A Compositional Distributional Model of Meaning

Stephen Clark **Bob Coecke** **Mehrnoosh Sadrzadeh**

Oxford University Computing Laboratory
Wolfson Building, Parks Road, OX1 3QD Oxford, UK
stephen.clark@comlab.ox.ac.uk – coecke@comlab.ox.ac.uk – ms6@ecs.soton.ac.uk

Abstract

We propose a mathematical framework for a unification of the distributional theory of meaning in terms of vector space models, and a compositional theory for grammatical types, namely Lambek's pregroup semantics. A key observation is that the monoidal category of (finite dimensional) vector spaces, linear maps and the tensor product, as well as any pregroup, are examples of compact closed categories. Since, by definition, a pregroup is a compact closed category with trivial morphisms, its compositional content is reflected within the compositional structure of any non-degenerate compact closed category. The (slightly refined) category of vector spaces enables us to compute the meaning of a compound well-typed sentence from the meaning of its constituents, by 'lifting' the type reduction mechanisms of pregroup semantics to the whole category. These sentence meanings live in a single space, independent of the grammatical structure of the sentence. Hence we can use the inner-product to compare meanings of arbitrary sentences. A variation of this procedure which involves constraining the scalars of the vector spaces to the semiring of Booleans results in the well-known Montague semantics.

Introduction

This paper combines, and then exploits, three results presented at the 2007 Quantum Interaction symposium. The first author and Pulman proposed the use of the Hilbert space tensor product to assign a distributional meaning to sentences (Clark & Pulman 2007). The second author showed that the Hilbert space formalism for quantum mechanics, when recast in category-theoretic terms, admits a true logic which enables automation (Coecke 2007). The third author showed that the logic of pregroup semantics for grammatical types is an essential fragment of this logic of the Hilbert space formalism (Sadrzadeh 2007).

The symbolic (Dowty, Wall, & Peters 1981) and distributional (Schuetze 1998) theories of meaning are somewhat orthogonal with competing pros and cons: the former is compositional but only qualitative, the latter is non-compositional but quantitative.[1] Following (Smolensky & Legendre 2005) in the context of Cognitive Science, where a similar problem exists between the connectionist and symbolic models of mind, the first author and Pulman argued for the use of the tensor product of vector spaces. They suggested that to implement this idea in linguistics one can, for example, traverse the parse tree of a sentence and tensor the vectors of the meanings of words with the vectors of their roles:

$$(\overrightarrow{\textit{John}} \otimes \overrightarrow{subj}) \otimes \overrightarrow{\textit{likes}} \otimes (\overrightarrow{\textit{Mary}} \otimes \overrightarrow{obj})$$

This vector in the tensor product space should then be regarded as the meaning of the sentence 'John likes Mary'. In this paper we will also pair vectors in meaning spaces and grammatical types, but in a way which overcomes some of the shortcomings of (Clark & Pulman 2007). One shortcoming is that, since inner-products can only be computed between vectors which live in the same space, sentences can only be compared if they have the same grammatical structure. In this paper we provide a procedure to compute the meaning of any sentence as a vector within a single space. A second problem is the lack of a method to compute the vectors representing the grammatical type; the procedure presented here does not require such vectors.

Abramsky and the second author proposed a *categorical quantum axiomatics* as a high-level framework to reason about quantum phenomena. Primarily this categorical axiomatics is a logic which lives on top of linear and multi-linear algebra and hence also applies to the use of vector space machinery outside the domain of quantum physics. The passage to the category-theoretic level allows for much simpler mathematical structures than vector spaces to exhibit quantum-like features. For example, while there is nothing quantum about sets, when organised in a category with relations (not functions) as morphisms and cartesian product (not disjoint union) as tensor, many typical quantum features emerge (Abramsky & Coecke 2004; Coecke 2005b). A syntax for these more general 'pseudo quantum categories' – more precisely, tensorial matrix calculi over some semiring – can be found in (Abramsky & Duncan 2006). In this paper we will exploit both the categorical logic which lives on top of vector spaces as well as the fact that it also lives on a much simper relational variant.

The use of pregroups for analysing the structure of natural languages is a recent development by (Lambek 1999)

[1] See (Gazdar 1996) for a discussion of the two competing paradigms in Natural Languge Processing.

which builds on the original Lambek calculus (Lambek 1958) where types are used to analyze the syntax of natural languages in a simple algebraic setting. Pregroups have been used to analyze the syntax of many languages, for example English (Lambek 2004), French (Bargelli & Lambek 2001b), Arabic (Bargelli & Lambek 2001a), Italian (Casadio & Lambek 2001), and Persian (Sadrzadeh 2006). They have also been provided with a Montague-style semantics (Preller 2007) which equips them with a translation into the lambda calculus and predicate logic. As discussed in (Sadrzadeh 2007) pregroups are posetal instances of the categorical logic of vector spaces as in (Abramsky & Coecke 2004) where juxtaposition of types corresponds to the tensor product of the monoidal category of vector spaces, linear maps and the tensor product. The diagrammatic toolkit of 'non-commutative' categorical quantum logic introduces reduction diagrams for typing sentences and allows the comparison of the grammatical patterns of sentences in different languages (Sadrzadeh 2006).

