Spider diagrams of Order and a Hierarchy of Star-Free Regular Languages.

Aidan Delaney[1], John Taylor[1], and Simon Thompson[2]

[1] Visual Modelling Group, University of Brighton.
[2] Computing Laboratory, University of Kent.

Abstract. The spider diagram logic forms a fragment of constraint diagram logic and is designed to be primarily used as a diagrammatic software specification tool. Our interest is in using the logical basis of spider diagrams and the existing known equivalences between certain logics, formal language theory classes and some automata to inform the development of diagrammatic logic. Such developments could have many advantages, one of which would be aiding software engineers who are familiar with formal languages and automata to more intuitively understand diagrammatic logics. In this paper we consider relationships between spider diagrams of order (an extension of spider diagrams) and the star-free subset of regular languages. We extend the concept of the language of a spider diagram to encompass languages over arbitrary alphabets. Furthermore, the product of spider diagrams is introduced. This operator is the diagrammatic analogue of language concatenation. We establish that star-free languages are definable by spider diagrams of order equipped with the product operator and, based on this relationship, spider diagrams of order are as expressive as first order monadic logic of order.

1 Introduction

Regular languages are defined by Type-3 grammars [3]. They are the least expressive class of phrase structured grammars of the well-known Chomsky-Schützenberger hierarchy. Work by Büchi [2], amongst others, provides a logical characterisation of regular languages. The study of regular languages, finite automata and associated algebraic formalisms is one of the oldest branches of computer science. In contrast diagrammatic logics are relatively new. Their formal consideration can arguably be dated to the work of Barwise and Etchemendy [1], Shin [15], and Hammer [9] which in turn builds on the work of Euler [7] and Venn [19]. Spider diagrams [8] are a more recently defined forming a fragment of constraint diagrams [12]. Our interest is in the relationship between an extension of spider diagrams called spider diagrams of order and regular languages. This paper builds on our previous work [4, 5] and provides a proof that star-free regular languages are definable in spider diagrams of order, when augmented with a product operator. Star-free languages may be described by regular expressions without the use of the Kleene star, a fact from which the name of the language

class derives [13]. For example, the language a^* over the alphabet $\Sigma = \{a, b\}$ is star free as it may be written as the star-free expression $\overline{\overline{\emptyset}b\overline{\emptyset}}$ i.e. the complement of the set of all words containing a 'b'. The expression $\overline{\emptyset}$ is the complement of the empty set of words and may be read as the set of all words over Σ, denoted Σ^*. The language $(aa)^*$ over the same alphabet is not star-free [14].

Of most interest to us is the Straubing-Thérin hierarchy (STH), which is one of three infinite hierarchies within the class of star-free languages. The other two are the dot-depth hierarchy and the group hierarchy. All three hierarchies are recursively constructed from a base case at their respective level 0. Level $\frac{1}{2}$ of each hierarchy is the polynomial closure of level 0, an operation which is explained in section 5. Each of the fractional levels $\frac{1}{2}, 1+\frac{1}{2}, 2+\frac{1}{2}, \ldots$ are similarly formed. Level 1 of each hierarchy is the finite boolean closure of level $\frac{1}{2}$ under the operations and $\cap$, or $\cup$ and complement $^-$. In general, whole numbered levels $1, 2, 3, \ldots$ are the finite boolean closure of the half level beneath them [13].

The study of the relationship between spider diagrams of order and regular languages provides a novel view of both subjects. We show, in this paper, that the logic of spider diagram of order describes which correspond to well-known subsets of star-free languages. We have previously shown that spider diagrams (without order) describe (sub)sets of regular languages that are incomparable with well-known hierarchies such as the Straubin-Thérin or dot-depth hierarchies [5]. Conversely regular languages have helped to inform the development of spider diagrams. Our introduction of the product operator is motivated by previous results from the theory of formal languages [18]. By furthering the study of the relationship between diagrammatic logic and formal language theory we hope to "import" well-known results.

This paper presents an overview of the syntax and semantics of spider diagrams of order in section 2. In section 3 we define the language of a spider diagram of order, generalising work in [5]. The product of spider diagrams is introduced in section 4. The central result of this paper, that all star-free regular languages are definable in spider diagrams of order, is presented in section 5.

2 Syntax and semantics of spider diagrams of order

This section provides an overview of the syntax and semantics of spider diagrams of order, originally presented in [5] which in turn extends [11].

The diagrams within rectangular boxes labelled d_1 and d_2 in figure 1 are *unitary spider diagrams of order*. Such diagrams, like the Euler diagrams they are based on, are wholly contained within a rectangular box. Each unitary spider diagram of order consists of contours and spiders. Contours are simple closed curves. The spider diagram d_1 contains two labelled *contours*, P and Q. The diagram also contains three minimal regions, called *zones*. There is one zone inside the contour P, another inside the contour Q and the other zone is outside both contours P and Q. The unitary spider diagram d_2 contains four zones. One zone is outside both contours P and Q, another is inside the contour P but outside the contour Q, yet another is inside the contour Q but inside the contour

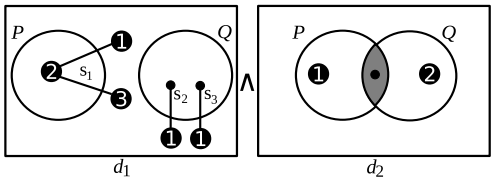

Fig. 1. A spider diagram of order.

