other spider feet.

Example 5. Let d_{α_1} be the diagram in Fig. 7(a) and $d_{\alpha_1} \wedge d_{\alpha_2}$ be the diagram in Fig. 7(b). We factor d_{α_1} into $d_{\alpha_1}^1$ and $d_{\alpha_2}^1$ where $d_{\alpha_1}^1$ contains all the lowest ranked spider feet from d_{α_1} ; i.e. those spider feet labelled 1. The diagram $d_{\alpha_1}^1 \wedge d_{\alpha_2}^1 \equiv d_{\alpha_1}$.

Formal description of rule 5. Let d be a unitary α -diagram containing only spiders whose feet are ranked and containing no shaded zones. Let k be the lowest rank of these feet. Let d_k and d' be diagrams such that

$$\begin{aligned} C(d_k) &= C(d') = C(d) \\ Z(d_k) &= Z(d') = Z(d) \\ ShZ(d_k) &= ShZ(d') = ShZ(d) = \{\} \end{aligned}$$

and

$$\begin{aligned} SI(d_k) &= \{(n, (k, z)) : (n, (k, z)) \in SI(d)\} \\ SI(d') &= SI(d) - SI(d_k). \end{aligned}$$

Then d may be replaced by $d_k \wedge d'$ and vice versa.

Theorem 5. Let $d (\neq \perp)$ be a unitary α -diagram containing only spiders whose feet are ranked and containing no shaded zones. Let $d \equiv d_k \wedge d'$ be the result of applying rule 5 factor lowest spiders to d . Then $d \equiv d_k \wedge d'$.

Rule 6 (Drop spider-foot rank). Let d be a unitary α -diagram such that each foot of each spider in d is of rank $k \in \mathbb{Z}^+$. Then d may be replaced by a diagram d' where d' is a copy of d and each foot of each spider foot in d' is unranked.

Example 6. Let d be the diagram in Fig. 8(a), in which all spiders contain the same rank spider feet. Then d may be replaced by d' in Fig. 8(b).

Formal description of rule 6. Let d be a unitary α -diagram and $k \in \mathbb{Z}^+$ where

$$(n, \{(j, z)\}) \in SI(d) \Rightarrow j = k.$$

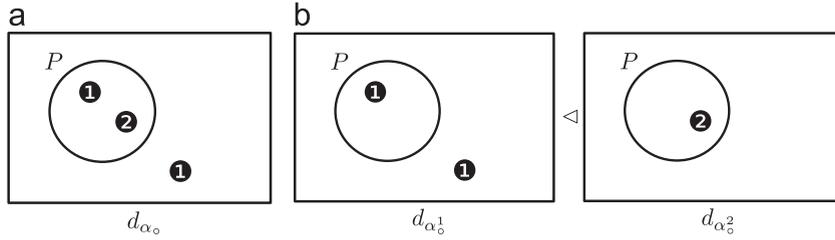


Fig. 7. The factor lowest spiders rule.

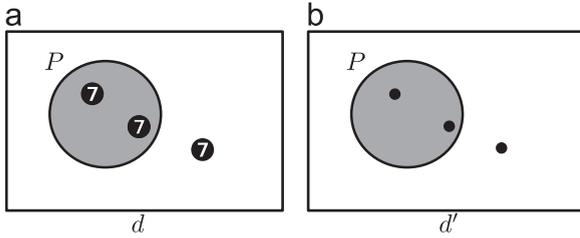


Fig. 8. The drop spider foot rank rule.

The diagram d may be replaced by d' where

$$C(d') = C(d),$$

$$Z(d') = Z(d),$$

$$ShZ(d') = ShZ(d),$$

and

$$SI(d') = \{(n, \{(\bullet, z)\}) : (n, \{(k, z)\}) \in SI(d)\}.$$

Theorem 6. Let $d (\neq \perp)$ be a unitary α -diagram where all spiders contain only feet of rank k . Let d' be the diagram produced by application of rule 6 to d . Then $d \equiv_{\models} d'$.

