The Expressiveness of Spider Diagrams

GEM STAPLETON, JOHN HOWSE and JOHN TAYLOR, *Visual Modelling Group, School of Computing, Mathematical and Information Sciences, University of Brighton, Brighton, BN2 4GJ, UK.*
E-mail: {g.e.stapleton,john.howse,john.taylor}@brighton.ac.uk

SIMON THOMPSON, *University of Kent, Canterbury, Kent, CT2 7NF UK.*
E-mail: s.j.thompson@kent.ac.uk

Abstract

Spider diagrams are a visual language for expressing logical statements. In this paper we identify a well-known fragment of first-order predicate logic, that we call $\mathcal{MFOL}_=$, equivalent in expressive power to the spider diagram language. The language $\mathcal{MFOL}_=$ is monadic and includes equality but has no constants or function symbols. To show this equivalence, in one direction, for each diagram we construct a sentence in $\mathcal{MFOL}_=$ that expresses the same information. For the more challenging converse we prove that there exists a finite set of models for a sentence S that can be used to classify all the models for S. Using these classifying models we show that there is a diagram expressing the same information as S.

Keywords: Spider diagrams, expressiveness, monadic logic, model theory.

1 Introduction

Euler diagrams [5] exploit topological properties of enclosure, exclusion and intersection to represent subset, disjoint sets and set intersection respectively. The diagram d_1 in Figure 1 is an Euler diagram and asserts that nothing is both a car and a van. Venn diagrams [17] are similar to Euler diagrams. In Venn diagrams, all possible intersections between contours must occur and shading is used to represent the empty set. The diagram d_2 in Figure 1 is a Venn diagram and also expresses that no element is both a car and a van.

Various visual languages have emerged that extend Euler and Venn diagrams. Peirce [14] increased the expressiveness of Venn diagrams by adding $\otimes$-*sequences*. The presence of an $\otimes$-sequence indicates the existence of an element. The Venn-II system, introduced by Shin [15], consists of Venn diagrams together with $\otimes$-sequences. The diagram d_3 in Figure 1 is a Venn-II diagram. In addition to the information which is expressed by the underlying Venn diagram, it also asserts that the set $Cars \cup Vans$ is not empty. In Venn-II, diagrams are joined by straight line segments to represent disjunction between diagrams. Venn-II diagrams can express whether a set is empty or not empty. Shin shows that Venn-II is equivalent in expressive power to a first order language that she calls L_0. The language L_0 is a pure monadic language (i.e. all the predicate symbols are 'one place') that does not include constants or function symbols.

Another visual language, called Euler/Venn, based on Euler diagrams is discussed by Swoboda and Allwein in [16]. These diagrams are similar to Venn-II diagrams but, instead of $\otimes$-sequences, *constant sequences* are used. The diagram d_4 in Figure 2 is an Euler/Venn diagram and asserts that no element is both a car and a van and that there is something called 'ford' that is either a car or a van. Swoboda and Allwein give an algorithm that determines whether a given monadic first-order formula is 'observable' from a given diagram. If the formula is observable from the diagram then it is a consequence of the information contained in the diagram, but need not express all the information

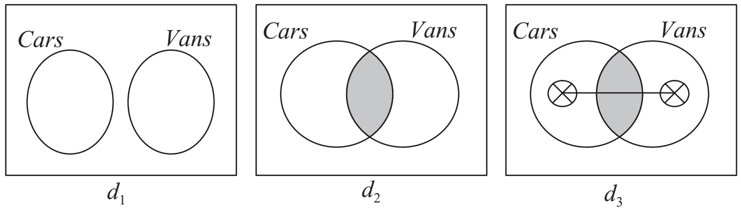


FIGURE 1. An Euler diagram, Venn diagram and a Venn-II diagram.

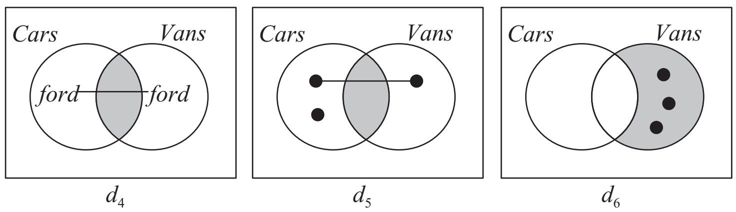


FIGURE 2. An Euler/Venn diagram and two spider diagrams.

in the diagram.

Like Euler/Venn diagrams, spider diagrams are based on Euler diagrams. Rather than allowing the use of constant sequences[1] as in Euler/Venn diagrams, *spiders* denote the existence of elements. Unlike the $\otimes$-sequences, distinct spiders denote distinct elements. The spider diagram d_5 in Figure 2 asserts that no element is both a car and a van and there are at least two elements, one is a car and the other is a car or a van. The spider diagram d_6 asserts that there are exactly three vans that are not cars. Spiders (by their existential import) allow a lower bound to be placed on the cardinality of sets. Shading allows upper bounds to be placed on the cardinality of sets.

Several sound and complete spider diagram systems have been developed [10, 11, 13]. A tool to support reasoning with spider diagrams has been developed, available from [18]. In [7] an algorithm is presented that, given any spider diagrams D_1 and D_2, either constructs a proof from D_1 to D_2, or provides a model for D_1 that is not a model for D_2. The proofs constructed by this algorithm tend to be long and unwieldy. In [6] a heuristic approach to proof writing in the spider diagram system is developed, but is restricted to the case of *unitary* spider diagrams. The authors invoke the A^* algorithm [2] to find a shortest proof, provided such a proof exists.

In this paper we prove that the spider diagram language is equivalent in expressive power to a fragment of first-order logic that we call $\mathcal{MFOL}_=$. The language $\mathcal{MFOL}_=$ extends L_0 by adding equality, so $\mathcal{MFOL}_=$ is monadic predicate logic with equality. Within L_0 it is not possible to express that a particular property, P, holds for a unique element:

$$\exists x \ (P(x) \land \forall y \ (P(y) \Rightarrow x = y)).$$

Thus spider diagrams increase expressiveness over Venn-II.

Although we do not include constants in $\mathcal{MFOL}_=$ or given spiders (to represent constants) in our spider diagram language, this is not a significant restriction. It is relatively straightforward to show that adding constants to either of these languages does not lead to an increase in expressiveness.

[1] In some spider diagram languages, **given spiders** [10] represent constants but for our purposes spiders represent existential quantification.

However, the omission of function symbols is more significant: the standard elimination of function symbols in terms of relation symbols relies upon binary predicate symbols which we do not have.

In Section 2 we give the syntax and semantics of spider diagrams. We define $\mathcal{MFOL}_=$ in Section 3. In Section 4 we identify when a diagram and a sentence express the same information. We address the task of mapping each diagram to a sentence expressing the same information in Section 5, showing that the spider diagram language is at most as expressive as $\mathcal{MFOL}_=$. In Section 6 we show that $\mathcal{MFOL}_=$ is at most as expressive as spider diagrams. We will outline Shin's algorithmic approach to show L_0 (in which there is no equality) is not more expressive than Venn-II. It is simple to adapt this algorithm to find a spider diagram that expresses the same information as a sentence in $\mathcal{MFOL}_=$ that does not involve equality. However, for sentences in $\mathcal{MFOL}_=$ that do involve equality, the algorithm does not readily generalize. Thus we take a different approach. To motivate our approach we consider relationships between models for diagrams. We consider the models for a sentence and show that there is a finite set of models that can be used to classify all the models for the sentence. These classifying models can then be used to construct a diagram that expresses the same information as the sentence.

2 Spider diagrams

In diagrammatic systems, it is helpful to distinguish two levels of syntax: concrete (or token) syntax and abstract (or type) syntax [9]. Concrete syntax captures the physical representation of a diagram. Abstract syntax 'forgets' semantically irrelevant spatial relations between syntactic elements in a concrete diagram. We include the concrete syntax to aid intuition but we work at the abstract level.

2.1 Informal concrete syntax

A **contour** is a simple closed plane curve. Each contour is labelled. Within a unitary diagram, the same label cannot be used twice. A **boundary rectangle** properly contains all contours. The boundary rectangle is not a contour and is not labelled. A **basic region** is the bounded area of the plane enclosed by a contour or a 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 and difference of r_1 and r_2 are regions provided these are non-empty. A **zone** is a region having no other region contained within it. A region is **shaded** if each of its component zones is shaded. A **spider** is a tree with nodes (called **feet**) placed in different zones. The connecting edges (called **legs**) are straight line segments. A spider **touches** a zone if one of its feet is placed in that zone. A spider is said to **inhabit** the region which is the union of the zones it touches. This union of zones is called the **habitat** of the spider.

A **concrete unitary spider diagram** is a single boundary rectangle together with a finite collection of contours, shading and spiders. No two contours in the same unitary diagram can have the same label. We place certain well-formedness conditions on unitary diagrams. We stipulate that each zone is connected. There must be at least one zone inside each contour (this follows from the fact that contours are simple closed plane curves). The boundary rectangle properly contains all contours, so there is a zone inside the boundary rectangle but outside all the contours.

EXAMPLE 2.1
Spider diagram d_6 in Figure 2 (Section 1) has two contours and four zones. The shaded zone is inhabited by three spiders, each with one foot.