P. The final zone is inside both contours and, in this example, is a shaded zone. Each zone in a unitary diagram d can be can be described by a two-way partition of d's contour labels.

In d_1, the zone inside P but outside Q and contains a vertex of one *spider*; spiders are trees whose vertices, called *feet*, are placed in zones. In general, any given spider may contain both *ordered feet* (those of the form ❶, ❷, ❸, ...) and *unordered feet* (those of the form •). In d_1, there is a single three footed spider labelled s_1 and two bi-footed spiders labelled s_2 and s_3. Spider diagrams can also contain *shading*, as in d_2.

Definition 1. *We define $\mathcal{C}$ to be a finite set of all contour labels used in spider diagrams. A **zone** is defined to be a pair, (in, out), of finite disjoint subsets of $\mathcal{C}$. The set "in" contains the labels of the contours that the zone is inside whereas "out" contains the labels of the contours that the zone is outside. The set of all zones is denoted $\mathcal{Z}$. A **region** of a diagram is a set of zones.*

The entire diagram in figure 1 is a compound spider diagram of order. It depicts the conjunction of statements made by its unitary components d_1 and d_2, denoted by the $\wedge$ symbol between the rectangular boxes. A $\vee$ symbol between boxes signifies disjunction between statements whereas a horizontal bar above a rectangle denotes negation.

The semantics of unitary spider diagrams of order are model based. In essence, contours represent sets and spiders represent the existence of elements. A model for a diagram is an assignment of sets to contour labels that ensures various conditions hold; these conditions are encapsulated by the *semantics predicate* defined below. To begin our formalisation of models, we start by defining spider feet and subsequently we define spiders. When we formalise the semantics, it is useful to have access to the region in which a spider is placed, called its *habitat*.

Definition 2. *A **spider foot** is an element of the set $(\mathbb{Z}^+ \cup \{\bullet\}) \times \mathcal{Z}$ and the set of all feet is denoted $\mathcal{F}$. A **spider**, s, is a set of feet together with a number: $s \in \mathbb{Z}^+ \times (\mathbb{P}\mathcal{F} - \{\emptyset\})$ and the set of all spiders is denoted $\mathcal{S}$. The **habitat** of a spider $s = (n, p)$ is the region habitat$(s) = \{z : \exists k\,(k, z) \in p\}$. A spider foot $(n, z) \in \mathcal{F}$ where $n \in \mathbb{Z}^+$ has **rank** n.*

Spiders are numbered because unitary diagrams can contain many spiders with the same foot set; essentially, we view a unitary diagram as containing a bag of spiders.

Definition 3. *A **unitary spider diagram of order** is a quadruple* $d = \langle C, Z, ShZ, SI \rangle$ *where*

$C = C(d) \subseteq \mathcal{C}$ *is a set of contour labels,*
$Z = Z(d) \subseteq \{(a, C(d) - a) : a \subseteq C(d)\}$ *is a set of zones,*
$ShZ = ShZ(d) \subseteq Z(d)$ *is a set of shaded zones,*
$SI = SI(d) \subsetneq \mathcal{S}$ *is a finite set of **spider identifiers** such that for all* $(n_1, p_1), (n_2, p_2) \in SI(d)$,

$$(p_1 = p_2 \implies n_1 = n_2) \wedge habitat(n_1, p_1) \subseteq Z(d).$$

The symbol $\perp$ *is also a unitary spider diagram. We define*

$$C(\perp) = Z(\perp) = ShZ(\perp) = SI(\perp) = \emptyset.$$

If d_1 *and* d_2 *are spider diagrams of order then* $(d_1 \wedge d_2), (d_1 \vee d_2)$ *and* $\neg d_1$ *are **compound spider diagrams of order**.*

It is useful to identify the set of spiders present in a diagram, which is implicit in the spider identifier set and to be able to arbitrarily select feet of spiders. For example, when defining the semantics, each spider, s, represents an element and the feet place a disjunction of constraints on that element; thus to identify whether an *interpretation* (see below) is a model for a unitary diagram there needs to be a choice of foot for which s satisfies the constraint imposed.

Definition 4. *The set of **spiders** in unitary diagram* d *is defined to be*

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

Let $FootSelect \colon S(d) \to \mathcal{F}$ *be a function. If, for all* $(n, p) \in S(d)$, $FootSelect(s) \in p$ *then* $FootSelect$ *is called a **foot selection function** for* d.

It is also useful to identify which zones could be present in a unitary diagram, given the label set, but are not present; semantically, *missing* zones provide information.

