A Constraint Diagram Reasoning System

Gem Stapleton, John Howse and John Taylor
The Visual Modelling Group
University of Brighton, Brighton, UK
www.cmis.brighton.ac.uk/research/vmg
Email: {g.e.stapleton, john.howse, john.taylor}@brighton.ac.uk

Abstract— **The Unified Modeling Language (UML) is a collection of notations which are mainly diagrammatic. These notations are used by software engineers in the process of object oriented modelling. The only textual notation in the UML is the Object Constraint Language (OCL). The OCL is used to express logical constraints such as system invariants. Constraint diagrams are designed to provide a diagrammatic alternative to the OCL. Since constraint diagrams are visual they complement existing notations in the UML. Spider diagrams form the basis of constraint diagrams and sound and complete reasoning systems have been developed. Spider diagrams allow subset relations between sets and cardinality constraints on sets to be expressed. In addition to this, constraint diagrams allow universal quantification and relational navigation and hence are vastly more expressive. In this paper we present the first constraint diagram reasoning system. We give syntax and semantics for constraint diagrams we call CD1 diagrams. We identify syntactic criteria that allow us to determine whether a CD1 diagram is satisfiable. We give descriptions of a set of sound and complete reasoning rules for CD1 diagrams.**

I. INTRODUCTION

Constraint diagrams were introduced in [10] as a notation for expressing constraints in object-oriented models. The notation integrates well with existing UML notations since all of the UML notations are diagrammatic, with the exception of the OCL [12] which is, essentially, a stylized form of first order predicate logic and is used to convey formal statements. Thus constraint diagrams provide a diagrammatic alternative to, and may be more intuitive than, the OCL. The diagram in Fig. 1 is a constraint diagram. It expresses the following. No mice are cats or dogs. No dogs are cats. Each cat is bigger than each mouse. There is a mouse that has been eaten by a cat. There is exactly one dog.

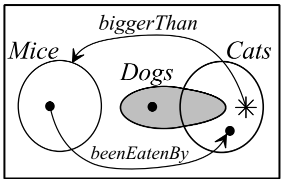

Fig. 1. A constraint diagram.

Constraint diagrams are based on Euler diagrams [1], introduced by Leonard Euler to illustrate subset relations between sets. Euler diagrams exploit topological properties of enclosure, exclusion and intersection to represent subset, disjoint sets and set intersection respectively. Venn diagrams [15] are similar to Euler diagrams. In Venn diagrams all possible intersections between contours must occur and shading is used to represent the empty set. Peirce [13] extended Venn diagrams by introducing ‘x-sequences’ to represent non-empty sets and ‘o-sequences’ to represent empty sets. Reasoning rules have been developed for Venn-Peirce diagrams [14] and Euler diagrams [6].

Spider diagrams [3], [7], [9], [11] modify and extend Venn-Peirce diagrams. Instead of using x-sequences to represent non-empty sets, *spiders* are used to represent the existence of elements and shading is used to place upper bounds on the cardinalities of sets. The motivation for the work on spider diagrams was to provide a basis for developing the much more expressive constraint diagram systems. Spider diagrams cannot express the complex constraints required when modelling software systems. Constraint diagrams can express statements involving two place predicates, whereas spider diagrams can only express statements involving one place predicates, along with equality.

For constraint diagrams to be considered as a formal language they must have formal underpinning. This underpinning is essential for the development of software tools to support the modelling process. Furthermore software developers who choose to use constraint diagrams for system specification may also require the ability to reason with these diagrams. To this end we give the syntax and semantics of a system of constraint diagrams, that we call CD1 diagrams, in section II. In section III we give syntactic criteria for identifying the satisfiability of constraint diagrams. A set of reasoning rules for CD1 diagrams are given in sections IV, V and VI. Many of these reasoning rules relate to arrows. We expect that this system will form the basis of future constraint diagram reasoning systems. The CD1 system we introduce here, while not as expressive as the full constraint diagram notation (which includes further syntactic elements), is considerably more expressive than previous spider diagram systems. We are using the CD1 system for pragmatic reasons and this work represents a significant step towards a reasoning system based on the full notation.

II. DESCRIPTION OF CD1 DIAGRAMS

A. The Syntax of Unitary CD1 Diagrams.

We now give an informal description of unitary CD1 diagrams. More details can be found in [4]. A *contour* is a labelled, simple closed plane curve. A *boundary rectangle* is a simple closed plane curve and is not labelled. A *basic region* is the bounded area of the plane enclosed by a contour or the boundary rectangle. A *region* is defined recursively: any basic region is a region and any non-empty union, intersection or difference of regions is a region. A *zone* is a region having no other region contained within it. An *existential spider* is a tree with nodes, called *feet*, placed in different zones. A *universal spider* is a star placed in a zone. No two universal spiders are placed in the same zone. A *spider* is either an existential spider or a universal spider. A spider *touches* a zone if one of its feet appears in that zone. A spider, s, is said to *inhabit* the region which is the union of the zones it touches. This region is called the *habitat* of s, denoted $\eta(s)$.

A region is *shaded* if each of its component zones is shaded. An *arrow* is a labelled, directed line from a spider to either an existential spider or a contour. The spider at the beginning of the arrow is called the *source* (of the arrow) and the spider or contour at the end is called the *target* (of the arrow). A *unitary diagram* is a finite collection of contours, spiders and arrows properly contained in a boundary rectangle. All universal spiders must be the source of an arrow and no two distinct arrows can have the same label, source and target.

For unitary diagram d define $L(d)$, $C(d)$, $Z(d)$, $Z^*(d)$, $ES(d)$, $US(d)$ and $A(d)$ to be the sets of contour labels, contours, zones, shaded zones, existential spiders, universal spiders and arrows of d respectively.