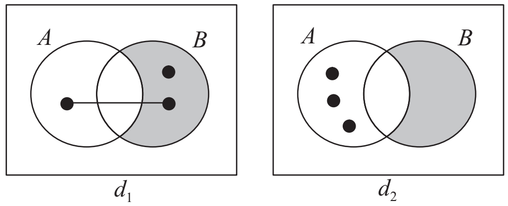


FIGURE 3. Two spider diagrams.

2.2 Formal abstract syntax

We can think of the contour labels used in our diagrams as being chosen from a countably infinite set, $\mathcal{L}$. A zone, at the concrete level, can be described by the set of labels of the contours that include it. When we reason with a spider diagram, its contour label set may change, so we will define an abstract zone to be a pair of sets, (a, b). The set a contains the labels of the contours that include (a, b) whereas b is the set of labels of the contours that do not include (a, b). So, a and b form a partition of the contour label set.

Now we consider how we represent spiders at the abstract level. In order to describe the spiders in a concrete diagram, it is sufficient to say how many spiders there are in each region. We could specify any finite set to be a collection of spiders, and map each of these spiders to a region in the diagram, giving its habitat. For any given concrete diagram, then, there would potentially be many choices for an abstract set of spiders. In order to give a unique abstraction from a concrete diagram we will use a bag of regions, called *spider identifiers*, rather than an arbitrary set of spiders.

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

1. $L = L(d) \subset \mathcal{L}$ is a finite set of contour labels.
2. $Z = Z(d) \subseteq \{(a, L - a) : a \subseteq L\}$ is a set of **zones** such that
   (i) for each label $l \in L$ there is a zone $(a, L - a) \in Z(d)$ such that $l \in a$ and
   (ii) the zone $(\emptyset, L)$ is in $Z(d)$.
3. $Z^* = Z^*(d) \subseteq Z$ is a set of **shaded zones**.
4. $SI = SI(d) \subset \mathbf{Z}^+ \times (\mathbf{P}Z - \{\emptyset\})$ is a finite set of **spider identifiers** such that

$$\forall (n_1, r_1), (n_2, r_2) \in SI \bullet r_1 = r_2 \Rightarrow n_1 = n_2$$

   If $(n, r) \in SI$ we say there are n **spiders** with **habitat** r.

Some remarks about the definition are in order. Every contour in a concrete diagram contains at least one zone and this is captured by condition 2 (i). In any concrete diagram, the zone inside the boundary rectangle but outside all the contours is present and this is captured by condition 2 (ii).

EXAMPLE 2.3
The diagram d_1 in Figure 3 has the following abstract description.

1. The set of contour labels is $L(d_1) = \{A, B\}$.
2. The set of zones is $Z(d_1) = \{(\emptyset, \{A, B\}), (\{A\}, \{B\}), (\{B\}, \{A\}), (\{A, B\}, \emptyset)\}$.
3. The set of shaded zones is $Z^*(d_1) = \{(\{B\}, \{A\})\}$.

4. The set of spider identifiers is

$$SI(d_1) = \{(1, \{(\{B\}, \{A\})\}), (1, \{(\{A\}, \{B\}), (\{B\}, \{A\})\})\}.$$

We define, for unitary diagram d, the **Venn zone set** to be

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

and the **missing zone set** to be $MZ(d) = VZ(d) - Z(D)$. If $Z(d) = VZ(d)$ then d is said to be in **Venn form**. If $z \in MZ(d)$ then z is **missing** from d. Missing zones represent the empty set.

Spiders represent the existence of elements and regions (an abstract region is a set of zones) represent sets – thus we need to know how many elements we have represented in each region. The number of spiders contained by region r_1 in d is denoted by $S(r_1, d)$. More formally,

$$S(r_1, d) = \sum_{(n, r_2) \in SI(d) \wedge r_2 \subseteq r_1} n.$$

So, any spider in d whose habitat is a subset of r_1 contributes to the sum $S(r_1, d)$. The number of spiders touching r_1 in d is denoted by $T(r_1, d)$. More formally,

$$T(r_1, d) = \sum_{(n, r_2) \in SI(d) \wedge r_2 \cap r_1 \neq \emptyset} n.$$

So, any spider in d that has a foot in r_1 contributes to the sum $T(r_1, d)$. In the diagram d_1, in figure 3, $S(\{(\{B\}, \{A\})\}, d_1) = 1$ and $T((\{(\{B\}, \{A\})\}, d_1) = 2$.

Unitary diagrams form the building blocks of **compound diagrams**. If D_1 and D_2 are spider diagrams then so are $\overline{D_1}$ ('not D_1'), $(D_1 \sqcup D_2)$ ('D_1 or D_2') and $(D_1 \sqcap D_2)$ ('D_1 and D_2'). Some diagrams are not satisfiable and we introduce the symbol $\perp$, defined to be a unitary diagram interpreted as false. Our convention will be to denote unitary diagrams by d and arbitrary diagrams by D.

2.3 Semantics

Regions in spider diagrams represent sets. We can express lower bounds and, in the case of shaded regions, upper bounds on the cardinalities of the sets that we are representing as follows. If region r contains n spiders in diagram d then d expresses that the set represented by r contains at least n elements. If r is shaded and touched by m spiders in d then d expresses that the set represented by r contains at most m elements. Thus, if d has a shaded, untouched region, r, then d expresses that r represents the empty set. Missing zones also represent the empty set. To formalize the semantics we shall map contour labels, zones and regions to subsets of some universal set. We define $\mathcal{Z}$ and $\mathcal{R}$ to be the sets of all abstract zones and abstract regions respectively. So,

$$\mathcal{Z} = \{(a, b) \in \mathbf{PF}(\mathcal{L}) \times \mathbf{PF}(\mathcal{L}) : a \cap b = \emptyset\}$$

where $\mathbf{PF}(\mathcal{L})$ denotes the set of all finite subsets of $\mathcal{L}$, and $\mathcal{R} = \mathbf{PF}(\mathcal{Z})$.

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

1. For each zone (a, b),
$$\Psi(a,b) = \bigcap_{l \in a} \Psi(l) \cap \bigcap_{l \in b} \overline{\Psi(l)}$$
where $\overline{\Psi(l)} = U - \Psi(l)$ and we define $\bigcap_{l \in \emptyset} \Psi(l) = U = \bigcap_{l \in \emptyset} \overline{\Psi(l)}$.
2. For each region r,
$$\Psi(r) = \bigcup_{z \in r} \Psi(z)$$
and we define $\Psi(\emptyset) = \bigcup_{z \in \emptyset} \Psi(z) = \emptyset$.

We introduce a *semantics predicate* which identifies whether a diagram expresses a true statement, with respect to an interpretation.

DEFINITION 2.5
Let D be a diagram and let $m = (U, \Psi)$ be an interpretation. We define the **semantics predicate** of D, denoted $P_D(m)$. If $D = \perp$ then $P_D(m)$ is $\perp$. If D $(\neq \perp)$ is a unitary diagram then $P_D(m)$ is the conjunction of the following three conditions.

1. **Distinct spiders condition.** For each region r in $\mathbf{P}Z(D) - \{\emptyset\}$,
$$|\Psi(r)| \geq S(r, D).$$
2. **Shading condition.** For each shaded region r in $\mathbf{P}Z^*(D) - \{\emptyset\}$,
$$|\Psi(r)| \leq T(r, D).$$
3. **Missing zones condition.** Any zone, z, in $MZ(D)$ satisfies $\Psi(z) = \emptyset$.

If $D = \overline{D_1}$ then $P_D(m) = \neg P_{D_1}(m)$. If $D = D_1 \sqcup D_2$ then $P_D(m) = P_{D_1}(m) \vee P_{D_2}(m)$. If $D = D_1 \sqcap D_2$ then $P_D(m) = P_{D_1}(m) \wedge P_{D_2}(m)$. We say m **satisfies** D, denoted $m \models D$, if and only if $P_D(m)$ is true. If $m \models D$ we say m is a **model** for D.

EXAMPLE 2.6
Defining $\Psi(A) = \{1\}$ and $\Psi(B) = \{2\}$ characterizes the interpretation $m = (\{1,2\}, \Psi)$ which is a model for d_1 in figure 3 but not for d_2.

3 The language $\mathcal{MFOL}_=$

Spider diagrams do not have syntactic elements to represent constants or functions. We can express statements of the form 'there are at least n elements in A' and 'there are at most m elements in A'. A first-order language equivalent in expressive power to the spider diagram language will involve equality, to allow us to express distinctness of elements, and monadic predicates, to allow us to express $x \in A$. In order to define such a language we require a countably infinite set of monadic predicate symbols, $\mathcal{P}$, from which all monadic predicate symbols will be drawn. Moreover, we also require a countably infinite set of variables, $\mathcal{V}$, from which all variables will be drawn.

DEFINITION 3.1
The first-order language $\mathcal{MFOL}_=$ consists of the following.

1. **Atomic formulae** which are defined as follows,

(a) if x_i and x_j are variables then $(x_i = x_j)$ is an atomic formula,
(b) if $P_i \in \mathcal{P}$ and x_j is a variable then $P_i(x_j)$ is an atomic formula.

2. **Formulae**, which are defined inductively.
(a) Atomic formulae are formulae.
(b) $\perp$ and $\top$ are formulae.
(c) If p and q are formulae so are $(p \wedge q)$, $(p \vee q)$ and $\neg p$.
(d) If p is a formula and x_j is a variable then $(\forall x_j\, p)$ and $(\exists x_j\, p)$ are formulae.

We define $\mathcal{F}$ and $\mathcal{S}$ to be the sets of formulae and sentences (formulae with no free variables) of the language $\mathcal{MFOL}_=$ respectively.