Definition 5. *Given a unitary diagram,* d, *a zone* (in, out) *is said to be **missing** if it is in the set* $\{(in, C - in) : in \subseteq C\} - Z(d)$ *with the set of such zones denoted* $MZ(d)$. *If* d *has no missing zones then* d *is in **Venn form** [11].*

Definition 6. *An **interpretation** is a triple* $(U, \Psi, <)$ *where* U *is a universal set and* $\Psi \colon \mathcal{C} \to \mathbb{P}U$ *is a function that assigns a subset of* U *to each contour label and* $<$ *is an irreflexive, antisymmetric and transitive relation on* U. *The function* Ψ *can be extended to interpret zones and sets of regions as follows:*

1. *each zone,* $(a, b) \in \mathcal{Z}$, *represents the set* $\bigcap_{l \in a} \Psi(l) \cap \bigcap_{l \in b} \overline{\Psi(l)}$ *and*
2. *each region,* $r \in \mathbb{P}\mathcal{Z}$, *represents the set which is the union of the sets represented by* r*'s constituent zones.*

For brevity, we will continue to write $\Psi\colon \mathcal{C} \to \mathbb{P}U$ but assume that the domain of Ψ includes the zones and regions also. Given an interpretation we wish to know whether it is a model for a diagram; in other words, when the information provided by the interpretation agrees with the intended meaning of the diagram. Informally, an interpretation is a *model* for unitary diagram d ($\neq\bot$) whenever

1. all of the zones which are missing represent the empty set,
2. all of the regions represent sets whose cardinality is at least the number of spiders placed entirely within that region and
3. all of the entirely shaded regions represent sets whose cardinality is at most the number of spiders with a foot in that region.
4. the elements represented by the spiders obey the ordering imposed on them by the rank of the spiders' feet.

We now make this notion precise.

Definition 7. *Let $I = (U, \Psi, <)$ be an interpretation and let d ($\neq\bot$) be a unitary spider diagram of order. Then I is a* ***model*** *for d if and only if the following conditions hold.*

1. ***The missing zones condition*** $\bigcup_{z \in MZ(d)} \Psi(z) = \emptyset$.
2. ***The function extension condition*** *There exists an extension of Ψ to spiders, $\Psi : \mathcal{C} \cup S(d) \to \mathbb{P}U$ which ensures the following further conditions hold.*
   (a) ***The habitats condition*** *All spiders represent elements (strictly, singleton sets) in the sets represented by their habitats:*
   $$\forall s \in S(d)\, \Psi(s) \subseteq \Psi(habitat(s)) \wedge |\Psi(s)| = 1.$$
   (b) ***The distinct spiders condition*** *Distinct spiders denote distinct elements:*
   $$\forall s_1, s_2 \in S(d) : \Psi(s_1) = \Psi(s_2) \implies s_1 = s_2.$$
   (c) ***The shading condition*** *Shaded regions represent sets containing elements denoted by spiders:*
   $$\Psi(ShZ(d)) \subseteq \bigcup_{s \in S(d)} \Psi(s).$$
   (d) ***The order condition*** *The ordering information provided by the spiders agrees with that provided by $<$: there exists a foot selection function, $FootSelect\colon S(d) \to \mathcal{F}$, for d such that*
   - *for all $s \in S(d)$, $FootSelect(s) = (n, z)$ implies $\Psi(s) \subseteq \Psi(z)$*
   - *for all $s_1, s_2 \in S(d)$ with $FootSelect(s_1) = (n_1, z_1)$ and $FootSelect(s_2) = (n_2, z_2)$, if $x < y$ where $\Psi(s_1) = \{x\}$ and $\Psi(s_2) = \{y\}$ then either $n_1 = n_2$ or $n_1 = \bullet$ or $n_2 = \bullet$ or $n_1 < n_2$.*

If $\Psi : \mathcal{C} \cup S(d) \rightarrow \mathbb{P}U$ ensures that the above conditions are satisfied then Ψ is a ***valid*** *extension to spiders for d. A foot selection function, $FootSelect : S(d) \rightarrow \mathcal{F}$, that ensures the above conditions are satisfied is also called* ***valid****. If $d = \perp$ then the interpretation is not a model for d.*

For compound diagrams, the definition of a model extends in the obvious inductive way.

3 The language of a spider diagram of order

A language is a set of words over a finite alphabet, typically denoted Σ. In order to discuss the language of a spider diagram of order we associate an alphabet with the contours that may appear in a diagram. A function is fixed mapping elements of $\mathcal{C}$ to sets of letters from the finite alphabet Σ. The use of this function allows us to consider the language of a diagram over an arbitrary alphabet. Previous work [5] considered a much more restricted set of alphabets. In the examples in figure 2 we assume $\mathcal{C} = \{P, Q\}$ and we assign the alphabet $\Sigma = \{a, b, c, d\}$ via a function called *lettermap* in the manner depicted in figure 2(a) i.e. $lettermap(P) = \{b, c\}$ and $lettermap(Q) = \{c, d\}$. It is important to note that the lower-case letters in figure 2(a) are not syntactic elements of spider diagrams of order. The depicted *lettermap* assignment satisfies the following defintion:

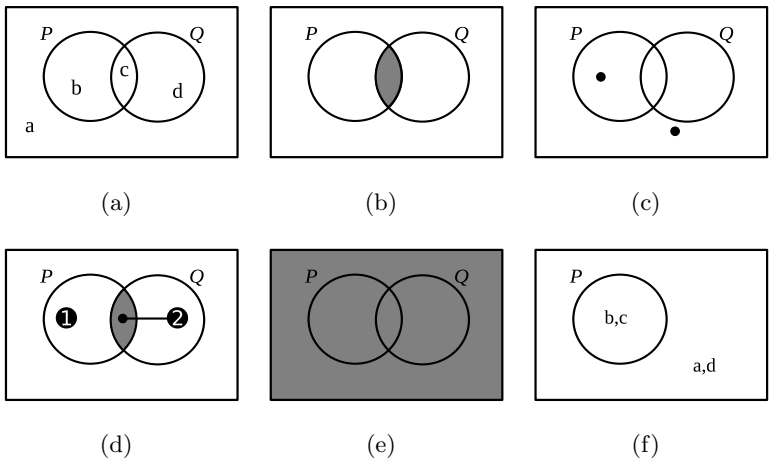

Fig. 2. Example unitary spider diagrams of order.

Definition 8. *The function lettermap : $\mathcal{C} \rightarrow \mathbb{P}\Sigma$ is a fixed assignment of contour labels to sets of letters. The function lettermap is extended to assign letters to zones as follows:*

1. *Each zone,*(in, out)*, maps to the set of letters inside the set of included contours but outside the set of excluded contours*

$$lettermap(in, out) = \bigcap_{c \in in} lettermap(c) \cap \bigcap_{c \in out} (\Sigma - lettermap(c)) \text{ and,}$$

2. *any zone,* (in, out)*, for which* $in \cup out = \mathcal{C}$ *is assigned at most one letter, this ensures that the spider diagram logic is capable of distinguishing each letter*

$$\text{if } in \cup out = \mathcal{C} \text{ then } |lettermap(in, out)| \leq 1.$$

Let a *be a letter in* Σ*. Then* $zone(a) = (inLabels(a), outLabels(a))$ *where we define*

- $inLabels(a) = \{c \in \mathcal{C} : a \in lettermap(c)\}$ *and*
- $outLabels(a) = \{c \in \mathcal{C} : a \notin lettermap(c)\}$.

We now have a relationship between the spider diagram logic over the labels in $\mathcal{C}$ and the alphabet Σ, therefore we may now discuss the language of the diagram. Figure 2(a) places no restrictions on the words of the language. This is because it contains no missing zones, no shading and no spiders. As such, the language of this diagram is the set of all words over the alphabet i.e. namely Σ^*. Figure 2(b) prevents words from containing a 'c' character. The words "abd", "aaaa" and "dbab" are in the language of the diagram, however unlike the words "c" and "abcd". Words in the language of the diagram are said to *correspond* to that diagram and the set of all such words is the *corresponding language*. Figure 2(c) asserts that words must contain a 'b' character and an 'a' character because of the presence of spiders. Words in the language of this diagram have a minimum length of two characters. The words "ab", "ba" and "aaabcd" are elements of the language. The word "acdd" is not, and neither is the word "cdbcd".

In figure 2(d) there are several restrictions placed on words in the language of the diagram. The first restriction imposed by the left-most spider is that the word must contain at least one 'b' character. This is because the spider asserts that the set represented by the zone $(\{P\}, \{Q\})$ is non-empty, thus any words corresponding to the diagram must contain at least one character specified by the previously given *lettermap* function. Further conditions are imposed by the right-most two-footed spider and the shaded zone. The right-most spider provides disjunctive information. In language terms it states that words corresponding to the diagram must contain either a 'c' or a 'd'. The shading indicates an upper bound on the number of 'c' letters in words of the language of the diagram. The final constraint is that a letter 'b' must occur before a letter 'd' if 'd' is a letter chosen based on the right-most spider. This is due to the explicit ordering prescribed by the spider feet. Thus the language of the diagram in figure 2(d) may be described by the regular expression

$$(a|b|d)^*b(a|b|d)^*d(a|b|d)^* \cup (a|b|d)^*b(a|b|d)^*c(a|b|d)^* \cup (a|b|d)^*c(a|b|d)^*b(a|b|d)^*.$$

The words of this language contain one 'b', at most one 'c' and if there is no 'c' then 'b' must occur before a 'd'. The diagram in figure 2(e) intuitively places

an upper-bound on all letters appearing in words of the corresponding language, the corresponding language is the set containing the empty word, $\{\lambda\}$.

Finally, consider the diagram in figure 2(f). The labels in this diagram are drawn from the set $\mathcal{C}$, although its label set contains only P. The models that satisfy the diagram in figure 2(a) also satisfy the diagram in figure 2(f). Similarly, the language that corresponds to the diagram in figure 2(a) also corresponds to the diagram in figure 2(f) due to the depicted assignment of letters to zones i.e. $lettermap(P) = \{b, c\}$ and $lettermap(Q) = \{c, d\}$. In general, semantically equivalent diagrams have the same language under any *lettermap* function.

In order to define how an interpretation models a word it is useful to treat a word as an array.

Definition 9. *Let w be a word of some language. $Array(w)$ is a set of pairs (a, i) where each i is a position in word w and a is the letter at position i.*

For example, consider the word $w = ab$ then $Array(w) = \{(a, 1), (b, 2)\}$.