The diagram in Fig. 1 contains three contours and five zones, of which two are shaded. There are two arrows. The source of the arrow labelled $beenEatenBy$ is an existential spider with a habitat that is the basic region inside the contour labelled $Mice$. Its target is an existential spider. The other arrow, labelled $biggerThan$, has a universal spider as its source and its target is the contour labelled $Mice$.

B. Semantics of Unitary CD1 Diagrams

Regions in CD1 diagrams represent sets. An existential spider represents the existence of an element in the set represented by its habitat. Distinct existential spiders represent the existence of distinct elements. In the set represented by a shaded region, all the elements are represented by existential spiders. Arrow labels represent relations. An arrow, together with its source and target, represents a property of the relation represented by its label. A universal spider represents universal quantification over the set represented by its habitat. Universal quantification over the set represented by a zone can be represented by a single universal spider. Some problems interpreting constraint diagrams are raised in [4] and resolved in [2]. These problems do not arise in this system due to restrictions we place on the syntax (for example, we do not include *derived contours*) and semantics of CD1 diagrams.

The diagram in Fig. 2 expresses the fact that (the sets represented by) A and B are disjoint, B is not empty and there is an x in $U-(A\cup B)$ such that for all a in A the relational image of a under f is x. This diagram could also be interpreted as 'for all a in A, there exists an x in $U-(A\cup B)$ such that the relational image of a under f is x', but we will not allow such a reading. To avoid ambiguity in diagram reading and to make the system tractable, we restrict the semantic interpretation so that 'there exists' takes precedence over 'for all'. Relaxing this semantic constraint has non-trivial outcomes.

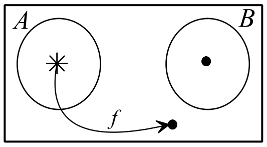

Fig. 2. A constraint diagram.

Later we will consider more than one unitary diagram. To ensure consistency of interpretation between diagrams we define $\mathcal{CL}$ and $\mathcal{AL}$ to be countably infinite sets of *contour labels* and *arrow labels* respectively, from which all labels will be drawn.

An *interpretation* of $\mathcal{CL}$ and $\mathcal{AL}$ is a triple $m=(\mathbf{U},\Psi,\phi)$ where $\mathbf{U}$ is a set, $\Psi\colon \mathcal{CL}\to\mathbb{P}\mathbf{U}$ is a function mapping contour labels to subsets of $\mathbf{U}$ and $\phi\colon \mathcal{AL}\to\mathbb{P}(\mathbf{U}\times\mathbf{U})$ is a function mapping arrow labels to relations on $\mathbf{U}$. To interpret a diagram first note that a zone can be identified by the contours that contain it. Define $c(z)$ and $e(z)$ to be the sets of labels of the contours in d that contain and exclude z respectively. We can then define $\Psi\colon Z(d)\to\mathbb{P}\mathbf{U}$ by

$$\Psi(z)=\bigcap_{l\in c(z)}\Psi(l)\cap\bigcap_{l\in e(z)}\overline{\Psi(l)}$$

where $\overline{\Psi(l)}=\mathbf{U}-\Psi(l)$. Further we define $\bigcap\limits_{l\in\emptyset}\Psi(l)=\bigcap\limits_{l\in\emptyset}\overline{\Psi(l)}=\mathbf{U}$. Any region is a union of zones, thus we define $\Psi\colon R(d)\to\mathbb{P}\mathbf{U}$ by

$$\Psi(r)=\bigcup_{z\subseteq r\wedge z\in Z(d)}\Psi(z).$$

For contour C we define $l(C)$ to be the label of C and also define $\Psi(C)=\Psi(l(C))$.

Next we formalize the notion of the image of a relation. Let R be a relation on a set $\mathbf{U}$. Define the *image* of R to be $im(\mathsf{R})=\{b\in\mathbf{U}:(a,b)\in\mathsf{R}\}$. Let A be a subset of $\mathbf{U}$. Define $A.\mathsf{R}$ to be $A.\mathsf{R}=im(\mathsf{R}\cap(A\times\mathbf{U}))$ and say $A.\mathsf{R}$ is the image of R with the domain restricted to A.

For each region, r, we define $S(r)$ and $T(r)$ to be the set of existential spiders that are completely within r and that touch r respectively. For each arrow, a, define $s(a)$, $t(a)$ and $l(a)$ to be the source, target and label of a respectively. Define $A_e(d)$ and $A_u(d)$ to be the sets of arrows in d with an existential source and universal source respectively. Arrows in $A_e(d)$ and $A_u(d)$ are called existential arrows and universal arrows respectively. The sets $A_e(d)$ and $A_u(d)$ partition $A(d)$.

The *semantics predicate*, $P_d(m)$, of a unitary diagram d is the conjunction of the following conditions.

(i) **Plane Tiling Condition.** All elements are in sets represented by zones:

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

(ii) There exists an extension of $\Psi\colon R(d) \rightarrow \mathbb{P}\mathbf{U}$ to $\Psi\colon R(d) \cup ES(d) \rightarrow \mathbb{P}\mathbf{U}$ such that the conjunction of the following conditions are satisfied.

  (a) **Spiders Condition.** Each existential spider represents the existence of an element in the set represented by its habitat:
  $\forall\, e \in ES(d) \bullet |\Psi(e)| = 1 \wedge \Psi(e) \subseteq \Psi(\eta(e))$.

  (b) **Strangers Condition.** No two existential spiders represent the existence of the same element:
  $\forall\, e_1, e_2 \in ES(d) \bullet \Psi(e_1) = \Psi(e_2) \Rightarrow e_1 = e_2$.