Our final rule, the rule of replacement, allows us to replace any spider diagram of order which is a sub-diagram in a compound expression with a semantically equivalent diagram. The purpose of this rule is to allow a sub-diagram in a compound expression to be replaced by the result of application of a reasoning rules to that sub-diagram. We first define a sub-diagram. We observe that the syntax of a spider diagram of order is defined by the following grammar in Backus–Naur form:

diagram $:: = \langle \text{unitary_diagram} \rangle \mid \text{conjunction} \mid \text{disjunction} \mid \text{negation} \mid \text{product};$

conjunction $:: = \text{diagram} \wedge \text{diagram};$

disjunction $:: = \text{diagram} \vee \text{diagram};$

negation $:: = \neg \text{diagram};$

product $:: = \text{diagram} \langle \text{diagram} \rangle;$

Given any spider diagram of order we may now construct its abstract syntax tree. Each tree contains unitary spider diagrams of order at leaf nodes and compound operators at non-leaf nodes.

The set of all abstract syntax trees is T and the set of all spider diagrams of order is Δ .

Theorem 7. Let D be a spider diagram of order. There exists a unique abstract syntax tree t and bijective function $\delta: \Delta \rightarrow T$ such that $\delta(D) = t$.

Let D be a spider diagram of order with abstract syntax tree t where t contains a non-leaf node r . The tree t_r with root node r is a sub-tree of t . Furthermore, $\delta^{-1}(t_r)$ is a sub-diagram of D .

Not only do we need to define what a sub-tree is, but we also need to know when two diagrams are syntactically equivalent. The following two definitions define transformational equivalence.

Definition 10 (Adapted from Molina [13], Stapleton [16]). Let D_1 and D_2 be spider diagrams of order. Then $D_1 \Vdash D_2$ if and only if D_1 can be transformed into D_2 by applying one of the reasoning rules given in this section. We say that D_2 is **obtainable** from D_1 , denoted $D_1 \vdash D_2$, if and only if there is a sequence of diagrams $\langle D^1, D^2, \dots, D^m \rangle$ such that $D^1 = D_1, D^m = D_2$ and $D^{k-1} \Vdash D^k$ for each $1 \leq k \leq m$.

Definition 11 (Adapted from Molina [13], Stapleton [16]). Let D_1 and D_2 be spider diagrams of order. If $D_1 \vdash D_2$ and $D_2 \vdash D_1$ then D_1 and D_2 are syntactically equivalent denoted, $D_1 \equiv_{\vdash} D_2$.

We may now define the rule of replacement.

Rule 7 (Rule of replacement). Let D and D' be spider diagrams of order. Let D_r be a sub-diagram of D , where D_r is syntactically equivalent to D' . Then an instance of D_r in D may be replaced by D' .

Example 7. In the diagram $d_1 \vee d_2$ in Fig. 9(a) we apply a rule to d_1 such that $d_1 \vdash d_x \vee d_y$ (specifically, the split spiders rule). Then we may replace d_1 by $d_x \vee d_y$ as seen in Fig. 9(b).

Formal description of rule 7. Let D_1 and D' be spider diagrams of order where D_r is a sub-diagram of D_1 . If $D_r \equiv_{\vdash} D'$ then any occurrence of the subtree $\delta(D_r)$ in $\delta(D_1)$ may be replaced by $\delta(D')$ to produce D_2 . Then D_1 can be replaced by D_2 and vice versa.

The soundness of the rule of replacement is given in the following theorem:

Theorem 8. Let D_1 and D_2 be spider diagrams of order where D_2 is obtained from D_1 by application of rule 7 rule of replacement. Then D_1 is semantically equivalent to D_2 .

