Journal of Visual Languages and Computing 24 (2013) 273–288

Contents lists available at SciVerse ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

On the expressiveness of spider diagrams and commutative star-free regular languages

Aidan Delaney [a,*], Gem Stapleton [b], John Taylor [a], Simon Thompson [b]

[a] Visual Modelling Group, University of Brighton, UK
[b] School of Computing, University of Kent, UK

ARTICLE INFO

Article history:
Received 22 September 2010
Received in revised form
3 December 2012
Accepted 14 February 2013
This paper has been recommended for acceptance by S.-K. Chang
Available online 27 February 2013

Keywords:
Spider diagrams
Diagrammatic logics
Regular languages
Expressiveness
Star-free languages

ABSTRACT

Spider diagrams provide a visual logic to express relations between sets and their elements, extending the expressiveness of Venn diagrams. Sound and complete inference systems for spider diagrams have been developed and it is known that they are equivalent in expressive power to monadic first-order logic with equality, MFOL[=]. In this paper, we further characterize their expressiveness by articulating a link between them and formal languages. First, we establish that spider diagrams define precisely the languages that are finite unions of languages of the form $K \sqcup\sqcup \Gamma^*$, where K is a finite commutative language and Γ is a finite set of letters. We note that it was previously established that spider diagrams define commutative star-free languages. As a corollary, all languages of the form $K \sqcup\sqcup \Gamma^*$ are commutative star-free languages. We further demonstrate that every commutative star-free language is also such a finite union. In summary, we establish that spider diagrams define precisely: (a) languages definable in MFOL[=], (b) the commutative star-free regular languages, and (c) finite unions of the form $K \sqcup\sqcup \Gamma^*$, as just described.

1. Introduction

Spider diagrams [7,11] provide a visual notation for finite sets, their members and interrelationships. An example spider diagram is shown in Fig. 1. They are given a model theoretic semantics, starting with a finite universal set, U, and interpreting each closed curve, called a *contour*, as a subset of U. Each minimal area (i.e. a region not further subdivided by any segment of a contour), called a *zone*, represents an intersection of sets and their complements. Between them, the zones represent the universal set: the union of the sets represented by the zones is U. For instance, in Fig. 1, there are two contours, representing the sets P and Q, along with four zones. The zone inside the contour P but outside the contour Q represents the set $P \cap \overline{Q}$. Each dot (or set of joined dots), called a *spider*, is interpreted as a distinct element of the universe belonging to the appropriate set. For example, the spider comprising a single dot in Fig. 1 tells us that there is an element in the set $P \cap \overline{Q}$ and the spider comprising two dots tells us that there is another element which is in the set $P \cap \overline{Q}$ or in the set $\overline{P} \cap \overline{Q}$; the line connecting the two dots represents disjunction. Finally, the set represented by a shaded zone can only include elements represented by spiders with a dot in that zone. In Fig. 1, therefore, the zone inside P but outside Q contains at most two elements, since there are two dots in this zone. This diagram is an example of a *unitary* spider diagram. More complex spider diagrams are formed by using logical operators, such as $\wedge$ and $\vee$. Spider diagrams also have an associated sound and complete reasoning system [20] and, in [11], Stapleton et al. showed that spider diagrams are equally as expressive as monadic first-order logic with

* Corresponding author. Tel.: +44 1273 642455.
E-mail addresses: a.j.delaney@brighton.ac.uk (A. Delaney), g.e.stapleton@brighton.ac.uk (G. Stapleton), john.taylor@brighton.ac.uk (J. Taylor), s.j.thompson@kent.ac.uk (S. Thompson).

1045-926X/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jvlc.2013.02.001

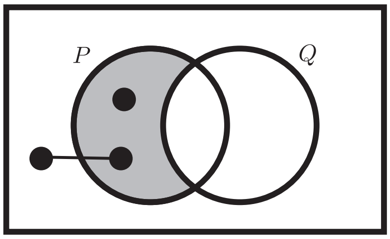

Fig. 1. A spider diagram.