The interpretation $I = (U, \Psi, <)$ is a model for word $w = ab$ given $lettermap(P) = \{a\}$ and $lettermap(Q) = \{b\}$ where

$U = \{1, 2\}$,
$\Psi(P) = \{1\}, \Psi(Q) = \{2\}$,
$<= \{(1, 2)\}$.

I models w as a bijection exists between U and the letters in w, namely $\{1 \mapsto a, 2 \mapsto b\}$, which respects the order relation $<$. Furthermore, the bijection and Ψ are in agreement on the assignment of letter sets to the interpretation of contours and zones. The following definition generalises the concept of a model of a word.

Definition 10. *Interpretation $I = (U, \Psi, <)$ is a **model** for a word w if there exists a bijection, f, between $Array(w)$ and U such that*

1. *the order of the letters in w is respected by $<$*

$$\forall (a_i, j), (a_k, l) \in Array(w) : f(a_i, j) < f(a_k, l) \Rightarrow j < l, \text{ and}$$

2. *the element in U chosen for the letter a_i is in a set represented by the zone that partitions $\mathcal{C}$ and contains a_i*

$$\forall (a_i, j) \in Array(w) : f(a_i, j) \in \bigcap_{P \in inLabels(a_i)} \Psi(P) \cap \bigcap_{Q \in outLabels(a_i)} \overline{\Psi(Q)}.$$

We may now define the language of a spider diagram of order.

Definition 11. *The **language of a diagram**, D, denoted $\mathcal{L}(D)$, is defined to be the set of words that are modelled by at least one interpretation that models D.*

Lemma 1. *The following properties hold for spider diagrams:*

- *The language of the disjunction of two diagrams is the union of languages of the components: $\mathcal{L}(D_1 \vee D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.*

– *Similarly, the language of the conjunction of two diagrams is the intersection of the components:* $\mathcal{L}(D_1 \wedge D_2) = \mathcal{L}(D_1) \cap \mathcal{L}(D_2)$.
– *The language of the negation of a diagram is the set complement of the language of the diagram with respect to the universe* Σ^**, formally:* $\mathcal{L}(\neg D_1) = \Sigma^* - \mathcal{L}(D_1) = \overline{\mathcal{L}(D_1)}$.

4 The product of spider diagrams

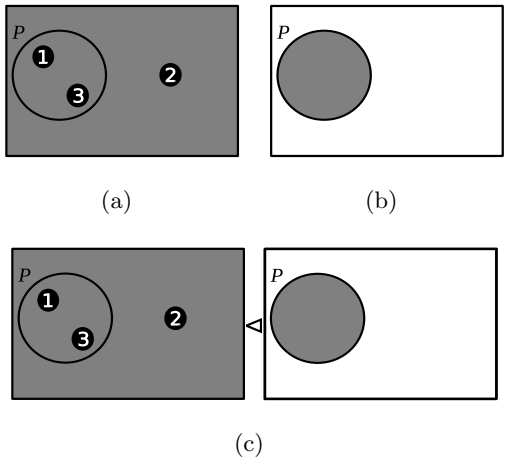

Fig. 3. Two unitary spider diagrams of order and their product.

The languages corresponding to the diagrams in figures 3(a) and 3(b) are aba and b^* (star-free $\overline{\overline{\emptyset}a\overline{\emptyset}}$) over $\Sigma = \{a, b\}$ respectively. The language $aba(b)^*$ is the concatenation of words from the language b^* and the word aba. It too is star-free and may be written as $aba\overline{\overline{\emptyset}a\overline{\emptyset}}$. Words in this language include any word which begins with the subword 'aba' followed zero or more 'b' characters eg: "aba", "abab" and "ababb". The spider diagram of order which defines the language $abab^*$ does not follow the intuitive structure concatenating the language aba with the language b^*. The language $abab^*$ is be defined by a spider diagram of order which is the conjunction of three unitary diagrams defining the languages $b^*ab^*ab^*$, $\overline{((a|b)^*a(a|b)^*b(a|b)^*b(a|b)^*a(a|b)^*)}$ and $\overline{((a|b)^*b(a|b)^*a(a|b)^*a(a|b)^*)}$. We show how diagrams define languages, such as these three, below. Furthermore, we strongly conjecture that there are star-free regular languages that do not correspond to any spider diagram of order.

To remove this expressiveness limitation we introduce the *product* of spider diagrams of order denoted $\lhd$ as depicted in figure 3(c). We first extend the syntax of spider diagrams of order to include $\lhd$ as a boolean operation.

Definition 12. *A diagram is a spider diagram of order if it*

- *is a unitary spider diagram of order,*
- *is of the form* $(D_1 \diamond D_2)$ *where* D_1 *and* D_2 *are spider diagrams of order and* $\diamond \in \{\vee, \wedge, \lhd\}$*, or*
- *is of the form* $(\neg D_1)$ *where* D_1 *is a spider diagram of order.*

In order to extend the semantics of spider diagrams of order we recall the following definition from [6].