Each of our seven reasoning rules is sound. However, our reasoning system is incomplete. Previous approaches to showing completeness of spider diagram based reasoning systems do not readily generalise to spider diagrams of order. As a first step in developing a complete reasoning system we produce a normal form for spider diagrams of order. In the next section we provide an algorithm that,

given an arbitrary spider diagram of order, produces a spider diagram of order in our normal form.

4. An algorithm to produce diagrams in normal form

In this section we define a normal form for spider diagrams of order. Our normal form allows the diagram \perp .

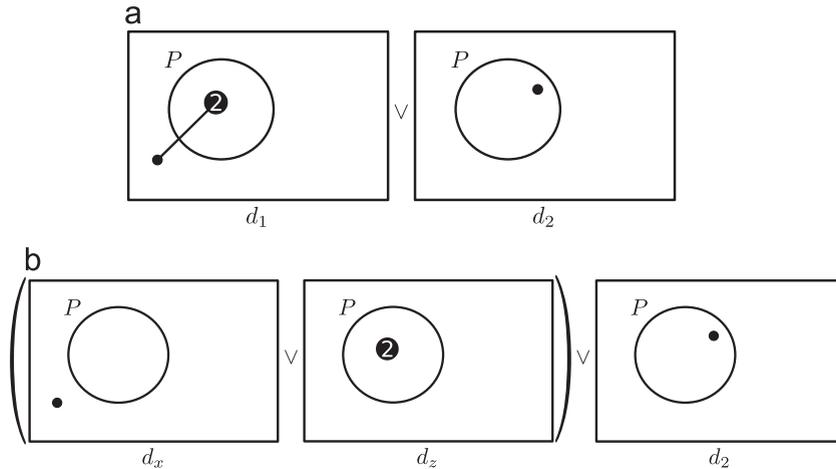


Fig. 9. The rule of replacement.

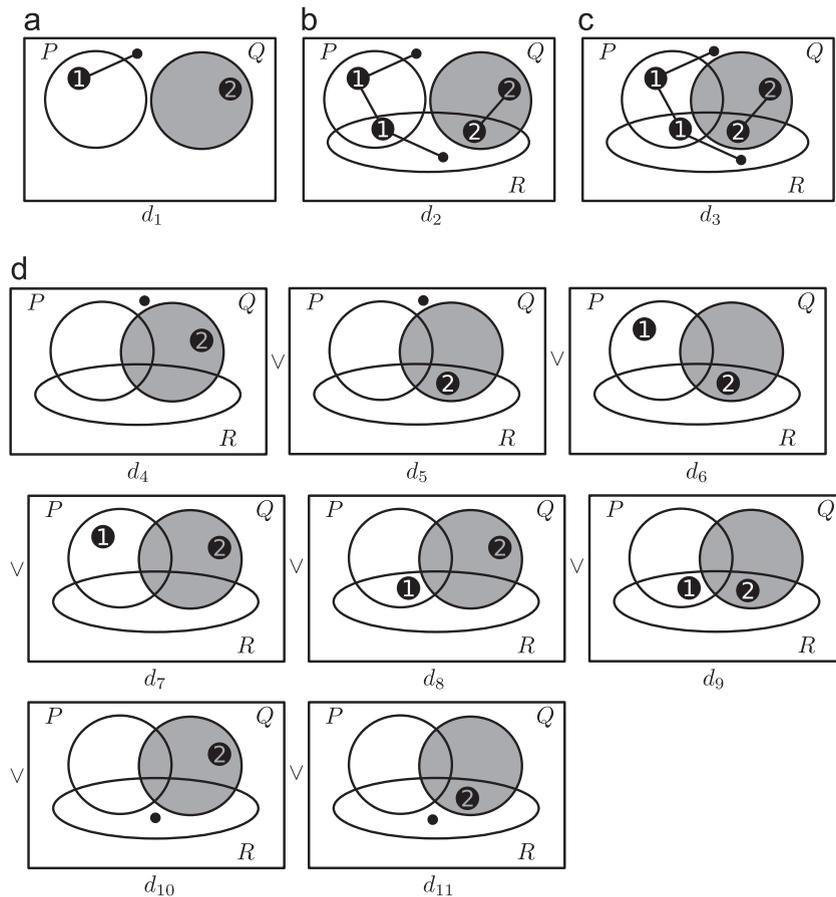


Fig. 10. Applying reasoning rules resulting in a disjunction of α -diagrams.