We shall assume the standard first-order predicate logic semantic interpretation of formulae in this language (see, for example, [1]) with one exception: we allow a structure to have an empty domain. Logic with potentially empty structures is explored in [8, 12]. The motivation for this non-standard choice comes from the intended application domain for spider diagrams: modelling object oriented software systems. The domain will consist of the objects in the system and in some instances there will be no objects (for example, in an initial state before any objects have been created).

4 Structures and interpretations

We wish to identify when a diagram and a sentence express the same information. To aid us formalize this notion, we map interpretations to structures in such a way that information is preserved. For this discussion we fix the set of labels $\mathcal{L} = \{L_1, L_2, ...\}$ and the set of monadic predicate symbols $\mathcal{P} = \{P_1, P_2, ...\}$. We identify corresponding labels and predicates L_i and P_i. We also fix $\mathcal{V} = \{x_1, x_2, ...\}$. Define $\mathcal{U}$ to be the class of all sets. The sets in $\mathcal{U}$ form the domains of structures in the language $\mathcal{MFOL}_=$.

DEFINITION 4.1
Define $\mathcal{INT}$ to be the class of all interpretations for spider diagrams over $\mathcal{L}$, that is

$$\mathcal{INT} = \{(U, \Psi) : U \in \mathcal{U} \wedge \Psi\colon \mathcal{L} \cup \mathcal{Z} \cup \mathcal{R} \to \mathbf{P}U\},$$

where (U, Ψ) is an interpretation. Define also $\mathcal{STR}$ to be the class of structures for the language $\mathcal{MFOL}_=$ over $\mathcal{P}$, that is

$$\mathcal{STR} = \{m : U \in \mathcal{U} \wedge m = \langle U, =^m, P_1^m, P_2^m, ...\rangle\},$$

where P_i^m is the interpretation of P_i in the structure m (that is, $P_i^m \subseteq U$) and we always interpret $=$ as the diagonal subset of $U \times U$, denoted $diag(U \times U)$.

LEMMA 4.2
The function, $h\colon \mathcal{INT} \to \mathcal{STR}$ defined by

$$h(U, \Psi) = \langle U, diag(U \times U), \Psi(L_1), \Psi(L_2), ...\rangle$$

is a bijection.

Essentially, $h(U, \Psi)$ is just a different way of writing (U, Ψ). Our aim is to identify, for each diagram, a sentence that expresses the same information. We also aim, for each sentence, to identify a diagram that expresses the same information and we now formalize this notion. A diagram and a sentence express the same information when h provides a bijective correspondence between their models, illustrated in Figure 4.

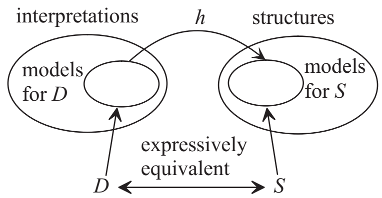


FIGURE 4. A model-level relationship between expressively equivalent diagrams and sentences.

DEFINITION 4.3
Let D be a diagram and S be a sentence. We say D and S are **expressively equivalent** if and only if

$$\{h(p) : p \in \mathcal{INT} \wedge p \models D\} = \{m \in \mathcal{STR} : m \models S\}.$$

So, a diagram and a sentence are expressively equivalent if they have essentially the same models.

5 Mapping from diagrams to sentences

To show that the spider diagram language is not more expressive than $\mathcal{MFOL}_=$ we will map diagrams to expressively equivalent sentences. An $\boldsymbol{\alpha}$**-diagram** is a spider diagram in which all spiders inhabit exactly one zone [13].

THEOREM 5.1
Every spider diagram is semantically equivalent to an α-diagram [11].

PROOF. **(Sketch)** Spider legs represent disjunction within a unitary diagram, d. Therefore, if there is a spider, s, in d that inhabits region $r_1 \cup r_2$ where $r_1 \cap r_2 = \emptyset$ then d is semantically equivalent to $d_1 \sqcup d_2$ where each of d_1 and d_2 are copies of d except that s inhabits r_1 in d_1 and r_2 in d_2, thus removing a spider's leg. This process of *splitting spiders* can be repeated until all spiders inhabit exactly one zone. ∎

It follows that to show that the spider diagram language is at most as expressive as $\mathcal{MFOL}_=$ it is sufficient to identify an expressively equivalent sentence for each α-diagram.

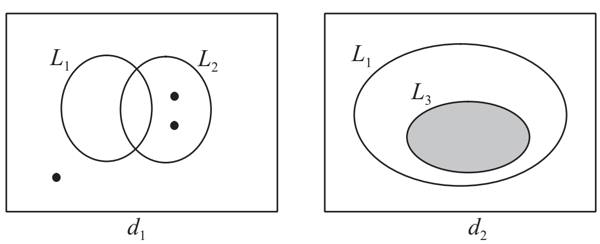


FIGURE 5. Two α-diagrams: from diagrams to sentences.

EXAMPLE 5.2
The diagram d_1 in Figure 5 contains three spiders, one outside both L_1 and L_2, the other two inside L_2 and outside L_1 and is expressively equivalent to the sentence

$$\exists x_1\,(\neg P_1(x_1) \wedge \neg P_2(x_1)) \wedge \exists x_1 \exists x_2\,(P_2(x_1) \wedge P_2(x_2) \wedge \neg P_1(x_1) \wedge \neg P_1(x_2) \wedge x_1 \neq x_2).$$

The diagram d_2 asserts that no elements can be in L_3 and not in L_1 (due to the missing zone) and no element can be in both L_1 and L_3 (due to the shading) and is expressively equivalent to the sentence

$$\forall x_1 \neg(P_3(x_1) \wedge \neg P_1(x_1)) \wedge \forall x_1 \neg(P_1(x_1) \wedge P_3(x_1)).$$

To construct sentences for diagrams, it is useful to map zones to formulae as follows.

DEFINITION 5.3
Define a function to map zones to formulae, $\mathcal{ZF}: \mathcal{Z} \times \mathcal{V} \rightarrow \mathcal{F}$ ($\mathcal{ZF}$ for 'zone formula') by, for each $(a,b) \in \mathcal{Z} - \{(\emptyset,\emptyset)\}$ and variable x_j,

$$\mathcal{ZF}((a,b),x_j) = \bigwedge_{L_k \in a} P_k(x_j) \wedge \bigwedge_{L_k \in b} \neg P_k(x_j)$$

and

$$\mathcal{ZF}((\emptyset,\emptyset),x_j) = \top.$$

We use the function $\mathcal{ZF}$ to construct a sentence of $\mathcal{MFOL}_=$ for each zone in a unitary α-diagram. We shall take these *zone sentences* in conjunction to identify a sentence expressively equivalent to the diagram. We define $\mathcal{D}_0^\alpha$ to be the class of all unitary α-diagrams and $\mathcal{D}^\alpha$ to be the class of all α-diagrams.

DEFINITION 5.4
The partial function $\mathcal{ZS}: \mathcal{Z} \times \mathcal{D}_0^\alpha \rightarrow \mathcal{S}$ ($\mathcal{ZS}$ for 'zone sentence') is specified for unitary α-diagram d and zone z in $VZ(d)$ (recall, $VZ(d)$ is the Venn zone set of d, defined in Section 2.2) as follows.

1. If z is not shaded in d and $S(\{z\},d) = 0$ then
$$\mathcal{ZS}(z,d) = \top.$$
2. If z is not shaded in d and $S(\{z\},d) = n > 0$ then
$$\mathcal{ZS}(z,d) = \exists x_1 ... \exists x_n \Big(\bigwedge_{1 \le j < k \le n} \neg(x_j = x_k) \wedge \bigwedge_{1 \le k \le n} \mathcal{ZF}(z,x_k) \Big).$$
3. If z is either missing from d or is shaded in d and $S(\{z\},d) = 0$ then
$$\mathcal{ZS}(z,d) = \forall x_1 \neg \mathcal{ZF}(z,x_1).$$
4. If z is shaded in d and $S(\{z\},d) = n > 0$ then
$$\begin{aligned}\mathcal{ZS}(z,d) \quad = \quad & \exists x_1 ... \exists x_n \Big(\bigwedge_{1 \le j < k \le n} \neg(x_j = x_k) \wedge \bigwedge_{1 \le k \le n} \mathcal{ZF}(z,x_k) \wedge \\ & \Big(\forall x_{n+1} \big(\bigvee_{1 \le j \le n} x_{n+1} = x_j \vee \neg \mathcal{ZF}(z,x_{n+1}) \big) \Big) \Big).\end{aligned}$$

DEFINITION 5.5
Define $\mathcal{DS}: \mathcal{D}^\alpha \rightarrow \mathcal{S}$ ($\mathcal{DS}$ for 'diagram sentence') as follows.

1. If $d = \perp$ then $\mathcal{DS}(d) = \perp$.

2. If d $(\neq \bot)$ is a unitary α-diagram then

$$\mathcal{DS}(d) = \bigwedge_{z \in VZ(d)} \mathcal{ZS}(z, d).$$

3. If $D = \overline{D_1}$ then $\mathcal{DS}(D) = \neg \mathcal{DS}(D_1)$.
4. If $D = D_1 \sqcup D_2$ then $\mathcal{DS}(D) = (\mathcal{DS}(D_1) \vee \mathcal{DS}(D_2))$.
5. If $D = D_1 \sqcap D_2$ then $\mathcal{DS}(D) = (\mathcal{DS}(D_1) \wedge \mathcal{DS}(D_2))$.