  (c) **Shading Condition.** Each shaded zone, z^*, represents a subset of the elements represented by the existential spiders touching z^*:
  $\forall\, z^* \in Z^*(d) \bullet \Psi(z^*) \subseteq \bigcup_{e \in T(z^*)} \Psi(e)$.

  (d) **Existential Arrows Condition.** For any existential arrow, a, the image of $\phi(l(a))$ with its domain restricted to $\Psi(s(a))$ equals $\Psi(t(a))$:
  $\forall\, a \in A_e(d) \bullet \Psi(s(a)).\phi(l(a)) = \Psi(t(a))$.

  (e) **Universal Arrows Condition.** For any universal arrow, a, the image of $\phi(l(a))$ with its domain restricted to any element in the set represented by the habitat of $s(a)$ equals $\Psi(t(a))$:
  $\forall\, a \in A_u(d)\, \forall\, x \in \Psi(\eta(s(a)))\bullet$
  $\{x\}.\phi((l(a)) = \Psi(t(a))$.

III. SATISFIABILITY

An interpretation m is said to *satisfy* diagram d, denoted $m \models d$, if $P_d(m)$ is true. If there exists an m such that $m \models d$ we say d is *satisfiable*. Diagram d_2 is a *logical consequence* of diagram d_1, denoted $d_1 \models d_2$, if every interpretation that satisfies d_1 also satisfies d_2. Diagrams d_1 and d_2 are semantically equivalent, denoted $d_1 \equiv_{\models} d_2$, if $d_1 \models d_2$ and $d_2 \models d_1$. Unlike spider diagrams, not all unitary CD1 diagrams are satisfiable.

The diagram in Fig. 3 is unsatisfiable. It expresses that there is an element in A that is related to exactly one element, x say, in $U - A$ under the relation f and exactly one element in $U - A$ distinct from x also under the relation f, which cannot happen.

A diagram d is an *α-diagram* if all of the existential spiders in d inhabit exactly one zone. We give syntactic criteria for identifying whether or not a unitary α-diagram is satisfiable.

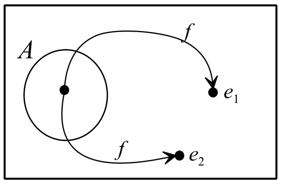

Fig. 3. An unsatisfiable unitary constraint diagram.

For each arrow, a in unitary α-diagram d, define the set of *hit existential spiders* of a, denoted $hit(a)$, to be

(i) $hit(a) = \{t(a)\}$ if $t(a) \in ES(d)$ and
(ii) $hit(a) = S(t(a))$ if $t(a) \in C(d)$,

where, for contour A, $S(A)$ is the set of existential spiders whose habitat is completely within A. The arrows of d are *pairwise compatible* if and only if

(i) every pair of arrows with the same existential spider as their source and the same label have the same hits and
(ii) every pair of arrows with the same universal spider, s, as their source and the same label have the same hits or the habitat of s is not inhabited by any existential spider and
(iii) if an existential spider has the same habitat as a universal spider and both are the source of arrows with the same label then these arrows have the same hits.

If the arrows of d are not pairwise compatible then d is said to contain *incompatible arrows*.

In Fig. 3 one arrow with label f has hit $\{e_1\}$ and the other has hit $\{e_2\}$. Since these arrows have the same source they are incompatible, failing condition (i). Since all unitary spider diagrams are satisfiable [9], incompatible arrows provide the only source of unsatisfiability for unitary α-diagrams.

Theorem 1: Unitary α-diagram d is satisfiable if and only if the arrows of d are pairwise compatible.

IV. REASONING RULES FOR UNITARY DIAGRAMS

In this section we give informal descriptions of purely syntactic reasoning rules which turn one unitary CD1 diagram into another. Some of the reasoning rules for CD1 diagrams are modifications and extensions of those in [9] and, in addition, new rules relating to arrows are included.

Rule 1: **Inconsistency.** A unitary α-diagram that contains incompatible arrows can be replaced by any diagram.

Rule 2: **Erasure of an arrow.** We may erase any arrow. If erasing an arrow results in a universal spider that is no longer the source of an arrow then that spider is also erased.

The arrow labelled f is erased from diagram d_1 in Fig. 4.

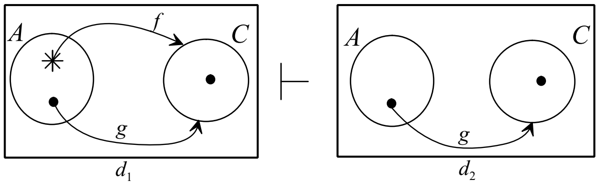

Fig. 4. An application of rule 2.

Rule 3: **Erasure of a contour.** We may erase any contour that is not the target of an arrow. When a contour is erased the following occurs.

- Any shading in only part of a zone is erased.
- If an existential spider has feet in two zones that combine to form a single zone then these feet are replaced by a single foot connected to the rest of the spider. Any arrows sourced (targeted) on this spider are still sourced (targeted) on this spider.
- Suppose there are universal spiders in the two old zones that combine to form a single zone and both spiders are

sources of an arrow with the same label and target, called *common arrows*. A new universal spider is placed in the new zone and is the source of one arrow for each pair of common arrows with the same label and target as the common arrows. All the old universal spiders whose habitat is a zone that combine with another zone are deleted, along with their arrows.