Furthermore, compound diagrams are formed from unitary α -spider diagrams in Venn-form containing all contours in \mathcal{C} and no ranked spiders. Compound expressions in normal form allow \wedge , \vee and \triangleleft as connectives and \neg as the unary operator. From the 7 reasoning rules, presented in the previous section, we show that any spider diagram of order is semantically equivalent to a diagram in our normal form. Specifically, we define a sequence of applications of reasoning rules for producing a diagram in normal form given an arbitrary spider diagram of order.

Definition 12. Let D be a spider diagram of order. It is the case that D is in *normal form* if the following conditions hold:

- No unitary component of D contains ranked spider feet.
- Each unitary component of D is an α -diagram and contains all contours in \mathcal{C} , or is \perp .
- There are no zones missing from any unitary component ($\neq \perp$) of D .
- The binary connectives \wedge , \vee and \triangleleft and the unitary connective \neg are the allowed connectives in D .

Given a spider diagram of order as input, the algorithm produces a spider diagram of order in normal form as output. The algorithm is outlined as follows, where applications of **rule 7**, the rule of replacement, are implicitly assumed:

- Let D be the input diagram.
- Apply **rule 1** to each unitary diagram in D , producing $D_{\mathcal{C}}$, until all contours in \mathcal{C} are present in the result $D_{\mathcal{C}}$.
- Apply **rule 2** to each unitary diagram in $D_{\mathcal{C}}$, producing $D_{\mathcal{Z}}$, until there are no missing zones in the result $D_{\mathcal{Z}}$.
- Apply **rule 3** to each unitary diagram in $D_{\mathcal{Z}}$ until there are no spiders with multiple feet in the result D_{α} .
- Apply **rule 4** to each unitary diagram in D_{α} , producing $D_{\alpha'}$.
- Apply **rule 5** to each unitary diagram in $D_{\alpha'}$ until every unitary diagram either contains no spider feet or spider feet of all the same rank.
- Apply **rule 6** to each unitary diagram in $D_{\alpha'}$, producing D_{\bullet} , the final result.

We now present an example of a unitary spider diagram, its corresponding normal form diagram and an