We wish to show, for unitary α-diagram d, that $\mathcal{DS}(d)$ is expressively equivalent to d. To do this, we shall consider each zone of d in turn. Thus it is useful to consider when an interpretation satisfies a zone, which we now define.

DEFINITION 5.6
Let $p = (U, \Psi)$ be an interpretation and let d be a unitary α-diagram. Let $z \in VZ(d)$. Given d, we say p **satisfies** z, denoted $p \models_d z$, if and only if the following hold.

1. The number of elements in the set represented by z is at least the number of spiders in z:

$$|\Psi(z)| \geq S(\{z\}, d).$$

2. If z is shaded or missing then the number of elements in the set represented by z equals the number of spiders in z:

$$z \in Z^*(d) \cup MZ(d) \Rightarrow |\Psi(z)| = S(\{z\}, d).$$

LEMMA 5.7
Let $p = (U, \Psi)$ be an interpretation and let d $(\neq \bot)$ be a unitary α-diagram. The interpretation p satisfies d if and only if p satisfies all the Venn zones of d:

$$p \models d \Leftrightarrow \forall z \in VZ(d) \; p \models_d z.$$

PROOF. **(Sketch)** Noting that when d is an α-diagram, $S(r, d) = T(r, d)$ for each region r in d the result follows from a straightforward restatement of the semantics predicate. ■

THEOREM 5.8
Let d be a unitary α-diagram. Diagram d is expressively equivalent to $\mathcal{DS}(d)$.

PROOF. **(Sketch)** For each zone, $z \in VZ(d)$, in turn, show that

$$\{h(p) \in \mathcal{INT} : p \models_d z\} = \{m \in \mathcal{STR} : m \models \mathcal{ZS}(z)\}.$$

The result then follows by Lemma 5.7. ■

COROLLARY 5.9
Let D be an α-diagram. Then D is expressively equivalent to $\mathcal{DS}(D)$.

THEOREM 5.10
The language of spider diagrams is at most as expressive as the language $\mathcal{MFOL}_=$.

6 Mapping from sentences to diagrams

We now consider the more challenging task of constructing a diagram for a sentence. Since every formula is semantically equivalent to a sentence obtained by prefixing the formula with $\forall x_i$ for each free variable x_i (i.e. constructing its universal closure), we only need to identify a diagram expressively equivalent to each sentence.

In [16] Swoboda and Allwein give an algorithm that determines whether a given first-order logic sentence containing only monadic predicates can be *observed* from a given Euler/Venn diagram. Sentences observable from a diagram are logical consequences of the diagram (but the diagram and the sentence are not necessarily expressing the same information). They also give an algorithm to determine if a diagram is observable from a sentence. First they manipulate the sentence into a special normal form that they call Euler/Venn conjunctive normal form (EVCNF). Using this normal form it is then possible to construct a directed acyclic graph (DAG) for the sentence. A DAG is also constructed for the given diagram. Transformation rules are then applied to the DAG for the sentence (analogous to reasoning rules for their Euler/Venn system) to determine whether it can be changed into the DAG arising from the diagram. If it can then the diagram is observable from the sentence. The approach to determine if a sentence is observable from a diagram is similar.

Shin's approach to show Venn-II is equally as expressive as language L_0 ($\mathcal{MFOL}_=$ without equality) is algorithmic [15]. To find a diagram expressively equivalent to a sentence, she first converts the sentence into prenex normal form, say $Q_1x_1...Q_nx_nG$ where each Q_i is a quantifier and G is quantifier free. If Q_n is universal then G is transformed into conjunctive normal form. If Q_n is existential then G is transformed into disjunctive normal form. The quantifier Q_n is then distributed through G and as many formulae are removed from its scope as possible. All n quantifiers are distributed through the sentence in this way. The sentence resulting from this process has no nested quantifiers. A diagram can then be drawn for each of the simple parts of the resulting formula. To adapt this algorithm to find expressively equivalent diagrams for sentences in $\mathcal{MFOL}_=$ that do not involve equality is straightforward.

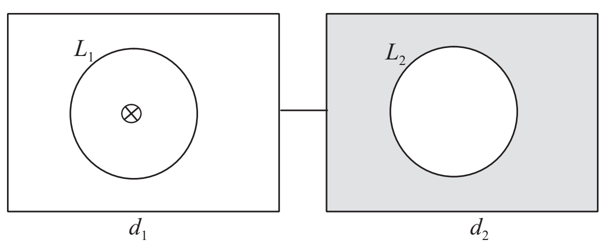
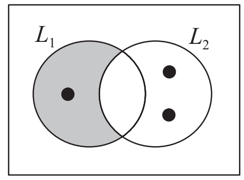


FIGURE 6. Illustrating Shin's algorithm.

EXAMPLE 6.1
Applying Shin's algorithm to the sentence $\exists x_1 \forall x_2 (P_1(x_1) \vee P_2(x_2))$ gives rise to the diagram shown in Figure 6 (recall that in Venn-II disjunction between diagrams is denoted by connecting them with a straight line segment).

Shin's algorithm does not readily generalize to arbitrary sentences in $\mathcal{MFOL}_=$ because $=$ is a dyadic predicate symbol which means nesting of quantifiers cannot necessarily be removed. We take a different approach, modelled on the classic result of Dreben and Goldforb [3, 209–210]. To establish the existence of a diagram expressively equivalent to a sentence we consider models for that sentence. To illustrate the approach we consider relationships between models for α-diagrams.



d

FIGURE 7. Extending models for a diagram.

EXAMPLE 6.2
The diagram in Figure 7 has a *minimal* model (in the sense that the cardinality of the universal set is minimal) $U = \{1, 2, 3\}$, $\Psi(L_1) = \{1\}$, $\Psi(L_2) = \{2, 3\}$ and, for $i \neq 1, 2$, $\Psi(L_i) = \emptyset$. This model can be used to characterize all the models for the diagram, up to isomorphism. We can use this model to generate further models, by adding elements to U and we may add these elements to images of contour labels if we so choose. As an example, the element 4 can be added to U and we redefine $\Psi(L_2) = \{2, 3, 4\}$ to give another model for d. No matter what changes we make to the model, we must ensure that the zone $(\{L_1\}, \{L_2\})$ always represents a set containing exactly one element or we will create an interpretation that does not satisfy the diagram. We can add elements to all and only the sets represented by zones which are not shaded. Adding elements in this way will generate all models for d, up to isomorphism.

In considering models for $\mathcal{MFOL}_=$ sentences we will use the notion of a *predicate intersection set*. This is the interpretation of the conjunction of certain monadic predicate symbols, and thus corresponds to the interpretation of a zone in a diagram. Suppose m is a model for sentence S. We will show that if a predicate intersection set satisfies certain cardinality conditions then we can increase the cardinality of that predicate intersection set (enlarging m) and still have a model for S. We are able to use this fact to show that there is a finite set of models for S that can be used to classify all the models for S. Moreover, we can use this classifying set to construct a diagram expressively equivalent to S.

DEFINITION 6.3
Let m be a structure and let X and Y be finite subsets of $\mathcal{P}$ (the countably infinite set of predicate symbols). Define the **predicate intersection set** in m with respect to X and Y, denoted $PI(m, X, Y)$, to be

$$PI(m, X, Y) = \bigcap_{P_i \in X} P_i^m \cap \bigcap_{P_i \in Y} \overline{P_i^m}$$

(recall that P_i^m is the interpretation of P_i in m). We define $\bigcap\limits_{P_i \in \emptyset} P_i^m = \bigcap\limits_{P_i \in \emptyset} \overline{P_i^m} = U$ where U is the domain of m.

In the context of $\mathcal{MFOL}_=$, we will identify all the structures that can be generated from a given structure, m, by adding or renaming elements subject to cardinality restrictions determined by sentence S. We will call this class of structures generated by m the *cone* of m, given S. For each sentence, S, we will show that there is a finite set of models, the union of whose cones is precisely the collection of models for S. Formalizing and proving this insight is the kernel of the result here. Central to our approach is the notion of *similar structures with respect to* S. To define similar structures we use the maximum number of *nested quantifiers* in S.[2]

[2] The maximum number of nested quantifiers in S is called the **quantifier rank** of S [4].

EXAMPLE 6.4
Let S be the sentence $\forall x_1\, P_1(x_1) \wedge \forall x_1\, \exists x_2\, \neg(x_1 = x_2)$. The formula $\forall x_1\, P_1(x_1)$ has one nested quantifier and $\forall x_1\, \exists x_2\, \neg(x_1 = x_2)$ has two nested quantifiers. Therefore the maximum number of nested quantifiers in S is two. Now, n nested quantifiers introduce n variable names, and so it is only possible to talk about (at most) n distinct individuals within the body of the formula. This has the effect of limiting the complexity of what can be said by such a formula. In the particular case here, this observation has the effect that if a model for S has at least two elements in certain predicate intersection sets then S does not place an upper bound on the cardinalities of those predicate intersection sets.