Here we blend these three ideas together, and articulate the result in a rigourous formal manner. We provide a mathematical structure where the meanings of words are vectors in vector spaces, their grammatical roles are types in a pregroup, and tensor product is used for the composition of meanings and types. We will pass from vector spaces to the category of vector spaces equipped with the tensor product, and refine this structure with the non-commutative aspects of the pregroup structure. Type-checking is now an essential fragment of the overall categorical logic. The vector spaces enrich these types with quantities: the reduction scheme to verify grammatical correctness of sentences will not only provide a statement on the well-typedness of a sentence, but will also assign a vector in a vector space to each sentence. Hence we obtain a theory with both pregroup analysis and vector space models as constituents, but which is inherently compositional and assigns a meaning to a sentence given the meanings of its words. The vectors $\overrightarrow{s}$ representing the meanings of sentences all live in the same meaning space S. Hence we can compare the meanings of any two sentences $\overrightarrow{s}, \overrightarrow{t} \in S$ by computing their inner-product $\langle \overrightarrow{s} | \overrightarrow{t} \rangle$.

Surprisingly, Montague semantics emerges as a simplified variant of our setting, by restricting the vectors to range over $\mathbb{B} = \{0, 1\}$, where sentences are simply true or false. Theoretically, this is nothing but the passage from the category of vector spaces to the category of relations as described in (Abramsky & Coecke 2004; Coecke 2005b). In the same spirit, one can look at vectors ranging over $\mathbb{N}$ or $\mathbb{Q}$ and obtain degrees or probabilities of meaning. As a final remark, in this paper we only set up our general mathematical framework and leave a practical implementation for future work.

Linguistic background

We briefly present two domains of Computational Linguistics which provide the linguistic background for this paper, and refer the reader to the literature for more details.

1. Vector space models of meaning. The key idea behind vector space models of meaning (Schuetze 1998) can be summed up by Firth's oft-quoted dictum that "you shall know a word by the company it keeps". The basic idea is that the meaning of a word can be determined by the words which appear in its contexts, where context can be a simple n-word window, or the argument slots of grammatical relations, such as the direct object of the verb *eat*. Intuitively, *cat* and *dog* have similar meanings (in some sense) because cats and dogs sleep, run, walk; cats and dogs can be bought, cleaned, stroked; cats and dogs can be small, big, furry. This intuition is reflected in text because *cat* and *dog* appear as the subject of *sleep*, *run*, *walk*; as the direct object of *bought*, *cleaned*, *stroked*; and as the modifiee of *small*, *big*, *furry*.

Meanings of words can be represented as vectors in a high-dimensional "meaning space", in which the orthogonal basis vectors are represented by context words. To give a simple example, if the basis vectors correspond to *eat*, *sleep*, and *run*, and the word *dog* has *eat* in its context 6 times (in some text), *sleep* 5 times, and *run* 7 times, then the vector for *dog* in this space is (6,5,7). The advantage of representing meanings in this way is that the vector space gives us a notion of distance between words, so that the inner product (or some other measure) can be used to determine how close in meaning one word is to another. Computational models along these lines have been built using large vector spaces (tens of thousands of context words/basis vectors) and large bodies of text (up to a billion words in some experiments). Experiments in constructing thesauri using these methods have been relatively successful. For example, the top 10 most similar nouns to *introduction*, according to the system of (Curran 2004), are *launch, implementation, advent, addition, adoption, arrival, absence, inclusion, creation.*

2. Pregroup semantics for grammatical types. Pregroup semantics was proposed by Lambek as a substitute for the Lambek Calculus (Lambek 1958), a well-studied type categorial grammar (Moortgat 1997). The main reason we chose the theory of pregroup semantics is its mathematical structure, which is ideal for the mathematical developments in this paper. The key ingredient is the mathematical object of a pregroup, which provides a tidy and simple algebraic structure to mechanically check if sentences of a language are grammatical.

A *partially ordered monoid* is a triple $(P, \leq, \cdot)$ where $(P, \leq)$ is a partially ordered set, $(P, \cdot)$ is a monoid with unit 1, and for all $a, b, c \in P$ with $a \leq b$ we have

$$c \cdot a \leq c \cdot b \qquad \text{and} \qquad a \cdot c \leq b \cdot c\,. \tag{1}$$

A *pregroup* is a partially ordered monoid $(P, \leq, \cdot)$ where each type $p \in P$ has a *left adjoint* p^l and a *right adjoint* p^r, which means that $p, p^l, p^r \in P$ satisfy

$$p^l \cdot p \leq 1 \leq p \cdot p^l \qquad \text{and} \qquad p \cdot p^r \leq 1 \leq p^r \cdot p\,.$$

From this it also follows that $1^l = 1^r = 1$.[2]

[2] Roughly speaking, the passage from Lambek Calculus to Pregroups is obtained by replacing the two residuals of the juxtaposition operator (monoid multiplication) with two unary operations. This enables us to work with a direct encoding of function arguments as adjoints rather than an indirect encoding of them through negation (implication by bottom).