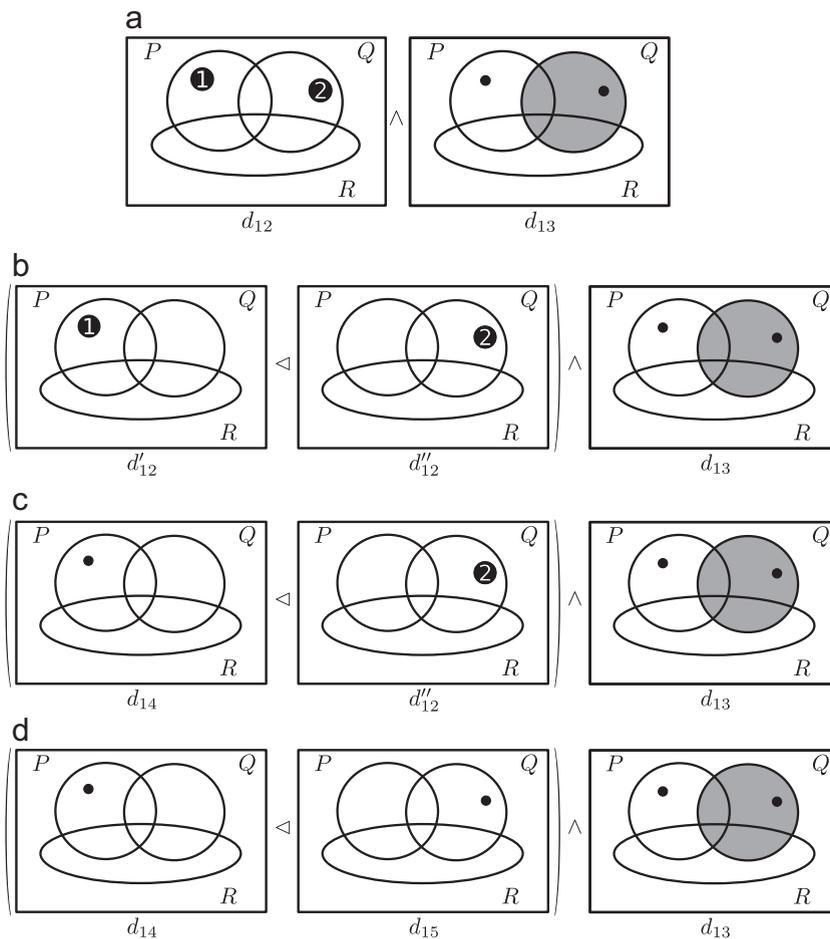


Fig. 11. Applying reasoning rules resulting in a diagram in normal form.

illustration of the algorithm to generate the required normal form.

Example 8. Let d_1 be the unitary spider diagram of order in Fig. 10(a) and let $C = \{P, Q, R\}$. The diagram d_1 contains the contours P and Q and the zones $(\{\}, \{P, Q\})$, $(\{P\}, \{Q\})$ and $(\{Q\}, \{P\})$. It is not in Venn form as the zone $(\{P, Q\}, \{\})$ is missing. Furthermore, the contour R is not present in d_1 and the diagram contains a spider with more than one foot. Our strategy is to first add all the missing contours to d_1 . The addition of R to d gives the diagram in Fig. 10(b). We then add all the zones that are missing from 10(b) forming 10(c). The addition of the zones $(\{P, Q\}, \{R\})$ and $(\{P, Q, R\}, \{\})$ to the diagram can be seen in Fig. 10(c). The diagram in Fig. 10(c) is in Venn form and contains no missing zones. It is semantically equivalent to d_1 . From this diagram in Venn form containing all contours in C we repeatedly employ a split spiders reasoning rule to produce a disjunction of α -diagrams. The diagram in Fig. 10(d) is a disjunction of α -diagrams where each unitary component contains all contours in C . It is semantically equivalent to d_1 .

Let d_7 (an arbitrary choice) be the unitary component of the diagram in Fig. 10(d) as annotated in the figure. We now show that d_7 and, by extension, any unitary α -diagram may be transformed, by application of reasoning rules, into a diagram in our normal form. We first separate order and shading resulting in diagram $d_{12} \wedge d_{13}$ in Fig. 11(a). Thereafter, we factor lowest spiders from d_{12} and replace d_{12} with $d'_{12} \wedge d''_{12}$ in Fig. 11(b). Finally, the precondition for the drop spider foot order rule is satisfied, so we drop the ordered foot from d'_{12} resulting in d_{14} in Fig. 11(a). Applying the drop spider foot order rule to d''_{12} and replacing the result into $(d_{14} \triangleleft d''_{12}) \wedge d_{13}$ yields the diagram $(d_{14} \triangleleft d_{15}) \wedge d_{13}$ in normal form in Fig. 11(d).

5. Conclusion

We have presented a reasoning system and normal form for spider diagrams of order. Spider diagrams of order are an interesting recent advance in Euler-diagram based visual logics as they incorporate an order relation into their semantics. Furthermore, it is known that spider diagrams of order are as expressive as star-free regular languages and that star-free regular languages are as expressive as linear temporal logic. In the future, we wish to use the normal form, developed in this paper, to directly compare spider diagrams of order and linear temporal logic.