The contour with label B can be erased from d_1 in Fig. 5 since it is not the target of any arrow. The existential spider in d_1 is replaced by a single footed spider in d_2 with a habitat that is the new zone outside A and C. The universal spider inside A is retained, along with the arrows sourced on it. The two universal spiders in d_1 that are outside both A and C have habitats combine to form a single zone. Both are sources of arrows labelled g, targeted on C. These two universal spiders are replaced by a single universal spider in d_2 with habitat outside A and C. The universal spider in d_1 inhabiting the zone outside A, B and C is also the source of an arrow labelled h and target A. Since the universal spider inside B is not the source of an arrow with label h and target A, this arrow is 'lost' in d_2.

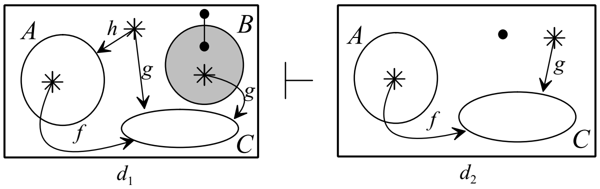

Fig. 5. An application of rule 3.

An application of any of the previous rules (potentially) loses information. The remaining rules in this section preserve semantic information and are therefore reversible.

Rule 4: **Introduction of a contour.** A contour may be drawn in the interior of the boundary rectangle provided the following occurs.

- The new contour has a label not present in the diagram.
- Each zone splits into two zones and shading is preserved.
- Each foot of an existential spider is replaced by a connected pair of feet – one in each new zone of the habitat.
- Each universal spider, u, is replaced by a pair of universal spiders – one in each zone of the habitat. Each arrow, a, sourced at u is replaced by a pair of arrows with the same label and target as a, one sourced on each new universal spider.

Fig.6 shows an application of rule 4.

Rule 5: **Introduction of an arrow: universal equivalence.** Let d be a diagram with a shaded zone z^* where every existential spider that touches z^* is the source of an arrow with label l and target t (l and t are fixed). Then we can introduce a universal arrow (and if necessary a universal spider) whose source inhabits z^*, labelled l with target t provided that the new arrow is not already present in d.

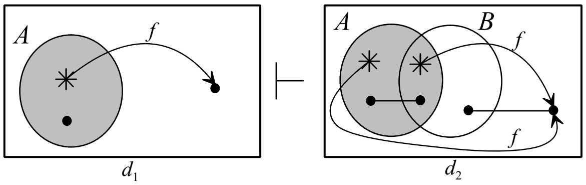

Fig. 6. An application of rule 4.

In Fig. 7, diagram d_1 expresses that each element in A has relational image, under f, that is B. Therefore a universal spider can be introduced to the zone inside A, which is the source of an arrow with label f, targeted on B.

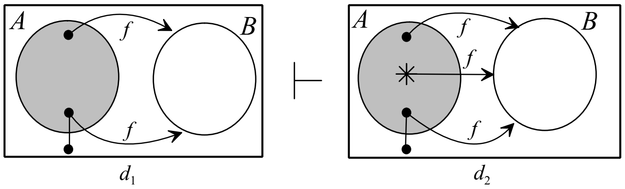

Fig. 7. An application of rule 5.

Rule 6: **Introduction of an arrow: universal deduction.** Let d be a diagram with a universal arrow, a, and an existential spider, e with the same habit as the source of a. Then we can introduce an arrow with source e and the same label and target as a provided that the new arrow is not already present in d.

Diagram d_1 in Fig. 8 expresses that there is exactly one element, x say, in B that each element in A is related to under f. Therefore the element represented by the existential spider inhabiting the zone within A is related to x under f.

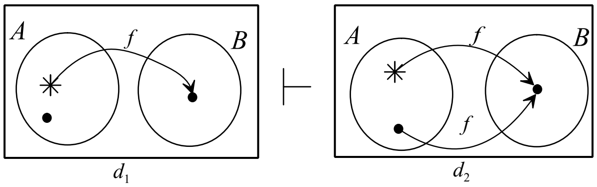

Fig. 8. An application of rule 6.

The following three rules allow us to introduce an arrow with the same source as an existing arrow, but with a new target. The naming of each of these rules refers to the targets of the existing and introduced arrows.

Rule 7: **Introduction of an arrow: spider to contour.** Let d be a diagram with an arrow, a, whose target is an existential spider, e. If the habitat of e is entirely within a contour, C, whose basic region is shaded and touched by no other existential spiders, then we can introduce an arrow with the same source and label as a, whose target is C provided that the new arrow is not already present in d.

Diagram d_1 in Fig. 9 expresses that there is exactly one element, x say, in B and all elements in A are related to x under f. Thus the image of any element in A under f is B.

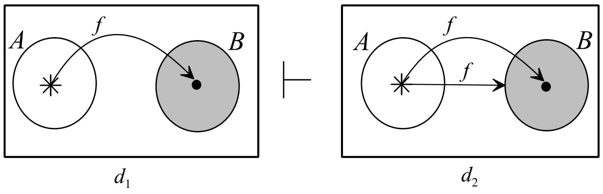

Fig. 9. An application of rule 7.

Rule 8: **Introduction of an arrow: contour to spider.** Let d be a diagram with an arrow, a, whose target is a contour, C. If the basic region inside C is shaded and there is an existential spider, e, such that $S(C) = \{e\} = T(C)$ then we can introduce an arrow with the same source and label as a whose target is e provided that the new arrow is not already present in d.

Rule 9: **Introduction of an arrow: contour to contour.** Let d be a diagram with a pair of contours, C_1 and C_2, whose symmetric difference is shaded and not touched by any existential spider and C_1 is the target of an arrow, a. Then we can introduce an arrow to d with the same source and label as a and target C_2 provided that the new arrow is not already present in d.

Diagram d_1 in Fig. 10 expresses that there is an element, x say, in A that is related to each element in B under f. Furthermore d expresses that B represents the same set as C. Thus x is related to each element in C under f.