Similar to type categorial grammars, one starts by fixing some basic grammatical roles and a partial ordering between them. We then freely generate a pregroup

$$(P, \leq, \cdot, (-)^l, (-)^r)$$

of these types where 1 is now the unit of juxtaposition, that is, the empty type. Examples of the basic types are:

- π for pronoun
- s for declarative statement
- q for yes-no question
- i for infinitive of the verb
- o for direct object.

In cases where the person of the pronoun and tense of the verb matters, we also have $\pi_j, s_k, q_k \in P$ for j'th person pronoun and k'th tense sentence and question. We require the following partial orders:

$$\pi_j \leq \pi \qquad s_k \leq s \qquad q_k \leq q$$

The adjoints and juxtapositions of these types are used to form the compound types. A type is assigned to each word in a sentence and the monoid multiplication is used for juxtaposition. The juxtaposition of adjacent adjoint types causes reduction to 1. This process is repeated until no more reduction is possible and a type is returned as the main type of the juxtaposition. If this type is the desired type (e.g. s for statement and q for question), the juxtaposition is a grammatical sentence. It has been shown in (Buszkowski 2001) that this procedure is decidable. Thus we obtain a decision procedure to determine if a given sentence of a language is grammatical.

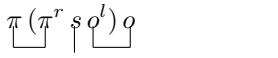
For simplicity, we use an arrow $\rightarrow$ for $\leq$ and drop the $\cdot$ between juxtaposed types. For the example sentence "He likes her", we have the following type assignment:

He	likes	her
π_3	$(\pi^r s o^l)$	o

for which we obtain the following reduction:

$$\pi_3(\pi^r s o^l)o \rightarrow \pi(\pi^r s o^l)o \rightarrow 1s1 \rightarrow s$$

The reduction can be represented diagrammatically by conjoining the adjacent adjoint types. For example, for the above reduction we have the following diagram:



The same method is used to analyze other types of sentences; for example, to type a yes-no question we assign (io^l) to the infinitive of the transitive verb and $(qi^l\pi^l)$ to 'does', and obtain the following reduction for 'Does he like her?':

Does	he	like	her?	$\rightarrow$	question
$(qi^l\pi^l)$	π_3	(io^l)	o	$\rightarrow$	q

The reduction diagrams are helpful in demonstrating the order of reductions, especially in compound sentences. They can also be used to compare the grammatical patterns of different languages (Sadrzadeh 2006).

Category-theoretic background

We now sketch the mathematical prerequisites which constitute the formal skeleton for the developments in this paper. Note that, in contrast to (Abramsky & Coecke 2004), here we consider the non-symmetric case of a compact closed category, non-degenerate pregroups being examples of essentially non-commutative compact closed categories. See (Freyd & Yetter 1989) for more details. Other references can be found in (Abramsky & Coecke 2004; Coecke 2005a; 2006).

1. Monoidal categories. The formal definition of monoidal categories is somewhat involved. It does admit an intuitive operational interpretation and an elegant, purely diagrammatic calculus. A (strict) monoidal category $\mathbf{C}$ requires the following data and axioms:

- a family $|\mathbf{C}|$ of *objects*;
  - for each ordered pair of objects (A, B) a corresponding set $\mathbf{C}(A, B)$ of *morphisms*; it is convenient to abbreviate $f \in \mathbf{C}(A, B)$ by $f : A \rightarrow B$;
  - for each ordered triple of objects (A, B, C), and each $f : A \rightarrow B$ and $g : B \rightarrow C$, there is a *sequential composite* $g \circ f : A \rightarrow C$; we moreover require that:

$$(h \circ g) \circ f = h \circ (g \circ f)\,;$$

  - for each object A there is an *identity morphism* $1_A : A \rightarrow A$; for $f : A \rightarrow B$ we moreover require that:

$$f \circ 1_A = f \qquad \text{and} \qquad 1_B \circ f = f\,;$$

- for each ordered pair of objects (A, B) a *composite object* $A \otimes B$; we moreover require that:[3]

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)\,;$$

- there is a *unit object* I which satisfies:[4]

$$\mathrm{I} \otimes A = A = A \otimes \mathrm{I}\,;$$

- for each ordered pair of morphisms $(f : A \rightarrow C, g : B \rightarrow D)$ a *parallel composite* $f \otimes g : A \otimes B \rightarrow C \otimes D$; we moreover require *bifunctoriality* i.e.

$$(g_1 \otimes g_2) \circ (f_1 \otimes f_2) = (g_1 \circ f_1) \otimes (g_2 \circ f_2)\,.$$

There is a very intuitive operational interpretation of monoidal categories. We think of the objects as *types of systems*. We think of a morphism $f : A \rightarrow B$ as a *process* which takes a system of type A (say, in some state ψ) as input and provides a system of type B (say, in state $f(\psi)$)

[3] In the standard definition of monoidal categories this 'strict' equality is not required but rather the existence of a *natural isomorphism* between $(A \otimes B) \otimes C$ and $A \otimes (B \otimes C)$. We assume strictness in order to avoid talking about natural transformations, and the corresponding *coherence conditions* between these natural isomorphisms. This simplification is justified by the fact that each monoidal category is categorically equivalent to a strict one, which is obtained by imposing appropriate congruences.