We view our algorithm for obtaining normal form to be the first step towards providing a completeness result for spider diagrams of order. This is because the completeness proofs for existing spider diagram logics, such as [10], as

well as their extension called constraint diagrams [11], rely on obtaining diagrams with the property that all spiders have single feet [2]. This property is delivered by our normal form. Once in this normal form, the completeness proofs use other rules to establish syntactic entailment. As it stands, the spider diagram of order logic does not have sufficient rules to establish completeness. Finding a complete set of rules is an interesting prospect for future work because it will provide insight into how to gain completeness when an order operator is present, contributing to our understanding of diagrammatic logics in general.

References

- [1] P. Bottoni, A. Fish, Policy enforcement and verification with timed modeling spider diagrams, in: IEEE Symposium on Visual Languages and Human-Centric Computing, 2013, pp. 27–34.
- [2] J. Burton, G. Stapleton, J. Howse, Completeness proof strategies for Euler diagram logics, in: Theory and Application of Diagrams, 2012.
- [3] P. Chapman, G. Stapleton, A. Delaney, On the expressiveness of second-order spider diagrams, *J. Vis. Lang. Comput.* 24 (5) (2013) 327–349.
- [4] L. Choudhury, M.K. Chakraborty, On extending Venn diagrams by augmenting names of individuals, in: Theory and Application of Diagrams, 2004, pp. 142–146.
- [5] J. Cohen, D. Perrin, J.-E. Pin, On the expressive power of temporal logic, *Comput. Syst. Sci.* 46 (1993) 271–294.
- [6] A. Delaney, Defining star-free regular languages using diagrammatic logic (Ph.D. thesis), University of Brighton, 2012.
- [7] H. Dunn-Davies, R. Cunningham, Propositional statecharts for agent interaction protocols, in: Proceedings of Euler Diagrams 2004, Brighton, UK, ENTCS, vol. 134, 2005, pp. 55–75.
- [8] H.-D. Ebbinghaus, J. Flum, *Finite Model Theory*, 2nd ed. Springer, Berlin, 1991.
- [9] Andrew Fish, John Taylor, Equivalences in Euler-based diagram systems through normal forms, *LMS J. Comput. Math.* 17 (2014).
- [10] J. Howse, G. Stapleton, J. Taylor, Spider diagrams, *LMS J. Comput. Math.* 8 (2005) 145–194.
- [11] S. Kent, Constraint diagrams: visualizing invariants in object oriented modelling, in: International Conference on Object Oriented Programming, Systems, Languages and Applications, ACM Press, New York, October 1997, pp. 327–341.
- [12] M. Leucker, C. Sanchez, Regular linear temporal logic, in: C. Jones, Z. Liu, J. Woodcock (Eds.), *Theoretical Aspects of Computing ICTAC 2007*, Lecture Notes in Computer Science, vol. 4711, Springer, Berlin, 2007, pp. 291–305.
- [13] F. Molina, Reasoning with extended Venn–Peirce diagrammatic systems (Ph.D. thesis), University of Brighton, 2001.
- [14] B. Nagy, S. Vályi, Visual reasoning by generalized interval-values and interval temporal logic, in: VLL, 2007, pp. 13–26.
- [15] S.-J. Shin, *The Logical Status of Diagrams*, Cambridge University Press, Cambridge, UK, 1994.
- [16] G. Stapleton, Reasoning with constraint diagrams (Ph.D. thesis), University of Brighton, August 2004.
- [17] G. Stapleton, A. Delaney, Evaluating and generalizing constraint diagrams, *J. Vis. Lang. Comput.* 19 (August) (2008) 499–521.
- [18] G. Stapleton, J. Howse, P. Chapman, A. Delaney, J. Burton, I. Oliver, Formalizing concept diagrams, in: Distributed Multimedia Systems, Knowledge Systems Institute, 2013, pp. 182–187.
- [19] G. Stapleton, J. Howse, J. Taylor, A decidable constraint diagram reasoning system, *J. Logic Comput.* 15 (December (6)) (2005) 975–1008.