In a model for S, the interpretation of P_1 has to contain all the elements, of which there must be at least two. Also, S constrains the predicate intersection set $PI(m, \emptyset, \{P_1\})$ to have cardinality zero. As an example, we consider two models, m_1 and m_2 with domains $U_1 = \{1, 2, 3, 4\}$ and $U_2 = \{1, 2, 5, 6, 7\}$, respectively, that are characterized by $P_1^{m_1} = \{1, 2, 3, 4\}$ and $P_1^{m_2} = \{1, 2, 5, 6, 7\}$. Now

$$|PI(m_1, \emptyset, \{P_1\})| = |\emptyset| = 0 < 2 \quad \text{and} \quad |PI(m_2, \emptyset, \{P_1\})| = |\emptyset| = 0 < 2.$$

Also

$$|PI(m_1, \{P_1\}, \emptyset)| = |U_1| \geq 2 \quad \text{and} \quad |PI(m_2, \{P_1\}, \emptyset)| = |U_2| \geq 2,$$

so S cannot place an upper bound on $|PI(m, \{P_1\}, \emptyset)|$. We can think of m_1 and m_2 as each enlarging the model m_3 with domain $U_3 = \{1, 2\}$ where $P_1^{m_3} = \{1, 2\}$ and $P_j^{m_3} = \emptyset$, for all $j \neq 1$.

The following definition, Lemmas 6.6, 6.8 and Corollary 6.7 are adapted (by changing the notation and adding details to the proofs) from [3, 209–210].

DEFINITION 6.5
Let S be a sentence and define $q(S)$ to be the maximum number of nested quantifiers in S and $P(S)$ to be the set of monadic predicate symbols in S. Two structures m_1 and m_2 are called **similar with respect to** S if and only if for each subset X of $P(S)$, either

(1) $PI(m_1, X, P(S) - X) = PI(m_2, X, P(S) - X)$ or
(2) $|PI(m_1, X, P(S) - X) \cap PI(m_2, X, P(S) - X)| \geq q(S)$

and, in addition to (1) or (2), for all subsets Y of $P(S)$ such that $X \neq Y$,

$$PI(m_1, X, P(S) - X) \cap PI(m_2, Y, P(S) - Y) = \emptyset.$$

In the previous example, m_1, m_2 and m_3 are all similar with respect to S. There is a close relationship between the notions of similar structures and homomorphic structures, although they are not equivalent. Consider the structures m_4 and m_5 defined below:

$$m_4 = \langle \{1\}, \{(1, 1)\}, \{1\}, \emptyset, \emptyset, ...\rangle$$

and

$$m_5 = \langle \{2\}, \{(2, 2)\}, \{2\}, \emptyset, \emptyset, ...\rangle.$$

These structures are homomorphic (indeed, they are isomorphic) but they are not similar with respect to the sentence $\forall x_1\, (P_1(x_1) \vee P_2(x_1))$. For example,

$$PI(m_4, \{P_1\}, \{P_2\}) = \{1\} \neq PI(m_5, \{P_1\}, \{P_2\}) = \{2\},$$

so

$$|PI(m_4, \{P_1\}, \{P_2\}) \cap PI(m_5, \{P_1\}, \{P_2\})| = |\emptyset| \not\geq q(\forall x_1\, (P_1(x_1) \vee P_2(x_1))) = 1.$$

Therefore, when $X = \{P_1\}$, neither condition (1) nor condition (2) in the definition of similar structures hold for m_4 and m_5. We also observe that, given a sentence S, if we restrict the set of predicate symbols in our language $\mathcal{MFOL}_=$ to include only those in S (i.e $P(S)$), along with equality, then similar structures are also homomorphic.

LEMMA 6.6
Let S be a sentence. Let m_1 and m_2 be similar structures with respect to S and with domains U_1 and U_2 respectively. For all (not necessarily proper) subformulas G of S and for each assignment of values in $U_1 \cap U_2$ to the free variables (if any) of G, G is true in m_1 under the assignment if and only if G is true in m_2 under the assignment.

PROOF. The proof is by induction on the complexity of G (i.e. the depth of G in an inductive construction of formulae). If G is atomic, then G is $P_k(v)$ or $v = w$. In the case when $v = w$ the result is obvious. For $P_k(v)$, assign $x \in U_1 \cap U_2$ to v. Suppose $P_k(v)$ is true in m_1 under this assignment. We will show that $P_k(v)$ is true in m_2 under this assignment. Now, there exist X and Y, both subsets of $P(S)$, such that

$$x \in PI(m_1, X, P(S) - X) \cap PI(m_2, Y, P(S) - Y).$$

Moreover, since $P_k(v)$ is true in m_1 under this assignment, $P_k \in X$. Since m_1 and m_2 are similar with respect to S it follows that $X = Y$. Thus $P_k(v)$ is true in m_2 under this assignment. The converse is similar.

If G is $H_1 \vee H_2$, $H_1 \wedge H_2$ or $\neg H_1$, then the result follows immediately if it holds for H_1 and H_2 separately.

Let G be $\exists v H$, and suppose an assignment of values in $U_1 \cap U_2$ to the free variables of G is fixed. Let Y be the set of values so assigned. Since G is a subformula of S, it contains at most $q(S) - 1$ free variables. Hence $|Y| < q(S)$. Suppose G is true in m_1 under the assignment. Hence there is an a in U_1 such that H is true in m_1 when, additionally, the variable v is assigned the value a. If $a \in U_2$, then by the inductive hypothesis, H is true in m_2 under the augmented assignment.

Suppose therefore that a is not in U_2, and let a be in $PI(m_1, X, P(S) - X)$, where $X \subseteq P(S)$. Thus

$$PI(m_1, X, P(S) - X) \neq PI(m_2, X, P(S) - X),$$

so

$$PI(m_1, X, P(S) - X) \cap PI(m_2, X, P(S) - X)$$

has cardinality at least $q(S)$. But then there is an element b of $(PI(m_1, X, P(S) - X) \cap PI(m_2, X, P(S) - X)) - Y$. Let $\gamma : U_1 \to U_1$ carry a to b, b to a and every other member of U_1 to itself. Then γ is an automorphism of the structure m_1, because the sets $PI(m_1, X, P(S) - X)$ completely characterize the model m_1 by partitioning the elements according to which of the monadic predicates that they satisfy and interchanging two elements within the same partition therefore changes none of the logical properties of the structure, and γ is the identity on Y. Hence H is true in m_1 under the original assignment augmented by assigning b to v. Then, by the inductive hypothesis, H is true in m_2 under this augmented assignment, so $\exists v H$ is true in m_2 under the original assignment. We have shown that if G is true in m_1 then G is true in m_2. The converse is similar.

The case $G = \forall v H$ remains. Since G is logically equivalent to $\neg \exists v \neg H$ the preceding arguments suffice. ∎

COROLLARY 6.7
If m_1 and m_2 are similar structures with respect to S, then m_1 is a model for S if and only if m_2 is a model for S.

LEMMA 6.8
Let S be a sentence. If S has a model of any cardinality at least $2^{|P(S)|}q(S)$ then S has models of every cardinality at least $2^{|P(S)|}q(S)$.

PROOF. Suppose S has a model m_1 with universe U_1 of cardinality at least $2^{|P(S)|}q(S)$. Then $|PI(m_1, X, P(S) - X)| \geq q(S)$ for at least one $X \subseteq P(S)$. So, for each $j \geq 2^{|P(S)|}q(S)$ there is a structure m_2 similar to m_1 whose universe has cardinality j. Hence there are models for S with every cardinality at least $2^{|P(S)|}q(S)$. ∎

The (upward) Löwenheim–Skolem theorem tells us that if a sentence of first-order logic has a model of a particular *infinite* cardinality, then it has models of all larger cardinalities; it is not the case that this holds for finite models. A simple counterexample is the sentence which states that P is an equivalence relation all of whose equivalence classes are of size two; the finite models of this will necessarily have even cardinality.

DEFINITION 6.9
Let S be a sentence and suppose m is a model for S. If the cardinality of m is at most $2^{|P(S)|}q(s)$ then we say m is a **small model** for S. Otherwise we say m is a **large model** for S.

DEFINITION 6.10
Let S be a sentence and suppose m_1 is a small model for S. An ***S*-extension** of m_1 is a structure, m_2, for $\mathcal{MFOL}_=$ such that for each subset, X, of $P(S)$

$$PI(m_1, X, P(S) - X) \subseteq PI(m_2, X, P(S) - X)$$

and, if $|PI(m_1, X, P(S) - X)| < q(S)$ then

$$PI(m_1, X, P(S) - X) = PI(m_2, X, P(S) - X).$$

The **cone** of m_1 given S, denoted $cone(m_1, S)$, is a class of structures such that $m_2 \in cone(m_1, S)$ if and only if m_2 is isomorphic to some S-extension of m_1.

The cone of m given S contains models for S that can be restricted to (models isomorphic to) m. We can think of elements of $cone(m, S)$ as extending m in certain 'directions' and fixing m in others.

EXAMPLE 6.11
Let S be the sentence $\exists x_1 \exists x_2\, P_1(x_1) \vee P_2(x_2)$ which has $q(S) = 2$. So, if we have predicate intersection sets containing two or more elements we can add arbitrarily many elements to them and preserve the fact that S holds. Consider

$$m = \langle \{1, 2, 3, 4\}, =^m, \{1, 2\}, \emptyset, \emptyset, ...\rangle.$$

A visual analogy of $cone(m, S)$ can be seen in Figure 8. The structure

$$m_1 = \langle \{1, 2, 3, 4, 5, 6\}, =^{m_1}, \{1, 2, 5\}, \emptyset, \emptyset, ...\rangle$$

can be obtained from m, extending $PI(m, \emptyset, \{P_1, P_2\})$ and $PI(m, \{P_1\}, \{P_2\})$ by adding elements to these sets (and the domain), but keeping $PI(m, \{P_2\}, \{P_1\})$ and $PI(m, \{P_1, P_2\}, \emptyset)$ fixed.