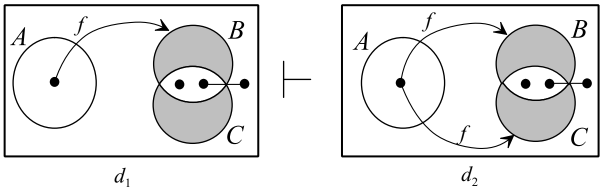

Fig. 10. An application of rule 9.

Rule 10: **Equivalent forms.** If d is not in Venn form we can introduce a new, shaded zone, that is not touched, to d.

To diagram d_1 in Fig. 11 we can introduce a shaded zone contained within the contours labelled B and C, shown in d_2.

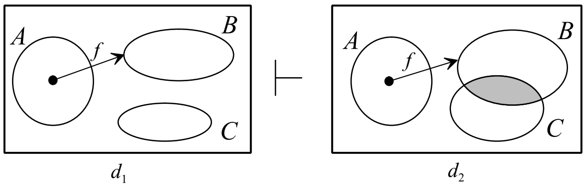

Fig. 11. An application of rule 10.

V. CONNECTING DIAGRAMS

Unitary diagrams form the building blocks of more complicated diagrams. To enable us to present disjunctive and conjunctive information, we introduce connectives: $\sqcup$ and $\sqcap$. If D_1 and D_2 are CD1 diagrams then so are $D_1 \sqcup D_2$ (pronounced D_1 or D_2) and $D_1 \sqcap D_2$ (pronounced D_1 and D_2). If $D = D_1 \sqcup D_2$ then the *semantics predicate*, $P_D(m)$, of D is $P_D(m) = P_{D_1}(m) \vee P_{D_2}(m)$. If $D = D_1 \sqcap D_2$ then the *semantics predicate*, $P_D(m)$, of D is $P_D(m) = P_{D_1}(m) \wedge P_{D_2}(m)$.

We now introduce three reversible reasoning rules that transform a unitary diagram into a disjunction of unitary diagrams.

Rule 11: **Splitting existential spiders.** Let d be a diagram containing an existential spider, e, whose habitat partitions into regions r_1 and r_2. We can replace d by $d_1 \sqcup d_2$, where d_1 and d_2 are a copies of d except that the habitat of e is r_1 in d_1 and r_2 in d_2. Any arrows sourced (targeted) on e in d are still sourced (targeted) on e in both d_1 and d_2.

Diagram d in Fig. 12 expresses that there is an element in A that is related to an element, x say, in $U - A$ under f. So x is either in B or $U - (A \cup B)$. We can split the spider representing the existence of x into two parts, one inside B and the other inside $U - (A \cup B)$ giving $d_1 \sqcup d_2$.

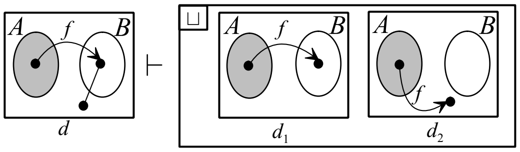

Fig. 12. An application of rule 11.

Rule 12: **Excluded middle for regions.** Let d be a diagram with a non-shaded region, r. We can replace d by $d_1 \sqcup d_2$, where d_1 and d_2 are copies of d except that r is shaded in d_1 and r contains an additional existential spider in d_2.

The excluded middle for regions rule is applied to diagram d in Fig. 13. We shade $B - C$ (giving d_1) and add an existential spider to $B - C$ (giving d_2), as shown in $d_1 \sqcup d_2$.

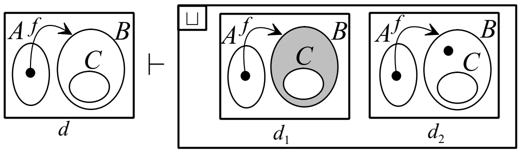

Fig. 13. An application of rule 12.

Rule 13: **Excluded middle for arrows.** Let d be an α-diagram such that every zone in d is shaded and, for each subset of $ES(d)$, E_i, such that $|E_i| \neq 1$, there is a contour, A, such that $S(A) = E_i$. Let $l \in \mathcal{AL}$ and let e be an existential spider in d that is not the source of an arrow with label l. Define $\mathcal{E}(d,l)$ to be the set of unitary diagrams, d_j, each of which is a copy of d except that d_j contains an additional arrow with source e, label l and any target. Then we may replace d by $\bigsqcup\limits_{d_j \in \mathcal{E}(d,l)} d_j$.

An application of this rule is illustrated in Fig. 14. Since every possible subset of U is represented, we can deduce that any given element must be related to nothing, itself, the other element or both elements under the relation l.

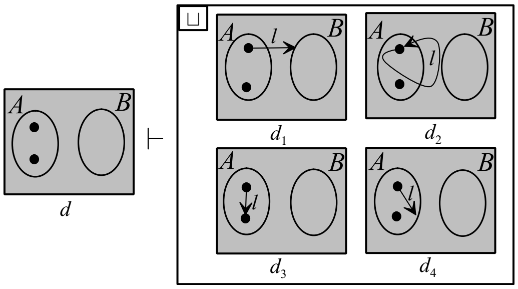

Fig. 14. An application of rule 13.

There are also many rules (not necessarily reversible), omitted for space reasons, that have analogies in propositional logic, for example associativity and distributivity.