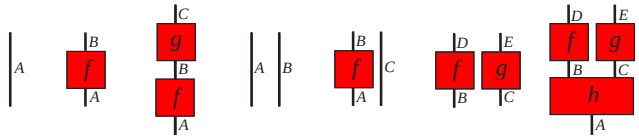
[4] Again we assume strictness in contrast to the usual definition.

as output. Composition of morphisms is sequential application of processes. The compound type $A \otimes B$ represents *joint systems*. We think of I as the trivial system, which can be either 'nothing' or 'unspecified'. More on this intuitive interpretation can be found in (Coecke 2006).

In the graphical calculus for monoidal categories we depict morphisms by boxes, with incoming and outgoing wires labelled by the corresponding types, with sequential composition depicted by connecting matching outputs and inputs, and with parallel composition depicted by locating boxes side by side. For example, the morphisms

$$1_A \quad f \quad g \circ f \quad 1_A \otimes 1_B \quad f \otimes 1_C \quad f \otimes g \quad (f \otimes g) \circ h$$

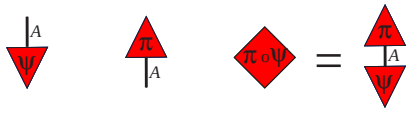
are depicted as:



The unit object I is represented by 'no wire'; for example

$$\psi : \mathrm{I} \to A \qquad \pi : A \to \mathrm{I} \qquad \pi \circ \psi : \mathrm{I} \to \mathrm{I}$$

are depicted as:



Morphisms $\psi : \mathrm{I} \to A$ are called *elements* of A. Operationally one can think of them as the *states* of system A.

2. Compact closed categories. A monoidal category is compact closed if for each object A there are also objects A^r and A^l, and morphisms

$$\eta^l : \mathrm{I} \to A \otimes A^l \qquad \epsilon^l : A^l \otimes A \to \mathrm{I}$$

$$\eta^r : \mathrm{I} \to A^r \otimes A \qquad \epsilon^r : A \otimes A^r \to \mathrm{I}$$

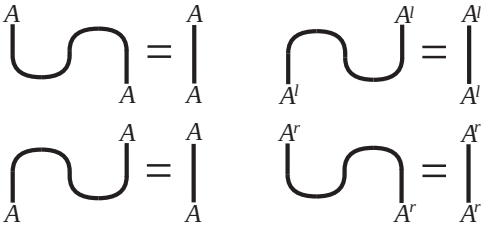
which satisfy:[5]

$$(1_A \otimes \epsilon^l) \circ (\eta^l \otimes 1_A) = 1_A \quad (\epsilon^l \otimes 1_{A^l}) \circ (1_{A^l} \otimes \eta^l) = 1_{A^l}$$

$$(\epsilon^r \otimes 1_A) \circ (1_A \otimes \eta^r) = 1_A \quad (1_{A^r} \otimes \epsilon^r) \circ (\eta^r \otimes 1_{A^r}) = 1_{A^r}$$

When depicting the morphisms $\eta^l, \epsilon^l, \eta^r, \epsilon^r$ as

these axioms substantially simplify to



i.e. they boil down to 'yanking wires'.

[5] These conditions guarantee that the category is closed, as explained in a simple diagrammatical manner in (Coecke 2007).

3. Vector spaces, linear maps and tensor product. Let $\mathbf{FVect}$ be the category which has vector spaces over the base field $\mathbb{R}$ as objects, linear maps as morphisms and the vector space tensor product as the monoidal tensor. To simplify the presentation we assume that each vector space comes with an inner product, that is, it is an *inner-product space*. For the case of vector space models of meaning this is always the case, since we consider a fixed base, and a fixed base canonically induces an inner-product. The reader can verify that compact closure arises, given a vector space V with base $\{e_i\}_i$, by setting $V^l = V^r = V$,

$$\eta^l = \eta^r : \mathbb{R} \to V \otimes V :: 1 \mapsto \sum_i e_i \otimes e_i$$

and

$$\epsilon^l = \epsilon^r : V \otimes V \to \mathbb{R} :: \sum_{ij} c_{ij}\, \psi_i \otimes \phi_j \mapsto \sum_{ij} c_{ij} \langle \psi_i | \phi_j \rangle \,.$$

So $\epsilon^l = \epsilon^r$ is the inner-product extended by linearity to the whole tensor product. Recall that if $\{e_i\}_i$ is a base for V and if $\{e'_i\}_i$ is a base for W then $\{e_i \otimes e'_j\}_{ij}$ is a base for $V \otimes W$. In the base $\{e_i \otimes e_j\}_{ij}$ for $V \otimes V$ the linear map $\epsilon^l = \epsilon^r : V \otimes V \to \mathbb{R}$ has as its matrix the row vector which has entry 1 for the base vectors $e_i \otimes e_i$ and which has entry 0 for the base vectors $e_i \otimes e_j$ with $i \neq j$. The matrix of $\eta^l = \eta^r$ is the column vector obained by transposition.