EXAMPLE 6.12
Let S be the sentence $\forall x_1 \forall x_2\, x_1 = x_2$ and consider the structure $m_1 = \langle \{1\}, =^{m_1}, \emptyset, \emptyset, \emptyset, ...\rangle$ which satisfies S. We have the following cone for m_1:

$$cone(m_1, S) = \{m_2 \in \mathcal{STR} : |PI(m_1, \emptyset, \emptyset)| = |\{1\}| = |PI(m_2, \emptyset, \emptyset)|\}.$$

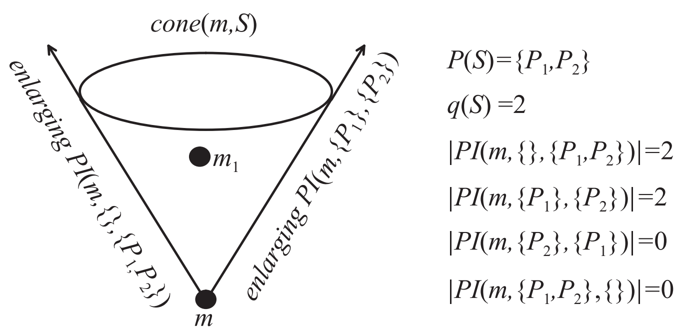


FIGURE 8. Visualizing cones.

The class $cone(m_1, S)$ contains only structures that are models for S but does not contain them all, for example $m_3 = \langle \emptyset, \emptyset, ...\rangle$ satisfies S but m_3 is not in $cone(m_1, S)$. All models for S are in the class $cone(m_1, S) \cup cone(m_3, S)$. In this sense, m_1 and m_3 classify all the models for S. We can draw a diagram expressively equivalent to S using information given by m_1 and m_3. This diagram is a disjunction of two unitary diagrams, shown in Figure 9.

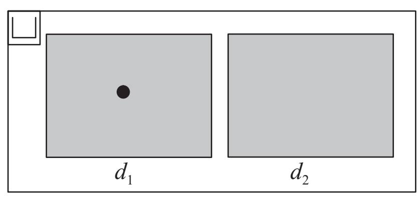


FIGURE 9. A diagram expressively equivalent to $\forall x_1 \forall x_2\, x_1 = x_2$.

LEMMA 6.13
Let S be a sentence and suppose m_1 is a large model for S. Then there exists a small model, m_2, for S such that $m_1 \in cone(m_2, S)$.

PROOF. Define m_2 as follows. Let X be a subset of $P(S)$. If $|PI(m_1, X, P(S)-X)| < q(S)$ define $M_X = PI(m_1, X, P(S)-X)$. Otherwise define M_X to be some chosen subset of $PI(m_1, X, P(S)-X)$ with cardinality $q(S)$. The domain of m_2 is

$$U_2 = \bigcup_{X \subseteq P(S)} M_X.$$

The set U_2 has cardinality at most $2^{|P(S)|}q(S)$. Define, for each $P_i \in \mathcal{P}$, $P_i^{m_2} = P_i^{m_1} \cap U_2$. We will show that structure m_2 is similar to m_1 and we will refer to the domain of m_1 by U_1. Let X be a subset of $P(S)$. Now

$$\begin{aligned} PI(m_2, X, P(S)-X) &= \bigcap_{P_i \in X} P_i^{m_2} \cap \bigcap_{P_i \in P(S)-X} \overline{P_i^{m_2}} \\ &= \bigcap_{P_i \in X} (P_i^{m_1} \cap U_2) \cap \bigcap_{P_i \in P(S)-X} \overline{(P_i^{m_1} \cap U_2)} \end{aligned}$$

$$\begin{aligned}
&= U_2 \cap \bigcap_{P_i \in X} P_i^{m_1} \cap (U_2 - \bigcup_{P_i \in P(S)-X} P_i^{m_1}) \\
&= U_2 \cap \bigcap_{P_i \in X} P_i^{m_1} \cap (U_1 - \bigcup_{P_i \in P(S)-X} P_i^{m_1}) \quad \text{since } U_2 \subseteq U_1 \\
&= U_2 \cap PI(m_1, X, P(S) - X).
\end{aligned}$$

It follows that $PI(m_2, X, P(S) - X) \subseteq PI(m_1, X, P(S) - X)$.

Suppose that $|PI(m_1, X, P(S))| \geq q(S)$. Then there is a subset of $PI(m_1, X, P(S) - X)$ with cardinality $q(S)$ that is also a subset of U_2, namely M_X. In which case $|PI(m_2, X, P(S) - X)| = q(s)$ and $|PI(m_1, X, P(S) - X) \cap PI(m_2, X, P(S) - X)| \geq q(S)$.

Alternatively, $|PI(m_1, X, P(S) - X)| < q(S)$. In which case $PI(m_1, X, P(S) - X) \subseteq U_2$. Hence

$$PI(m_1, X, P(S) - X) = PI(m_2, X, P(S) - X).$$

Let Y be a subset of $P(S)$ that is distinct from X. Now

$$PI(m_1, X, P(S) - X) \cap PI(m_1, Y, P(S) - Y) = \emptyset$$

and

$$PI(m_2, Y, P(S) - Y) \subseteq PI(m_1, Y, P(S) - Y).$$

Therefore

$$PI(m_1, X, P(S) - X) \cap PI(m_2, Y, P(S) - Y) = \emptyset.$$

Hence m_1 and m_2 are similar with respect to S. By Corollary 6.7, m_2 is a model for S, so m_2 is a small model for S.

We now show that m_1 is in the class $cone(m_2, S)$. For each subset X of $P(S)$, we have

$$PI(m_2, X, P(S) - X) \subseteq PI(m_1, X, P(S) - X).$$

If

$$|PI(m_2, X, P(S) - X)| < q(S)$$

then

$$PI(m_2, X, P(S) - X) = PI(m_1, X, P(S) - X)$$

and it follows that m_1 is an S-extension of m_2. Hence m_1 is in the class $cone(m_2, S)$. Thus for each large model, m_1, for S there exists a small model, m_2, for S such that $m_1 \in cone(m_2, S)$. ∎

LEMMA 6.14
Let m_1 be a small model for sentence S. Then $cone(m_1, S)$ only contains models for S.

PROOF. It is sufficient to prove that any S-extension of m_1 is a model for S, since it is clear that isomorphism preserves the sentences modelled by structures. Let m_2 be an S-extension of m_1. We will show that m_2 is similar to m_1, with respect to S. Since m_2 is an S-extension of m_1, it is the case that, for each subset X of $P(S)$,

$$PI(m_1, X, P(S) - X) \subseteq PI(m_2, X, P(S) - X)$$

and, when $|PI(m_1, X, P(S) - X)| < q(S)$,

$$PI(m_1, X, P(S) - X) = PI(m_2, X, P(S) - X).$$

Let $Y \subseteq P(S))$ such that $Y \neq X$. Now

$$PI(m_2, X, P(S) - X) \cap PI(m_2, Y, P(S) - Y) = \emptyset.$$

Furthermore

$$PI(m_1, X, P(S) - X) \subseteq PI(m_2, X, P(S) - X),$$

thus

$$PI(m_1, X, P(S) - X) \cap PI(m_2, Y, P(S) - Y) = \emptyset.$$

Therefore m_2 is similar to m_1, with respect to S. By Corollary 6.7, m_2 is a model for S. ∎

We will show that, given a sentence, S, there is a finite set of small models, the union of whose cones gives rise to only and all the models for S. We are able to use these models to identify a diagram expressively equivalent to S. In order to identify such a finite set we require the notion of *partial isomorphism* between structures.

DEFINITION 6.15
Let m_1 and m_2 be structures for $\mathcal{MFOL}_=$ with domains U_1 and U_2 respectively. Let Q be a set of monadic predicate symbols. If there exists a bijection $\gamma : U_1 \to U_2$ such that

$$\forall P_i \in Q \, \forall x \in U_1 \, (x \in P_i^{m_1} \Leftrightarrow \gamma(x) \in P_i^{m_2}),$$

then m_1 and m_2 are **isomorphic restricted to *Q*** and γ is a **partial isomorphism**.

LEMMA 6.16
Let S be a sentence and let m_1 and m_2 be structures. If m_1 and m_2 are isomorphic restricted to $P(S)$ then m_1 is a model for S if and only if m_2 is a model for S.

LEMMA 6.17
There are finitely many small models for sentence S, up to isomorphism restricted to $P(S)$.

PROOF. **(Sketch)** There is a finite choice for the size of each of the predicate intersection sets (because they are small) and a finite number of these, given $P(S)$. ∎

LEMMA 6.18
Let S be a sentence and let m_1 and m_2 be structures isomorphic restricted to $P(S)$. If m_1 and m_2 are small models for S then $cone(m_1, S) = cone(m_2, S)$.

PROOF. Since m_1 and m_2 are isomorphic restricted to $P(S)$, for each subset X of $P(S)$ it is the case that

$$|PI(m_1, X, P(S) - X)| = |PI(m_2, X, P(S) - X)|.$$

For each S-extension of m_1 there is an S-extension of m_2 to which m_1 is isomorphic, shown by extending γ in the obvious way. Similarly any S-extension of m_2 is isomorphic to an S-extension of m_1. It follows that $cone(m_1, S) = cone(m_2, S)$. ∎

DEFINITION 6.19
Let S be a sentence. A set of small models, $c(S)$, for S is called a **classifying set of models** for S if for each small model, m_1, for S there is a unique m_2 in $c(S)$ such that m_1 and m_2 are isomorphic, restricted to $P(S)$.