VI. DISJUNCTIFYING DIAGRAMS

We now introduce a further rule that allows us to replace a diagram with a disjunction of unitary diagrams. The spider diagram version of this rule is essential to the completeness proof strategies used in spider diagram systems. To extend the strategy to the CD1 system, we require a constraint diagram version of this rule and we call the process *disjunctification*. The basic operation of disjunctification is performed on unitary α-diagrams which have the same sets of contour labels. In spider diagram systems, disjunctifying two such unitary diagrams results in a unitary diagram. For CD1 diagrams this is not the case. Firstly we consider an example where disjunctifying two unitary diagrams gives a unitary diagram. Diagram $d_1 \sqcap d_2$ in Fig. 15 is semantically equivalent to the unitary diagram d.

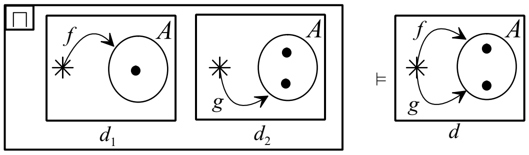

Fig. 15. Disjunctifying constraint diagrams.

Next we consider an example where disjunctifying does not result in a unitary diagram. Diagrams $d \sqcap d'$ and $d_1 \sqcup d_2 \sqcup d_3 \sqcup d_4$ in Fig. 16 are semantically equivalent. In $d \sqcap d'$ the spiders inside A could represent the same element or distinct elements. Similarly for B. This pair of choices gives four alternatives, each represented by one of d_1, d_2, d_3 and d_4.

To define the *disjunctification* of two unitary diagrams we first identify the unitary components that form the disjunctification, called *partial combinations*. Define zones z_1 and z_2 to be *corresponding* if and only if $c(z_1) = c(z_2)$ and $e(z_1) = e(z_2)$. Corresponding zones represent the same set. A more thorough treatment of corresponding regions can be found in [8].

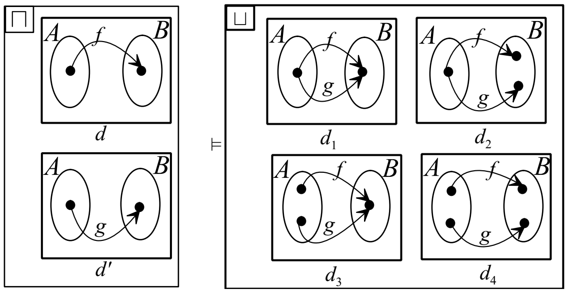

Fig. 16. Disjunctifying constraint diagrams.

Let d_0 and d_1 be two unitary α-diagrams such that each zone in d_0 has a corresponding zone in d_1 and vice versa. We say the zone sets of d_0 and d_1 are corresponding, denoted $Z(d_0) \equiv Z(d_1)$. Let d be a unitary α-diagram that does not contain incompatible arrows. Then d is called a *partial combination* of d_0 and d_1 if and only if each of the following are satisfied.

(i) The zone sets of d and d_0 are corresponding.
(ii) All zones in d that have a corresponding shaded zone in either d_0 or d_1 are shaded in d and no other zones are shaded in d.
(iii) The number of existential spiders in any shaded zone in d is the maximum number of existential spiders inhabiting any corresponding zone in d_0 or d_1.
(iv) The number of existential spiders in any unshaded zone in d is at most the maximum of
    (a) the number of existential spiders inhabiting the corresponding zone in d_0,
    (b) the number of existential spiders inhabiting the corresponding zone in d_1,
    (c) the sum total of existential spiders that are sources or targets of arrows inhabiting corresponding zones in d_0 and d_1.
(v) The number of existential spiders in any zone in d is at least the largest number in one of the corresponding zones in d_0 and d_1.
(vi) There is a universal spider in a zone in d if there is one in a corresponding zone in d_0 or d_1.
(vii) All the arrows in d_0 occur in d, similarly for d_1, and no others.

We define $\mathcal{D}_{pc}(d_0 \sqcap d_1)$ to be the set of partial combinations of $d_0 \sqcap d_1$.

It may be that $\mathcal{D}_{pc}(d_0 \sqcap d_1) = \emptyset$, for example, if one of d_0 and d_1 contains incompatible arrows. Thus it is useful to define an unsatisfiable unitary diagram denoted by $\perp$.

Let d_0 and d_1 be unitary α-diagrams such that $Z(d_0) \equiv Z(d_1)$ or $d_0 = \perp$ or $d_1 = \perp$. Define the *disjunctification* of d_0 and d_1, denoted $d_0 * d_1$, as follows.

1) If $d_0 = \perp$ or $d_1 = \perp$ then $d_0 * d_1 = \perp$.
2) If a zone in one diagram contains more existential spiders than in a corresponding shaded zone in the other diagram then $d_0 * d_1 = \perp$.
3) If $\mathcal{D}_{pc}(d_0 \sqcap d_1) = \emptyset$ then $d_0 * d_1 = \perp$.

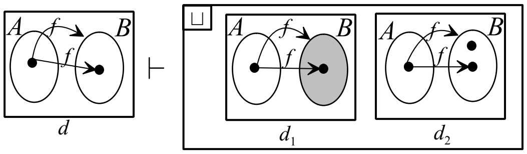

Fig. 17. The first step towards introduction of shading.

4) Otherwise $d_0 * d_1 = \bigsqcup_{d\in\mathcal{D}_{pc}(d_0\sqcap d_1)} d$.

To summarize, $d_0 * d_1 \equiv_{\models} d_0 \sqcap d_1$ and $d_0 * d_1$ is a disjunction of unitary α-diagrams.

Rule 14: **Disjunctifying unitary α-diagrams.** Let d_0 and d_1 be unitary α-diagrams such that $Z(d_0) = Z(d_1)$ or $d_0 =\perp$ or $d_1 =\perp$. Then $d_0 \sqcap d_1$ may be replaced by $d_0 * d_1$. This rule is reversible.