4. Pregroups as compact closed categories. A pregroup is an example of a *posetal category*, that is, a category which is also a poset. For a category this means that for any two objects there is either one or no morphism between them. In the case that this morphism is of type $A \to B$ then we write $A \leq B$, and in the case it is of type $B \to A$ we write $B \leq A$. The reader can then verify that the axioms of a category guarantee that the relation $\leq$ on $|C|$ is indeed a partial order. Conversely, any partially ordered set $(P, \leq)$ is a category. For 'objects' $a, b \in P$ we take $P(a, b)$ to be the singleton $\{a \leq b\}$ whenever $a \leq b$, and empty otherwise. If $a \leq b$ and $b \leq c$ we define $a \leq c$ to be the composite of the 'morphisms' $a \leq b$ and $b \leq c$.

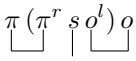
A partially ordered monoid is a monoidal category with the monoid multiplication as tensor on objects; whenever $a \leq c$ and $b \leq d$ then we have $a \cdot b \leq c \cdot d$ by equation (1), and we define this to be the tensor of 'morphisms' $a \leq c$ and $b \leq d$. Bifunctoriality, as well as any equational statement between morphisms in posetal categories, is trivially satisfied, since there can only be one morphism between any two objects.

Finally, each pregroup is a compact closed category for

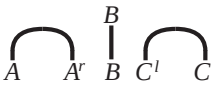
$$\eta^l = [1 \leq p \cdot p^l] \qquad \epsilon^l = [p^l \cdot p \leq 1]$$

$$\eta^r = [1 \leq p^r \cdot p] \qquad \epsilon^r = [p \cdot p^r \leq 1]$$

and so the required equations are again trivially satisfied. Diagrammatically, the links representing the type reductions in



are exactly the 'caps':

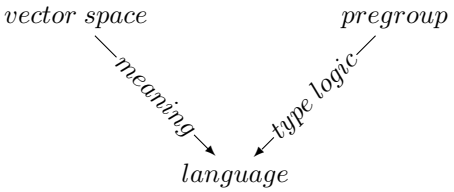


of the compact closed structure. The symbolic counterpart of this diagrammatically depicted morphism is:

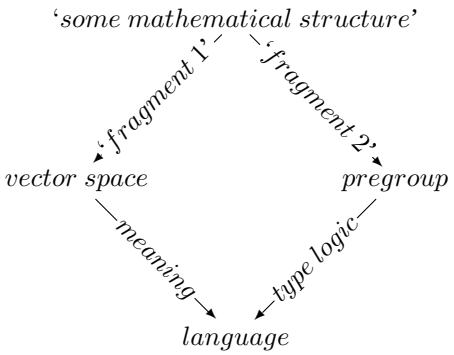
$$\epsilon^r_A \otimes 1_B \otimes \epsilon^l_C : A \otimes A^r \otimes B \otimes C^l \otimes C \to B\,.$$

Composing meanings = quantifying type logic

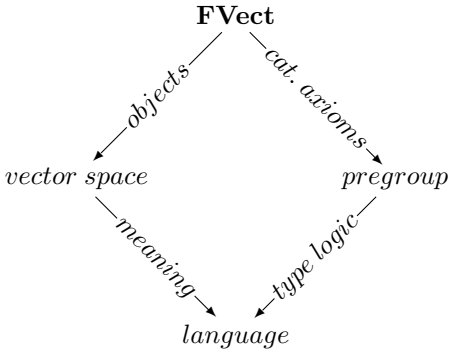
We have described two approaches to analysing structure and meaning in natural language, one in which vector spaces are used to assign meanings to words in a language, and another in which pregroups are used to assign grammatical structure to sentences:



We aim for a theory that unifies both approaches, in which the compositional structure of pregroups would lift to the level of assigning meaning to sentences and their constituents. Our approach proceeds as follows: we look for a mathematical structure which comprises both the compositional pregroup structure and the vector space structure as fragments.



Our previous mathematical analysis suggests the following:



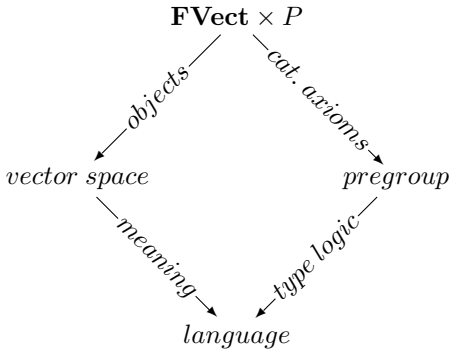
This suggestion is problematic, however. The compact closed structure of **FVect** is somewhat degenerate since $A^l = A^r = A$. Moreover, there are canonical isomorphisms $V \otimes W \to W \otimes V$ which translate to posetal categories as $a \cdot b = b \cdot a$. Therefore we have to refine types to retain the full grammatical content obtained from the pregroup analysis. There is an easy way of doing this: rather than objects in **FVect** we will consider objects in the product category $\mathbf{FVect} \times P$, where P is the pregroup generated from basic types. Explicitly, **FVect** is the category which has pairs (V, a) with V a vector space and $a \in P$ as objects, and the following pairs as morphisms:

$$(f : V \to W\,,\; p \leq q)$$

which we can also write as

$$(f, \leq) : (V, p) \to (W, q).$$

Note that if $p \not\leq q$ then there are no morphisms of type $(V, p) \to (W, q)$. It is easy to verify that the compact closed structure of **FVect** and P lifts component-wise to one on $\mathbf{FVect} \times P$:



We can now also lift the mechanism for establishing well-typedness of sentences within pregroups to morphisms in **FVect**: given a reduction $p \cdots q \to s$, there is a morphism

$$(f, \leq) : (V, p) \otimes \ldots \otimes (W, q) \to (S, s)$$

which assigns to each vector $\overrightarrow{v} \otimes \ldots \otimes \overrightarrow{w} \in V \otimes \ldots \otimes W$ (the meaning of a sentence of type $p \cdots q$) a vector:

$$f(\overrightarrow{v} \otimes \ldots \otimes \overrightarrow{w}) \in S\,,$$

where S is the meaning space of all sentences. Note that this resulting vector $f(\overrightarrow{v} \otimes \ldots \otimes \overrightarrow{w})$ does not depend on the grammatical type $p \cdots q$.

Computing the meaning of a sentence

We define a meaning space to be a pair consisting of a vector space V and a grammatical type p, where, following the vector space model of meaning, the vectors in V encode the meaning of words of type p.[6] To assign meaning to a sentence of type $\pi(\pi^r s o^l)o$, we take the tensor product of the meaning spaces of its constituents, that is:

$$(V, \pi) \otimes (T, \pi^r s o^l) \otimes (W, o) \;=\; \big(V \otimes T \otimes W, \pi(\pi^r s o^l)o\big)\,.$$

[6]The pair $(\overrightarrow{v}, p)$, where $\overrightarrow{v}$ is a vector which represents the meaning of a word and p is the pregroup element which represents its type, is our counterpart of the pure tensor $\overrightarrow{v} \otimes \overrightarrow{p}$ in the proposal of (Clark & Pulman 2007), in which $\overrightarrow{p}$ is a genuine vector. The structure proposed here can easily be adapted to also allow types to be represented in a vector space.

From the type $\pi^r so^l$ of the transitive verb, we know that the vector space in which it is described is of the form:

$$T = V \otimes S \otimes W\,. \tag{2}$$

The linear map f which realizes

$$\left(V \otimes T \otimes W\,, o(\pi^r so^l)o\right) \xrightarrow{(f,\leq)} (S, s)\,,$$

and arises from the type-reductions, is in this case:

$$f = \epsilon^r_V \otimes 1_S \otimes \epsilon^l_W : V \otimes T \otimes W \to S\,.$$

Diagrammatically, the linear map can be represented as follows:

The matrix of f has $dim(V)^2 \times dim(S) \times dim(W)^2$ columns and $dim(S)$ rows, and its entries are either 0 or 1. We can also express $f(\overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w}) \in S$ for $\overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w} \in V \otimes S \otimes W$ in terms of the inner-product. If

$$\Psi = \sum_{ijk} c_{ijk}\, \overrightarrow{v}_i \otimes \overrightarrow{s}_j \otimes \overrightarrow{w}_k \in V \otimes S \otimes W$$

then

$$\begin{aligned} f(\overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w}) &= \sum_{ijk} c_{ijk} \langle \overrightarrow{v} | \overrightarrow{v}_i \rangle \overrightarrow{s}_j \langle \overrightarrow{w}_k | \overrightarrow{w} \rangle \\ &= \sum_j \left(\sum_{ik} c_{ik} \langle \overrightarrow{v} | \overrightarrow{v}_i \rangle \langle \overrightarrow{w}_k | \overrightarrow{w} \rangle \right) \overrightarrow{s}_j\,. \end{aligned}$$

This vector is the meaning of the sentence of type $\pi(\pi^r so^l)o$, and assumes as given the meanings of its constituents $\overrightarrow{v} \in V$, $\overrightarrow{\Psi} \in T$ and $\overrightarrow{w} \in W$, obtained from data using some suitable method. In Dirac notation, $f(\overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w})$ would be written as:

$$\left(\langle \epsilon^r_V | \otimes 1_S \otimes \langle \epsilon^r_V | \right) \left| \overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w} \right\rangle.$$

As mentioned in the introduction, our focus in this paper is not on how to practically exploit the mathematical framework, which would require substantial further research, but to expose the mechanisms which govern it. To show that this particular computation does indeed produce a vector which captures the meaning of a sentence, we explicitly compute $f(\overrightarrow{v} \otimes \overrightarrow{\Psi} \otimes \overrightarrow{w})$ for some simple examples, with the intention of providing the reader with some insight into the underlying mechanisms and how the approach relates to existing frameworks.

Example 1. Consider the sentence

John likes Mary. (3)