Definition 13. *The* ***ordered sum*** *of two interpretations* $m_1 = (U_1, \Psi_1, <_1)$ *and* $m_2 = (U_1, \Psi_1, <_1)$ *denoted* $m_1 \lhd m_2$ *where* U_1 *and* U_2 *are disjoint is the interpretation* $m = (U, \Psi, <)$ *such that*

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

The definition for the model of the product of diagrams follows from the ordered sum of interpretations.

Definition 14. *Let* m *be an interpretation. Then* m *is a* ***model*** *for* $D_1 \lhd D_2$ *if there exists interpretations* m_1 *and* m_2 *such that* $m = m_1 \lhd m_2$ *and* m_1 *and* m_2 *are models for* D_1 *and* D_2 *respectively.*

Finally, we extend the properties of the language of a spider diagram (lemma 1) to include

- The language of the newly introduced product of two diagrams is the concatenation of its components $\mathcal{L}(D_1 \lhd D_2) = \mathcal{L}(D_1) \cdot \mathcal{L}(D_2)$.

The key insight underlying this is that the elements of a model of the compond diagram will be the disjoint union of elements for models of the constituent diagrams, with all the elements of the first being less than all the elements of the second under the ordering of the compound model. So, the word denoted by the compound diagram will be the concatenation of the words denoted by the individual diagrams. We may now see that the language corresponding to the diagram in figure 3(c) is $abab^*$. We now have a complete picture of the syntax and semantics of spider diagrams of order augmented with a product operator. Furthermore we have seen that the product of diagrams is a diagrammatic analogue of language catenation.

5 Comparing classes of regular languages and spider diagrams

In this section we prove that all star-free regular languages are definable using spider diagrams of order. We prove this by induction on the levels of the Straubing-Thérin hierarchy. Level 0 is our base case, it contains the langauges $\{\}$ and Σ^*. We assume that any language at level n of the STH is definable in

spider diagrams of order. As the STH has whole numbered levels $0, 1, 2, \ldots$ and half levels $0 + \frac{1}{2}, 1 + \frac{1}{2}, 2 + \frac{1}{2}$ our proof shows that languages at level $n + \frac{1}{2}$ and $n + 1$ are definable in spider diagrams of order. We also pay particular attention to the structure of diagrams which define languages at level $\frac{1}{2}$ and level 1 as these sets of languages are the well-known shuffle-ideal [10] and piecewise-testable [17] classes respectively.

Lemma 2. *The languages at level* 0 *of the Straubing-Thérin hierarchy are definable in spider diagrams of order.*

Proof. Let L be a language at level 0 of the STH. Then, by definition, L is either $\{\}$ or Σ^*. Considering the diagram d_1 in figure 4 it can be shown that $letterzone(\{\}, \{\}) = \Sigma$ so the language of the diagram d_1 is Σ^*, which is one element of level 0. Given any *lettermap* function $\mathcal{L}(\perp) = \{\}$ where $\perp$ is a unitary spider diagram of order, which is the other element of level 0.

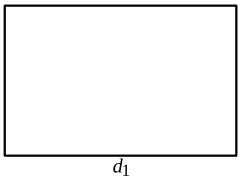

Fig. 4. Unitary spider diagram of order to which Σ^* corresponds.

We denote the unshaded diagram that contains no spiders or contours to which the language Σ^* corresponds by the symbol $\square$.

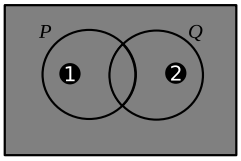

Fig. 5. The language $\{bd\}$ as a spider diagram of order.

The following lemma shows that the diagram in figure 5 may be constructed such that only the word "bd" corresponds to it. Due to the nature of the constructed diagram the *lettermap* function maps zones of the diagram to singleton sets or the empty set. In order to construct the diagram we examine the first

letter of the word and place a spider with single foot of rank 1 in the zone that *lettermap* states it is contained in. We repeat this exercise for the second letter however, this time, placing a spider with single foot of rank 2.

Lemma 3. *Any language containing a single word is definable by a unitary spider diagram of order.*

Proof. Let L be a set over alphabet Σ containing exactly one word w of length n and let $index(w, i)$ be a function returning the i^{th} character of word w. Furthermore, let d be a unitary spider diagram of order in Venn form where all contours from $\mathcal{C}$ are present all of the zones in d are shaded. As the diagram contains all contours and has no missing zones then, as a consequence of definition 8, *lettermap* must assign each zone to a singleton set containing a letter or the empty set.

For each i^{th} position of the word in L we place a single-footed spider in d with a foot of rank i in the zone $zone(\{index(w, i)\})$.

We denote a fully shaded diagram, as constructed in the above lemma, with corresponding single word language $\{w\}$ by the symbol $\blacksquare_w$.

Corollary 1. *Any arbitrary finite set of words of finite length is definable by a spider diagram of order.*

Proof. Let L be an arbitrary finite set of finite length words $\{w_1, w_2, \ldots, w_n\}$ over an alphabet Σ. By lemma 3 we may create a diagram $\blacksquare_{w_i}$ for each $w_i \in L$. The finite disjunction of these diagrams $\blacksquare_{w_1} \vee \blacksquare_{w_2} \vee \ldots \vee \blacksquare_{w_n}$ is the diagram to which L corresponds.