VII. OBTAINABILITY AND DERIVED RULES

Let D_1 and D_2 be two diagrams. Define $D_1 \Vdash D_2$ if and only if D_1 can be transformed to D_2 by a single application of one of the reasoning rules. D_2 is *obtainable* from D_1, denoted $D_1 \vdash D_2$, if and only if there is a sequence of diagrams $\langle D^1, D^2, ..., D^m\rangle$ such that $D^1 = D_1$, $D^m = D_2$ and $D^k \Vdash D^{k+1}$ for each k (where $1 \leq k < m$).

Let r be a reasoning rule. If, whenever D_1 can be transformed into D_2 by a single application of r, there is a sequence of reasoning rules distinct from r yielding $D_1 \vdash D_2$ then we say r is *derived*. The remaining rules we present are derived. Although not necessary for completeness, these rules aid the reasoning process.

Rule 15: **Erasure of shading.** If d is a unitary diagram with a shaded region r we may erase the shading from r [11].

Rule 16: **Erasure of an existential spider.** If d is a unitary diagram with a spider s which is not the source or target of any arrow and whose habitat is a non-shaded region then we may erase s from d.

Another useful derived rule permits the introduction of shading to a unitary diagram, resulting in a semantically equivalent diagram.

Rule 17: **Introduction of shading.** Let d be a unitary α-diagram containing a non-shaded zone z. If introducing an existential spider to z would result in a diagram containing incompatible arrows then we can introducing shading to z.

From diagram d in Fig. 17 we can deduce that B contains exactly one element, because there is some element in A, x say, such that $x.f \in B$ and $x.f = B$. Apply the excluded middle for regions rule to d giving $d_1 \sqcup d_2$. Diagram d_2 contains incompatible arrows. Use the inconsistency rule to obtain $d_1 \sqcup d_1$ which is equivalent to d_1.

We now extend the disjunctification rule given for two unitary α-diagrams with corresponding zone sets to any diagram. To disjunctify diagrams in general we transform them into α-diagrams with corresponding zone sets. Take a diagram D. Let L be the union of all of the label sets of the unitary components of D. Introduce contours to each unitary component until they all have the same label sets and denote the resulting diagram D^L. Next, introduce zones until all unitary components have corresponding zone sets, giving D^Z. Apply the splitting spiders rule to the unitary components of D^Z until we obtain an α-diagram, and denote the resulting diagram ${}^{\alpha}D^Z$. The disjunctification of D, denoted D^*, is a disjunction of unitary α-diagrams defined recursively as follows.

- If ${}^{\alpha}D^Z$ is a unitary diagram then $D^* = {}^{\alpha}D^Z$.
- If ${}^{\alpha}D^Z = D_1 \sqcup D_2$ then $D^* = D_1^* \sqcup D_2^*$ where D_1^* and D_2^* are the disjunctifications of D_1 and D_2 respectively.
- If ${}^{\alpha}D^Z = D_1 \sqcap D_2$ then $D^* = \bigsqcup_{d\in\mathcal{D}} d$ where
$$\mathcal{D} = \{d_i * d_j : d_i \in comp(D_1^*) \wedge d_j \in comp(D_2^*)\}$$
and $comp(D_i^*)$ is the set of unitary components of which D_i^* consists.

Rule 18: **Disjunctification.** We may replace D with D^*. This rule is reversible.

VIII. SOUNDNESS AND COMPLETENESS

A reasoning rule, r, is *valid* if, whenever D_2 is obtained from D_1 by one application of r, $D_1 \Vdash D_2$ implies $D_1 \models D_2$.

All the reasoning rules are valid. Hence the system is sound.

Theorem 2: **Soundness.** Let D_1 and D_2 be constraint diagrams. If $D_1 \vdash D_2$ then $D_1 \models D_2$.

Theorem 3: **Completeness.** Let D_1 and D_2 be constraint diagrams. If $D_1 \models D_2$ then $D_1 \vdash D_2$.

The strategy for proving completeness of spider diagram systems extends to CD1. Part of the completeness proof strategy used in spider diagram systems begins with a disjunction of unitary α-diagrams (acquired using rule 18) and uses the excluded middle for regions rule to add shading and spiders to produce a disjunction of unitary β-*diagrams*. A β-diagram is an α-diagram in which every zone is either shaded or touched by an existential spider. For spider diagrams, there are simple syntactic checks that establish when one β-diagram is a logical consequence of another. If we consider the subset of all spider diagrams consisting of unitary β-diagrams with the same label set, then all the rules necessary for completeness are erasure rules.

In CD1 it is possible to transform any diagram into a disjunction of β-diagrams, in the same way. However it is not easy to determine whether one unitary β-diagram is a logical consequence of another. There are examples of unitary β-diagrams with the same label sets, where one, d_2 say, is a logical consequence of another but requires more complex rules than simple erasure of components to establish syntactic entailment. An example of two such diagrams is given in Fig. 18.

If we introduce 'all possible' syntactic elements to a unitary β-diagram, d_1, using our reasoning rules, giving d_2, then any unitary β-diagram, d_3, that is a logical consequence of d_1 will 'contain only syntactic elements that are in d_2'. Although the details are non-trivial, this allows us to establish that d_3 is

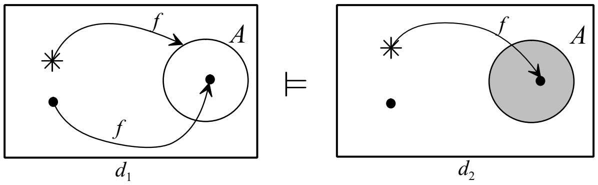

Fig. 18. Diagram d_2 is a logical consequence of d_1.

obtainable from d_1 by simply erasing components of d_2. This gives a completeness result for unitary β-diagrams.