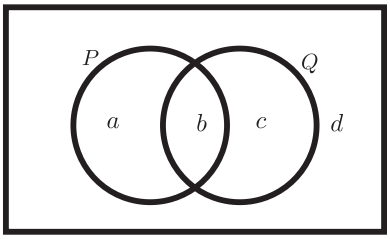

Fig. 2. An assignment of letters to zones.

equality, MFOL[=]. To illustrate this result, the spider diagram in Fig. 1 is equivalent to the MFOL[=] sentence

$$\exists x_1 \exists x_2 (P(x_1) \wedge \neg Q(x_1) \wedge \neg Q(x_2) \wedge x_1 \neq x_2 \wedge \forall y((P(y) \wedge \neg Q(y)) \Rightarrow (y = x_1 \vee y = x_2))).$$

In this paper, we present a novel approach to the study of spider diagrams, through examining their relationship with regular languages. Regular languages lie at the heart of theoretical computer science. Much is known about how they relate to finite automata, symbolic logic, and algebraic formalisms. Each of these relationships gives a different insight into regular languages as well as illuminating the other areas themselves. As with earlier work, our study of spider diagrams with regular languages provides insights into both regular languages and diagrammatic logic. For instance, we can now determine whether two spider diagrams are semantically equivalent by establishing whether they define the same language; two languages are equal if the minimal automata that accept them are the same.

We now explain how spider diagrams are associated with languages. The first step assigns sets of letters to contours, so that each zone corresponds to a single letter. If we have contours labelled P and Q, as in Fig. 1, and alphabet $\Sigma = \{a,b,c,d\}$ then we can assign $\{a,b\}$ to P and $\{b,c\}$ to Q. This induces an assignment of the letters to zones as follows:

1. a is assigned to the zone that is inside the contour P but outside the contour Q,
2. b is assigned to the zone that is inside both contours P and Q,
3. c is assigned to the zone that is inside the contour Q but outside the contour P, and
4. d is assigned to the zone that is outside both contours P and Q.

This assignment of letters to zones is illustrated in Fig. 2. Using this assignment, we can use spider diagrams to define languages by considering the information provided by the diagram. The presence of a spider in a diagram corresponds to the presence of a letter in a word. For instance, in Fig. 1, the spider comprising a single dot inside P but outside Q tells us that words must contain an a, and the other spider tells us that words must contain either an a (in addition to that present because of the first spider) or a d (because of the dot outside both contours). The disjunctive information arises from the fact that this spider comprises two dots connected by a line; the line represents disjunction. Thus, all words in the language defined by this spider diagram must contain one of the words aa, ad and da as a scattered subword (defined in Section 3) of a. The shading provides an upper bound on the number of occurrences of letters in words: all of the letters that are assigned to shaded zones must be represented by spiders. So, in Fig. 1, the shading tells us that the only a letters arise from spiders because the shaded zone is assigned the letter a. Apart from the restriction on the number of as, any other letters can be present. Thus, this spider diagram defines the language $\{aa,ad,da\} \sqcup\!\sqcup \{b,c,d\}^*$; this is the *shuffle product* of $\{aa,ad,da\}$ and $\{b,c,d\}^*$ which comprises of all words formed by interspersing the letters of words in $\{aa,ad,da\}$ with words in $\{b,c,d\}^*$. Of note is that spider diagrams cannot assert any ordering information between the letters of a word, so they define only commutative languages.

We connect our work to Thomas' definition of a language definable by a sentence in MFOL[<] [22]. Thomas proves that the star-free regular languages, including those which are not commutative, are precisely those definable in monadic first-order logic of order (MFOL[<]), in which the only binary predicate is $<$, interpreted as strict total order; the requirement for a strict total order arises from the fact that languages definable in MFOL[<] need not be commutative, so $<$ is necessary when placing constraints on the order of letters. The notion of when a MFOL[<] sentence defines a language requires a correspondence to be defined between monadic predicate symbols and sets of letters, just as we demonstrated when linking spider diagrams to languages in our example above by assigning sets of letters to contour labels. To illustrate, using the same example alphabet $\Sigma = \{a,b,c,d\}$, we assign the set $\{a,b\}$ to the predicate symbol P and the set $\{b,c\}$ to the predicate symbol Q, just as we assigned these sets to contours above. In this case, the MFOL[<] sentence

$$\exists x (P(x) \wedge \forall y (y \neq x \Rightarrow x < y))$$

defines the language consisting of all words that begin with a letter a or a letter b; intuitively, there is a letter ($\exists x$) that is in the set $\{a,b\}$ (since $P(x)$ holds and P is assigned $\{a,b\}$) that comes before every other letter (since $\forall y(y \neq x \Rightarrow x < y)$). Since this language is not commutative, it is not definable by a spider diagram or in MFOL[=].

In light of the observation that spider diagrams define commutative languages, spider diagrams of order were proposed [4], which are expressively equivalent to MFOL[<] [5], and therefore define all star-free regular