Languages at fractional levels of the STH are the polynomial closure of languages at the integer level beneath them.

Definition 15. *The **polynomial closure** of a set of languages S is finite union of languages of the form*

$$L_0 a_1 L_1 \ldots L_{n-1} a_n L_n$$

where each $L_i \in S$ and $a_j \in \Sigma$.

We now consider the form of diagrams that define the class of shuffle-ideal languages.

Lemma 4. *The languages at level $\frac{1}{2}$ of the Straubing-Thérin hierarchy (the class of shuffle-ideal languages) are definable by spider diagrams of order.*

Proof. Let L be a language at level $\frac{1}{2}$ of the STH. Languages at level $\frac{1}{2}$ are the polynomial closure of languages at level 0 [13]. The polynomial closure of level 0 results in languages of the form $\Sigma^* a_1 \Sigma^* \ldots \Sigma^* a_n \Sigma^*$ [10]. By lemma 3 we may construct a diagram $\blacksquare_{a_i}$ to which the language $\{a_i\}$ corresponds. Furthermore, by lemma 2 we may construct a diagram $\square$ to which the language Σ^* corresponds. Therefore, L corresponds to finite disjunctions of diagrams of the form

$$\square \lhd \blacksquare_{a_1} \lhd \square \lhd \blacksquare_{a_2} \ldots \square \lhd \blacksquare_{a_n} \lhd \square.$$

Lemma 4 may be developed into a more succinct characterisation.

Theorem 1. *The set of shuffle-ideal languages are definable by spider diagrams of order where each diagram is unshaded and the only binary connectives used are* $\vee$ *and* $\lhd$.

Proof. Let D be a diagram from lemma 4 to which a shuffle-ideal language corresponds. Then D is a finite disjunction of diagrams of the form

$$\Box \lhd \blacksquare_{a_1} \lhd \Box \lhd \blacksquare_{a_2} \ldots \Box \lhd \blacksquare_{a_n} \lhd \Box.$$

The diagram D' is semantically equivalent to D, where D' is a finite disjunction of diagrams of the form

$$\Box \lhd \blacksquare_{a_1} \lhd \Box \lhd \Box \lhd \blacksquare_{a_2} \lhd \Box \ldots \lhd \Box \lhd \blacksquare_{a_{n-1}} \lhd \Box \lhd \Box \lhd \blacksquare_{a_n} \lhd \Box.$$

In D' each diagram $\blacksquare_{a_i}$ is proceeded by a diagram $\Box$ and is followed by $\Box$. The diagram $\blacksquare_{a_{i+1}}$ is proceeded by a diagram $\Box$ which is not the diagram that follows $\blacksquare_{a_i}$. In language terms we are stating that the language

$$\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \ldots \Sigma^* a_n \Sigma^*$$

is equivalent to the language

$$\Sigma^* a_1 \Sigma^* \Sigma^* a_2 \Sigma^* \ldots \Sigma^* a_{n-1} \Sigma^* \Sigma^* a_n \Sigma^*.$$

Each $\Box \lhd \blacksquare_{a_i} \lhd \Box$ may be collapsed into a single diagram $\Box_{a_i}$ where $\Box_{a_i}$ is $\blacksquare_{a_i}$ in Venn form with the shading removed from all zones. The language $\Sigma^* a_i \Sigma^*$ corresponds to the diagram $\Box_{a_i}$. Therefore D is equivalent to finite disjunctions of diagrams of the form

$$\Box_{a_1} \lhd \Box_{a_2} \lhd \ldots \lhd \Box_{a_n}$$

where each $\Box_{a_i}$ is unitary and contains no missing or shaded zones.

Lemma 5. *The languages at level* 1 *of the Straubing-Thérin hierarchy are definable by spider diagrams of order.*

Proof. Languages at level 1 are formed by taking boolean combinations of languages at level $\frac{1}{2}$. The boolean operators $\cup$ and $^-$ over languages at level $\frac{1}{2}$ are semantically equivalent to $\vee$ and $\neg$ over the diagrams to which the languages at level $\frac{1}{2}$ correspond.

Given the example of a star-free language ab^* and the example of the regular but not star-free language $(aa)^*$ over alphabet $\Sigma = \{a, b\}$ presented in the Introduction, a reader may see how a spider diagram may be constructed such that its corresponding language is ab^*. The language ab^* may be written as the star-free expression $a\overline{\overline{\emptyset}a\overline{\emptyset}}$ or the diagram $\blacksquare_a \lhd (\neg(\Box \lhd \blacksquare_a \lhd \Box))$ (by theorem 1 a more succinct diagram may be derived). It may also be seen that it is impossible to construct a spider diagram of order to express $(aa)^*$. We now provide a lower bound on the class of languages definable by spider-diagrams of order

Theorem 2. *All star-free languages are definable by spider diagrams of order.*

Proof. From [18] we know that the Straubung-Thérin hierarchy, in the limit, contains all and only the star-free languages. We therefore prove the theorem inductively where lemma 2 is the base case. It is assumed that languages at level n of the Straubing-Thérin hierarchy to correspond to spider diagrams of order. We prove that languages at level $n + \frac{1}{2}$ and languages at level $n + 1$ correspond to spider diagrams of order.