We use the completeness result for unitary β-diagrams to prove completeness of the system. Consider two CD1 diagrams that satisfy $D_1 \vDash D_2$. All the rules that transform a CD1 diagram into a disjunction of β-diagrams are reversible, so we transform D_1 and D_2 into a disjunction of β-diagrams, ${}^{\beta}D_1$ and ${}^{\beta}D_2$ respectively, in the way outlined for spider diagrams. We then apply the excluded middle for regions rule repeatedly to ${}^{\beta}D_1$ until the number of existential spiders in each zone of each unitary component exceeds the number of existential spiders in any corresponding zone in any unitary component of ${}^{\beta}D_2$, giving a diagram D, say. We then add ‘all possible’ syntactic elements to D using our reasoning rules giving diagram D', say. We can then show each unitary component of D', say d, semantically entails a unitary component of ${}^{\beta}D_2$, say d_i. From the completeness result for unitary β-diagrams, $d \vdash d_i$ and it follows that $D_1 \vdash D_2$. Moreover, to prove completeness, we have constructed an algorithm to transform D_1 into D_2.

Theorem 4: **Decidability.** Let D_1 and D_2 be constraint diagrams. There is an algorithm which determines whether or not $D_1 \vdash D_2$.

IX. CONCLUSION

We have presented, informally, a sound and complete constraint diagram reasoning system. The syntactic conditions given to identify the satisfiability of unitary α-diagrams are sufficient, along with rule 18 (disjunctification), to identify the satisfiability of any diagram.

The CD1 system is more expressive than any spider diagram system. CD1 diagrams can express statements involving two place predicates whereas spider diagrams can only express statements involving one place predicates and equality. Previous work on spider diagrams provided a basis for the development of the CD1 system and it is anticipated that CD1 will provide a basis for the development of future constraint diagram systems. A long term aim is to develop sound and complete reasoning rules for the full constraint diagram notation and to develop software tools to aid the modelling and reasoning process.

The natural next step to take on this route would be to relax the constraint that ‘there exists’ takes precedence over ‘for all’. There are implications of this: if ‘for all’ takes precedence over ‘there exists’ it is not necessarily possible to split existential spiders. If we interpret d in Fig. 19 as ‘for all x in A there is a y in $U - A$ such that $x.f = y$’ we cannot split the existential spider in d. From diagram $d_1 \sqcup d_2$ we deduce for each x in A there is a y in B such that $x.f = y$ or for each x in A there is a y in $U - (A \cup B)$ such that $x.f = y$, which is not logically equivalent to the interpretation of d. Relaxing the constraint that ‘there exists’ takes precedence over ‘for all’ is likely to make the process of disjunctification more complicated: the operation will need to be defined for diagrams that are not necessarily α-diagrams.

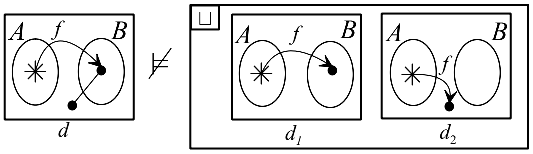

Fig. 19. Alternative semantics: incorrectly splitting spiders.

ACKNOWLEDGMENT

Gem Stapleton thanks the UK EPSRC for support under grant number 01800274. John Howse and John Taylor were partially supported by the UK EPSRC grant GR/R63516. Thanks also to Andrew Fish, Jean Flower and Chris John for their comments on earlier drafts of this paper. Thanks to the anonymous referees for their comments.

REFERENCES

[1] L. Euler. Lettres a une princesse d’allemagne, 1761.
[2] A. Fish, J. Flower, and J. Howse. A Reading algorithm for constraint diagrams. Accepted for *IEEE Symposium on Visual Languages and Formal Methods*, 2003 .
[3] J. Gil, J. Howse, and S. Kent. Formalising spider diagrams. In *Proceedings of IEEE Symposium on Visual Languages (VL99)*, pages 130–137. IEEE Computer Society Press, 1999.
[4] J. Gil, J. Howse, and S. Kent. Towards a formalization of constraint diagrams. In *Proc IEEE Symp. on Human-Centric Computing*, pages 72–79. IEEE Computer Society Press, Sept 2001.
[5] J. Gil, J. Howse, S. Kent and J. Taylor. Projections in Venn-Euler diagrams. In *Proc. IEEE Symposium on Visual Languages (VL2000)*, pages 119–126. IEEE Computer Society Press, Sept 2000.
[6] E. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.
[7] J. Howse, F. Molina, and J. Taylor. SD2: A sound and complete diagrammatic reasoning system. In *Proc. IEEE Symposium on Visual Languages (VL2000)*, pages 127–136. IEEE Computer Society Press, 2000.
[8] J. Howse, G. Stapleton, J. Flower, and J. Taylor. Corresponding regions in Euler diagrams. In *Proceedings of Diagrams 2002*, pages 146–160. Springer-Verlag, 2002.
[9] J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. In preparation, to appear: www.cmis.brighton.ac.uk/research/vmg.
[10] S. Kent. Constraint diagrams: Visualising invariants in object oriented models. In *Proceedings of OOPSLA97*, 1997.
[11] F. Molina. *Reasoning with extended Venn-Peirce diagrammatic systems*. PhD thesis, University of Brighton, 2001.
[12] OMG. UML specification, version 1.3. Available from www.omg.org.
[13] C. Peirce. Collected Papers. *Harvard University Press*, 1933.
[14] S.-J. Shin. *The Logical Status of Diagrams*. CUP, 1994.
[15] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *Phil.Mag*, 1880.