We encode this sentence as follows; we have:

$$\overrightarrow{\textit{John}} \in V, \quad \overrightarrow{\textit{likes}} \in T, \quad \overrightarrow{\textit{Mary}} \in W$$

where we take V to be the vector space spanned by men and W the vector space spanned by women.[7] We will conveniently assume that all men are named *John*, using indices to distinguish them: $\textit{John}_i$. Similarly every woman will be referred to as $\textit{Mary}_j$, for some j, and the set of vectors $\{\overrightarrow{\textit{Mary}_j}\}_j$ spans W. Let us assume that *John* in (3) is $\textit{John}_3$ and that *Mary* is $\textit{Mary}_4$. Assume also that we are only interested in the truth or falsity of a sentence. Therefore we take the sentence space S to be spanned by a single vector $\overrightarrow{1}$, which we identify with *true*, and we identify the origin $\overrightarrow{0}$ with *false*. The transitive verb $\overrightarrow{\textit{likes}}$ is encoded as the *superposition*:

$$\overrightarrow{\textit{likes}} = \sum_{ij} \overrightarrow{\textit{John}_i} \otimes \overrightarrow{\textit{likes}_{ij}} \otimes \overrightarrow{\textit{Mary}_j}$$

where $\overrightarrow{\textit{likes}_{ij}} = \overrightarrow{1}$ if $\textit{John}_i$ likes $\textit{Mary}_j$ and $\overrightarrow{\textit{likes}_{ij}} = \overrightarrow{0}$ otherwise. Of course, in practice, the vector that we have constructed here would be obtained automatically from data using some suitable method. Finally, we obtain:

$$\begin{aligned} & f\left(\overrightarrow{\textit{John}_3} \otimes \overrightarrow{\textit{likes}} \otimes \overrightarrow{\textit{Mary}_4}\right) \\ &= \sum_{ij} \left\langle \overrightarrow{\textit{John}_3} \mid \overrightarrow{\textit{John}_i} \right\rangle \otimes \overrightarrow{\textit{likes}_{ij}} \otimes \left\langle \overrightarrow{\textit{Mary}_j} \mid \overrightarrow{\textit{Mary}_4} \right\rangle \\ &= \sum_{ij} \delta_{3i}\, \overrightarrow{\textit{likes}_{ij}}\, \delta_{j4} \\ &= \overrightarrow{\textit{likes}_{34}} \end{aligned}$$

So we indeed obtain the correct meaning of our sentence. Note in particular that the transitive verb acts as a function requiring an argument of type subject on its left and another argument of type object on its right, in order to output an argument of type sentence.

Example 2. We alter the above example as follows. Let S be spanned by two vectors, $\overrightarrow{l}$ and $\overrightarrow{h}$. Set

$$\overrightarrow{\textit{loves}} = \sum_{ij} \overrightarrow{\textit{John}_i} \otimes \overrightarrow{\textit{loves}_{ij}} \otimes \overrightarrow{\textit{Mary}_j}$$

$$\overrightarrow{\textit{hates}} = \sum_{ij} \overrightarrow{\textit{John}_i} \otimes \overrightarrow{\textit{hates}_{ij}} \otimes \overrightarrow{\textit{Mary}_j}$$

where $\overrightarrow{\textit{loves}_{ij}} = \overrightarrow{l}$ if $\textit{John}_i$ loves $\textit{Mary}_j$ and $\overrightarrow{\textit{loves}_{ij}} = \overrightarrow{0}$ otherwise, and $\overrightarrow{\textit{hates}_{ij}} = \overrightarrow{h}$ if $\textit{John}_i$ hates $\textit{Mary}_j$ and $\overrightarrow{\textit{hates}_{ij}} = \overrightarrow{0}$ otherwise. Set

$$\overrightarrow{\textit{likes}} = \frac{3}{4}\overrightarrow{\textit{loves}} + \frac{1}{4}\overrightarrow{\textit{hates}}\,.$$

The reader can verify that we obtain

$$\left\langle f(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{loves}} \otimes \overrightarrow{M}_4) \;\middle|\; f(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{likes}} \otimes \overrightarrow{M}_4) \right\rangle = \frac{3}{4}$$

[7] In terms of context vectors this means that each word is its own and only context vector, which is of course a far too simple idealisation for practical purposes.

$$\left\langle f\big(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{likes}} \otimes \overrightarrow{M}_4\big) \;\middle|\; f\big(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{hates}} \otimes \overrightarrow{M}_4\big)\right\rangle = \frac{1}{4}$$

$$\left\langle f\big(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{loves}} \otimes \overrightarrow{M}_4\big) \;\middle|\; f\big(\overrightarrow{J}_3 \otimes \overrightarrow{\textit{hates}} \otimes \overrightarrow{M}_4\big)\right\rangle = 0\,.$$

Hence the meaning of the distinct verbs $loves$, $likes$ and $hates$ in the different sentences propagates through the reduction mechanism and reveals itself when computing inner-products between sentences in the sentence space.

Due to lack of space we leave more involved examples to future papers, including those which involve comparing the meanings of sentences of different types. Of course in a full-blown vector space model, which has been automatically extracted from large amounts of text, we obtain 'imperfect' vector representations for words, rather than the 'ideal' ones presented here. But the mechanism of how the meanings of words propagate to the meanings of sentences remains the same.

Change of model while retaining the logic

When fixing a base for each vector space we can think of $\mathbf{FVect}$ as a category of which the morphisms are matrices expressed in this base. These matrices have real numbers as entries. It turns out that if we consider matrices with entries not in $(\mathbb{R}, +, \times)$, but in any other semiring[8] $(R, +, \times)$, we again obtain a compact closed category. This semiring does not have to be a field, and can for example be the positive reals $(\mathbb{R}^+, +, \times)$, positive integers $(\mathbb{N}, +, \times)$ or even Booleans $(\mathbb{B}, \vee, \wedge)$.