LEMMA 6.20
Let S be a sentence. Then there exists a set of classifying models for S and all such sets are finite.

PROOF. Choose one small model from each equivalence class of small models under the relation of partial isomorphism restricted to $P(S)$ to give $c(S)$. Finiteness follows from Lemma 6.17. ∎

We will now show that the union of the cones of the models in $c(S)$ is precisely the collection of models for S.

THEOREM 6.21
Let S be a sentence and $c(S)$ be a classifying set of models for S. Then $\bigcup_{m \in c(S)} cone(m, S)$ is precisely the collection of models for S.

PROOF. By Lemma 6.14, $\bigcup_{m \in c(S)} cone(m, S)$ only contains models for S.

We must now show that all the models for S are in $\bigcup_{m \in c(S)} cone(m, S)$. The first step is to show that any small model, m_1, for S is in $\bigcup_{m \in c(S)} cone(m, S)$. If $m_1 \in c(S)$ then it is trivial that $m_1 \in \bigcup_{m \in c(S)} cone(m, S)$. If $m_1 \notin c(S)$ then there is some small model $m_2 \in c(S)$ that is isomorphic, restricted to $P(S)$, to m_1. By Lemma 6.18, $cone(m_1, S) = cone(m_2, S)$. It follows that $m_1 \in \bigcup_{m \in c(S)} cone(m, S)$. Finally we must show that each large model, m_3, for S is in $\bigcup_{m \in c(S)} cone(m, S)$. By Lemma 6.13, there is a small model, m_4, such that $m_3 \in cone(m_4, S)$. If $m_4 \in c(S)$ then we are done. If $m_4 \notin c(S)$ then there is an $m_5 \in c(S)$ such that m_4 is isomorphic restricted to $P(S)$ to m_5. Therefore $m_3 \in cone(m_5, S)$. Thus all the models for S are in $\bigcup_{m \in c(S)} cone(m, S)$. Hence $\bigcup_{m \in c(S)} cone(m, S)$ is precisely the collection of models for S. ∎

To summarize, we have shown that every sentence, S, has a finite set of classifying models and the union of the cones of these classifying models is precisely the collection of models for S. We will now use these classifying models to construct a diagram expressively equivalent to S.

DEFINITION 6.22
Let m be a small model for a sentence S. The unitary α-diagram, d, **representing** m given S, denoted $\mathcal{REP}(m, S) = d$, is defined as follows.[3]

1. The contour labels arise from the predicate symbols in $P(S)$:

$$L(d) = \{L_i \in \mathcal{L} : \exists P_i \in \mathcal{P}\, P_i \in P(S)\}.$$

2. The diagram is in Venn form:

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

   That is, d contains all possible zones.

3. The shaded zones in d are given as follows. Let X be a subset of $P(S)$ such that $|PI(m, X, P(S) - X)| < q(S)$. The zone $(a, L(d) - a)$ in $Z(d)$ where $a = \{L_i \in L(d) : P_i \in X\}$ is shaded.

[3]In fact, d is a β-diagram (every zone is shaded or inhabited by at least one existential spider) [13] except when $S = \top$.

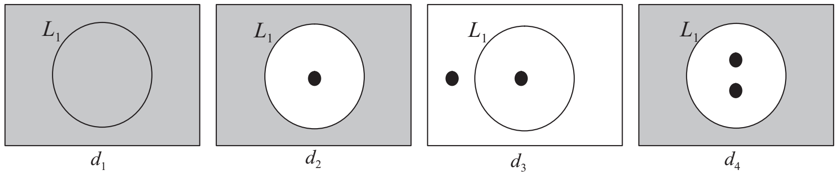


FIGURE 10. Constructing diagrams from models.

4. The number of spiders in each zone is the cardinality of the set $|PI(m, X, P(S) - X)|$ where X gives rise to the containing set of contour labels for that zone. More formally, the set of spider identifiers is:

$$SI(d) = \{(n, r) : \exists X\ X \subseteq P(S) \wedge |PI(m, X, P(S) - X)| > 0 \wedge$$

$$n = |PI(m, X, P(S) - X)| \wedge r = \{(a, L(d) - a) \in Z(d) : a = \{L_i \in L(d) : P_i \in X\}\}\}.$$

Let $c(S)$ be a set of classifying models for S. Define $\mathcal{SD}(S)$ to be a disjunction of unitary diagrams, given by

$$\mathcal{SD}(S) = \bigsqcup_{m \in c(S)} \mathcal{REP}(m, S),$$

unless $c(S) = \emptyset$, in which case $\mathcal{SD}(S) = \perp$.

EXAMPLE 6.23
Let S be the sentence $\exists x_1 P_1(x_1) \vee \forall x_1 P_1(x_1)$. To find a classifying set of models we must consider structures of all cardinalities up to $2^{|\{P_1\}|} \times q(S) = 2^1 \times 1 = 2$. There are six distinct structures (up to isomorphism restricted to $P(S)$) with cardinality at most 2. Four of these structures are models for S and are listed below.

1. $m_1 = \langle \emptyset, \emptyset, ... \rangle$,
2. $m_2 = \langle \{1\}, =^{m_2}, \{1\}, \emptyset, \emptyset, ... \rangle$,
3. $m_3 = \langle \{1, 2\}, =^{m_3}, \{1\}, \emptyset, \emptyset, ... \rangle$,
4. $m_4 = \langle \{1, 2\}, =^{m_4}, \{1, 2\}, \emptyset, \emptyset, ... \rangle$.

Therefore, the class $cone(m_1, S) \cup cone(m_2, S) \cup cone(m_3, S) \cup cone(m_4, S)$ contains only and all the models for S. We use each of these models to construct a diagram. The models m_1, m_2, m_3 and m_4 give rise to the diagrams d_1, d_2, d_3 and d_4 respectively in Figure 10. The diagram $d_1 \sqcup d_2 \sqcup d_3 \sqcup d_4$ is expressively equivalent to S. This is not the 'natural' diagram one would associate with S. We note here that m_4 is an S-extension of m_2, so $cone(m_2, S) \subseteq cone(m_4, S)$. The sentence S is, therefore, expressively equivalent to $d_1 \sqcup d_2 \sqcup d_3$. In general, when constructing a diagram expressively equivalent to S we only need to draw a diagram for each model in $c(S)$ that is not (isomorphic to) an S-extension of some other model in $c(S)$.

In fact, we can make further refinements to our approach. We note that $d_2 \sqcup d_3$ is semantically equivalent to d_5 in figure 11. By capturing this kind of property at the model level, which may involve defining an algebra of structures, we could further reduce the number of models required to define $\mathcal{SD}(S)$. We would, though, need to mark each predicate intersection set with whether it could be extended indefinitely.

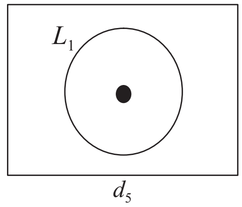


FIGURE 11. Refining the approach.

THEOREM 6.24
Let S be a sentence. Then S is expressively equivalent to $\mathcal{SD}(S)$.

PROOF. Let $c(S)$ be a set of classifying models for S. For each $m_1 \in c(S)$, we will show that the models for the diagram $\mathcal{REP}(m_1, S)$ are in bijective correspondence (under h defined in Lemma 4.2) with the structures in $cone(m_1, S)$. To do so, we show first that any model for $d = \mathcal{REP}(m_1, S)$ is in $cone(m_1, S)$. Second we will show that the inverse, under h, of any element in $cone(m_1, S)$ is a model for d.

Let (U, Ψ) be a model for d. We will now show $h(U, \Psi) \in cone(m_1, S)$. To do so, we will show that $h(U, \Psi)$ is an S-extension of some small model, m_2, for S and that m_2 is isomorphic, restricted to $P(S)$, to m_1.

We define m_2 as follows. Let X be a subset of $P(S)$. Choose $z = (a, b) \in Z(d)$ such that $a = \{L_i \in L(d) : P_i \in X\}$. Then, since $(U, \Psi) \models_d z$,

$$|\Psi(z)| \geq S(\{z\}, d).$$

Now

$$\begin{aligned}
|\Psi(z)| &= |\bigcap_{L_i \in a} \Psi(L_i) \cap \bigcap_{L_i \in b} \overline{\Psi(L_i)}| \\
&= |\bigcap_{P_i \in X} P_i^{h(U,\Psi)} \cap \bigcap_{P_i \in P(S) - X} \overline{P_i^{h(U,\Psi)}}| \\
&= |PI(h(U, \Psi), X, P(S) - X)| \\
&\geq S(\{z\}, d) \\
&= |PI(m_1, X, P(S) - X)|.
\end{aligned}$$

Therefore there exists an injection,

$$f_X : PI(m_1, X, P(S) - X) \to PI(h(U, \Psi), X, P(S) - X).$$

Choose such an injection, f_X. We define the domain of m_2 to be U_2 where

$$U_2 = \bigcup_{X \subseteq P(S)} im(f_X).$$

We note that $U_2 \subseteq U$ and, since m_1 is a small model for S, $|U_2| \leq 2^{|P(S)|} q(S)$. Moreover, $|U_2| = |U_1|$ (where U_1 is the domain of m_1). Next we define, for each $P_i \in \mathcal{P}$,

$$P_i^{m_2} = P_i^{h(U,\Psi)} \cap U_2.$$

We define a bijection, $\gamma\colon U_1 \to U_2$, by $\gamma = \bigcup_{X \subseteq P(S)} f_X$. It is straightforward to verify that γ is a partial isomorphism. It follows that $cone(m_2, S) = cone(m_1, S)$, by Lemma 6.18.