Let L be a language, over Σ, at level $n + \frac{1}{2}$. Then L is the polynomial closure of level n i.e. finite unions of languages of the form

$$L_0 a_1 L_1 a_2 \dots a_{n-1} L_{n-1} a_n L_n$$

where each $a_i \in \Sigma$ and the languages L_j are the languages at level n. We may take the diagram D_j to which language L_j corresponds and create a diagram $\blacksquare_{a_i}$ (by lemma 3) to which the language $\{a_i\}$ corresponds. Then diagrams to which languages at level $n + \frac{1}{2}$ correspond are finite disjunctions of diagrams of the form

$$D_0 \lhd \blacksquare_{a_1} \lhd D_1 \lhd \blacksquare_{a_2} \lhd \dots \lhd \blacksquare_{a_{n-1}} \lhd D_{n-1} \lhd \blacksquare_{a_n} \lhd D_n.$$

Let L now be a language at level $n + 1$ of the STH. Languages at level $n + 1$ are the finite boolean closure (over union and complement) of languages at level $n + \frac{1}{2}$. Therefore L corresponds to a diagram in the finite boolean closure (over $\vee$ and $\neg$) of diagrams to which languages at level $n + \frac{1}{2}$ correspond.

From [16] we know that spider diagrams without order are expressively equivalent to monadic first order with equality, denoted $MFoL[=]$. Our extension to spider diagrams adds an order relation to the semantic models. Thus, an alternative statement of the theorem 2 may be made in terms of first order logic.

Theorem 3. *Spider diagrams of order are at least as expressive as logical sentences in monadic first order logic equipped with an order relation* $MFoL[<]$.

Proof. We recall from [18] that star-free regular languages are definable in $MFoL[<$] and the result that languages are definable in $MFoL[<]$ iff they are star-free. We have shown that all star-free languages correspond to spider diagrams of order when considered over the connectives $\vee, \wedge, \neg$ and the newly introduced $\lhd$. Therefore, spider diagrams of order are at least as expressive as $MFoL[<]$.

Given the established relationship between spider diagrams of order, star-free regular languages and monadic first order logic over structures containing an order relation we may now establish the expressive power of spider diagrams of order.

Theorem 4. *Spider diagrams of order are expressively equivalent to monadic first order logic of order.*

6 Conclusion

In this paper we have introduced a product operation on spider diagrams of order. This operation is a spider diagram analogue of language concatenation. We have further shown that spider diagrams of order, when augmented with this product operation, are as expressive as monadic first order logic of order. This increase in expressiveness was suggested by previous work in [5]. Our intention now is to further the results from [4] and develop an algorithm to construct a minimal finite state automaton given a spider diagram of order. Such an algorithm will answer questions concerning the succinctness of description afforded by spider diagrams of order when compared to finite automata.

References

1. J. Barwise and J. Etchemendy. *Hyperproof*. CSLI Press, 1994.
2. J. Büchi. Weak second order arithmetic and finite automata. *Z. Math. Logik und Grundl. Math.*, (6):66–92, 1960.
3. Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, pages 113–124, 1956.
4. Aidan Delaney and Gem Stapleton. On the descriptional complexity of a diagrammatic notation. In *Visual Languages and Computing*, September 2007.
5. Aidan Delaney and Gem Stapleton. Spider diagrams of order. In *International Workshop on Visual Languages and Logic*, September 2007.
6. Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, second edition, 1991.
7. L. Euler. Lettres a une princesse dallemagne sur divers sujets de physique et de philosophie. *Letters*, 2:102–108, 1775. Berne, Socit Typographique.
8. J. Gil, J. Howse, and S. Kent. Formalising spider diagrams. In *Proceedings of IEEE Symposium on Visual Languages (VL99), Tokyo*, pages 130–137. IEEE Computer Society Press, September 1999.
9. E. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.
10. Pierre-Cyrille Héam. On shuffle ideals. *Theoretical Informatics and Applications*, 36:359–384, 2002.
11. J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS Journal of Computation and Mathematics*, 8:145–194, 2005.
12. S. Kent. Constraint diagrams: Visualizing invariants in object oriented modelling. In *Proceedings of OOPSLA97*, pages 327–341. ACM Press, October 1997.
13. Jean-Eric Pin. *Syntactic semigroups*, pages 679–746. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
14. Jean-Eric Pin, Howard Straubing, and Denis Thérien. Some results on the generalized star-height problem. *Information and Computation*, 101:219–250, 1992.
15. S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.
16. G. Stapleton, S. Thompson, J. Howse, and J. Taylor. The expressiveness of spider diagrams. *Journal of Logic and Computation*, 14(6):857–880, December 2004.
17. Jacques Stern. Characterizations of some classes of regular events. *Theoretical Computer Science*, 35:17–42, 1985.
18. Wolfgang Thomas. Classifying regular events in symbollic logic. *Journal of Computer and System Sciences*, 25:360–376, 1982.
19. J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *Phil.Mag*, 1880.