We now show that $h(U, \Psi)$ is an S-extension of m_2. Let X be a subset of $P(S)$. Now

$$\begin{aligned}
PI(m_2, X, P(S) - X) &= \bigcap_{P_i \in X} P_i^{m_2} \cap \bigcap_{P_i \in P(S) - X} \overline{P_i^{m_2}} \\
&= \bigcap_{P_i \in X} (P_i^{h(U,\Psi)} \cap U_2) \cap \bigcap_{P_i \in P(S) - X} \overline{(P_i^{h(U,\Psi)} \cap U_2)} \\
&= U_2 \cap \bigcap_{P_i \in X} P_i^{h(U,\Psi)} \cap (U_2 - \bigcup_{P_i \in P(S) - X} \overline{P_i^{h(U,\Psi)}} \\
&= U_2 \cap \bigcap_{P_i \in X} P_i^{h(U,\Psi)} \cap (U - \bigcup_{P_i \in P(S) - X} \overline{P_i^{h(U,\Psi)}} \quad \text{since } U_2 \subseteq U \\
&= U_2 \cap PI(h(U, \Psi), X, P(S) - X) \qquad (1)
\end{aligned}$$

It follows that $PI(m_2, X, P(S) - X) \subseteq PI(h(U, \Psi), X, P(S) - X)$.

In order to show that $h(U, \Psi)$ is an S-extension of m_2, all that remains is to show that when $|PI(m_2, X, P(S) - X)| < q(S)$ we have

$$PI(m_2, X, P(S) - X) = PI(h(U, \Psi), X, P(S) - X).$$

Suppose $|PI(m_2, X, P(S) - X| < q(S)$. In which case $|PI(m_1, X, P(S) - X)| < q(S)$, since

$$|PI(m_1, X, P(S) - X)| = |PI(m_2, X, P(S) - X)|$$

(which follows from the fact that m_1 and m_2 are isomorphic restricted to $P(S)$). By the definition of d, the zone $z = (a, b) \in Z(d)$ where $a = \{L_i \in L(d) : P_i \in X\}$ is shaded. Since $(U, \Psi) \models_d z$, $|\Psi(z)| = S(\{z\}, d)$. Therefore

$$|\Psi(z)| = |PI(m_1, X, P(S) - X)|$$

and it follows that f_X is bijective. Thus $PI(h(U, \Psi), X, P(S) - X) = im(f_X)$. Therefore $PI(h(U, \Psi), X, P(S) - X) \subseteq U_2$ and we deduce from (1)

$$PI(m_2, X, P(S) - X) = PI(h(U, \Psi), X, P(S) - X).$$

Hence $h(U, \Psi)$ is an S-extension of m_2. Therefore $h(U, \Psi) \in cone(m_2, S)$. Therefore, by Lemma 6.14, $h(U, \Psi) \in cone(m_2, S) = cone(m_1, S)$. Hence

$$\{h(U, \Psi) : (U, \Psi) \in \mathcal{INT} \wedge (U, \Psi) \models \mathcal{REP}(m_1, S))\} \subseteq cone(m_1, S).$$

We must now show that

$$\{h(U, \Psi) : (U, \Psi) \in \mathcal{INT} \wedge (U, \Psi) \models \mathcal{REP}(m_1, S))\} \supseteq cone(m_1, S).$$

Let $m_2 \in cone(m_1, S)$ and let $z = (a, b) \in Z(d)$. We show $h^{-1}(m_2) = (U_2, \Psi) \models_d z$. Define X to be the subset of $P(S)$ that satisfies $a = \{L_i \in L(d) : P_i \in X\}$. Since $m_2 \in cone(m_1, S)$, the structure m_2 is isomorphic to some S-extension, m_3 say, of m_1. Now $PI(m_1, X, P(S) - X) \subseteq PI(m_3, X, P(S) - X)$, therefore there exists an injective map

$$f_X : PI(m_1, X, P(S) - X) \to PI(m_2, X, P(S) - X).$$

So

$$\begin{aligned} |\Psi(z)| &= |PI(m_2, X, P(S) - X)| \\ &\geq |PI(m_1, X, P(S) - X)| \\ &= S(\{z\}, d). \end{aligned}$$

Suppose that z is shaded in d. Then $|PI(m_1, X, P(S) - X)| < q(S)$ and

$$PI(m_1, X, P(S) - X) = PI(m_3, X, P(S) - X).$$

In which case there is a bijection

$$f_X : PI(m_1, X, P(S) - X) \to PI(m_3, X, P(S) - X).$$

Therefore $|\Psi(z)| = S(\{z\}, d)$. It follows that $h^{-1}(m_2) \models_d z$. Since z was an arbitrary zone we deduce, by lemma 5.7, $h^{-1}(m_2) \models d$. Therefore

$$\{h(U, \Psi) : (U, \Psi) \in \mathcal{INT} \wedge (U, \Psi) \models \mathcal{REP}(m_1, S)\} \supseteq cone(m_1, S).$$

Hence

$$\{h(U, \Psi) : (U, \Psi) \in \mathcal{INT} \wedge (U, \Psi) \models \mathcal{REP}(m_1, S)\} = cone(m_1, PS).$$

It follows that $\mathcal{SD}(S)$ is expressively equivalent to S. ∎

THEOREM 6.25
The language of spider diagrams and $\mathcal{MFOL}_=$ are equally expressive.

7 Conclusion

In this paper we have identified a fragment of first-order predicate logic equivalent in expressive power to the spider diagram language. To show that the spider diagram language is at most as expressive as $\mathcal{MFOL}_=$, we identified a sentence in $\mathcal{MFOL}_=$ that expressed the same information as a given diagram. To show that $\mathcal{MFOL}_=$ is at most as expressive as the language of spider diagrams we considered relationships between models for sentences. We have shown that it is possible to classify all the models for a sentence by a finite set of models. We then used these classifying models to define a spider diagram expressively equivalent to S. An interesting area, yet to be explored, is how the reasoning rules for first-order logic compare with the reasoning rules for spider diagrams.

8 Acknowledgements

Gem Stapleton thanks the UK EPSRC for support under grant number 01800274. John Howse, John Taylor and Simon Thompson are partially supported by the UK EPSRC grant numbers GR/R63516 and GR/R63509 for the Reasoning with Diagrams project. Thanks also to Andrew Fish, Jean Flower and Chris John for their comments on earlier drafts of this paper.

References

[1] S. N. Burris. *Logic for Mathematics and Computer Science*. Prentice Hall, 1998.

[2] R. Dechter and J. Pearl. Generalised best first search strategies and the optimality of a*, *Journal of the Association for Computing Machinery*, **32**, 505–536, 1985.

[3] B. Dreben and D. Goldforb. *The Decision Problem. Solvable Classes of Quantificational Formulas*. Addison Wesley, 1979.

[4] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1991.

[5] L. Euler. Lettres a une princesse d'allemagne. Sur divers sujets de physique et de philosophie, 1761. Letters No. 102-108 vol 2 Basel, Birkhauser.

[6] J. Flower, J. Masthoff, and G. Stapleton. Generating readable proofs: A heuristic approach to theorem proving with spider diagrams. In *Proceedings of International Conference on the Theory and Application of Diagrams*, pp. 166–181, Cambridge, UK, March 2004. Springer-Verlag.

[7] J. Flower and G. Stapleton. Automated theorem proving with spider diagrams. In *Proceedings of CATS'04, Computing: The Australasian Theory Symposium*, Electronic Note in Theoretical Computer Science, pp. 116–132, Dunedin, New Zealand, January 2004. Science Direct.

[8] W. Hodges. *Logic*. Penguin Books, 1977.

[9] J. Howse, F. Molina, S-J. Shin, and J. Taylor. On diagram tokens and types. In *Proceedings of International Conference on the Theory and Application of Diagrams*, Callaway Gardens, Georgia, USA, April 2002. *Lecture Notes in Artificial Intelligence*, pp. 76–90, Springer-Verlag.

[10] J. Howse, F. Molina, J. Taylor, S. Kent, and J. Gil. Spider diagrams: A diagrammatic reasoning system. *Journal of Visual Languages and Computing*, **12**, 299–324, June 2001.

[11] J. Howse, G. Stapleton, and J. Taylor. Reasoning with spider diagrams. Technical report, University of Brighton, 2004. Available from `www.cmis.brighton.ac.uk/research/vmg/SDRules.html`.

[12] P. T. Johnstone. *Notes on Logic and Set Theory*. Cambridge University Press, 1987.

[13] F. Molina. *Reasoning with Extended Venn-Peirce Diagrammatic Systems*. PhD thesis, University of Brighton, 2001.

[14] C. Peirce. Collected papers, vol 4. Harvard University Press, 1933.

[15] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.

[16] N. Swoboda and G. Allwein. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. In *International Workshop on Graph Transformation and Visual Modeling Techniques*. Elsevier Science, Barcelona, October 2002.

[17] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 1880.

[18] The visual modelling group web site, `www.cmis.brighton.ac.uk/research/vmg`.

Received 15 March 2004