In the case of $(\mathbb{B}, \vee, \wedge)$, we obtain an isomorphic copy of the category $\mathbf{FRel}$ of finite sets and relations with the cartesian product as tensor, as follows. Let X be a set whose elements we have enumerated as $X = \{x_i \mid 1 \leq i \leq |X|\}$. Each element can be seen as a column with a 1 at the row equal to its number and 0 in all other rows. Let $Y = \{y_j \mid 1 \leq j \leq |Y|\}$ be another enumerated set. A relation $r \subseteq X \times Y$ is represented by an $|X| \times |Y|$ matrix, where the entry in the ith column and jth row is 1 iff $(x_i, y_j) \in r$ and 0 otherwise. The composite $s \circ r$ of relations $r \subseteq X \times Y$ and $s \subseteq Y \times Z$ is

$$\{(x, z) \mid \exists y \in Y : (x, y) \in r, (y, z) \in s\}\,.$$

The reader can verify that this composition induces matrix multiplication of the corresponding matrices.

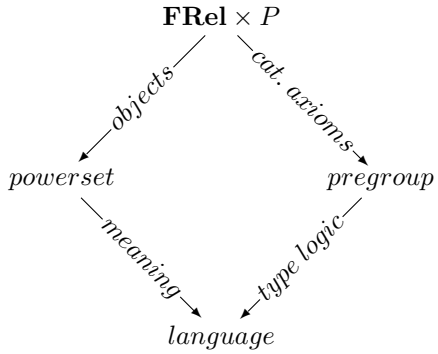
Interestingly, in the world of relations (but not functions) there is a notion of *superposition* (Coecke 2006). The relations of type $r \subseteq \{*\} \times X$ (in matricial terms, all column vectors with 0's and 1's as entries) are in bijective correspondence with the subsets of X via the correspondence

$$r \mapsto \{x \in X \mid (*, x) \in r\}\,.$$

Each such subset can be seen as the superposition of the elements it contains. The *inner-product* of two subsets is 0 if they are disjoint and 1 if they have a non-empty intersection. So we can think of two disjoint sets as being *orthogonal*.

[8] A semiring is a set together with two operations, addition and multiplication, for which we have a distributive law but no additive nor multiplicative inverses. Having an addition and multiplication of this kind suffices to have a matrix calculus.

Since the abstract nature of our procedure for assigning meaning to sentences did not depend on the particular choice of $\mathbf{FVect}$ we can now repeat it for the following situation:



Montague-style semantics

In $\mathbf{FRel} \times P$ we recover Montague semantics. Example 1 above was chosen in such a manner that it is essentially *relational*. The singleton $\{*\}$ has two subsets, namely $\{*\}$ and $\emptyset$, which we respectively identify with *true* and *false*. We now have sets V, W and $T = V \times \{*\} \times W$ with

$$V := \{\textit{John}_i\}_i\,, \quad \textit{likes} \subset T, \quad W := \{\textit{Mary}_j\}_j$$

such that:

$$\begin{aligned}\textit{likes} &:= \{(\textit{John}_i, *, \textit{Mary}_j) \mid \textit{John}_i \textit{ likes } \textit{Mary}_j\} \\ &= \bigcup_{ij} \{\textit{John}_i\} \times *_{ij} \times \{\textit{Mary}_j\}\end{aligned}$$

where $*_{ij}$ is either $\{*\}$ or $\emptyset$. Finally we have

$$\begin{aligned}&f\,(\{\textit{John}_3\} \times \textit{likes} \times \{\textit{Mary}_4\}) \\ &= \bigcup_{ij} (\{\textit{John}_3\} \cap \{\textit{John}_i\}) \times *_{ij} \times \big(\{\textit{Mary}_j\} \cap \{\textit{Mary}_4\}\big) \\ &= *_{34}\,.\end{aligned}$$

Future Work

There are several implementation issues which need to be investigated. It would be good to have a Curry-Howard like isomorphism between non-commutative compact closed categories, bicompact linear logic (Buszkowski 2001), and Abramsky's planar lambda calculus. This will enable us to automatically obtain computations for the meaning and type assignments of our categorical setting. Also, Montague-style semantics lives in $\mathbf{FRel} \times P$ whereas truly distributed semantics lives in $\mathbf{FVect} \times P$. It would be interesting to see where the so called 'non-logical' axioms of Montague live and how they are manifested at the level of $\mathbf{FVect} \times P$. Categorical axiomatics is flexible enough to accommodate *mixed states* (Selinger 2007), so in principle we can implement the proposals of (Bruza & Widdows 2007).

Acknowledgements

Bob Coecke is supported by EPSRC Advanced Research Fellowship EP/D072786/1 *The Structure of Quantum Information and its Ramifications for IT* and by EC-FP6-STREP 033763 *Foundational structures for quantum information and computation*. We thank Keith Van Rijsbergen and Stephen Pulman for their feedback at QI'07 in Stanford.

References

Abramsky, S., and Coecke, B. 2004. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, 415–425. IEEE Computer Science Press. arXiv:quant-ph/0402130.

Abramsky, S., and Duncan, R. W. 2006. A categorical quantum logic. *Mathematical Structures in Computer Science* 16:469 – 489.

Bargelli, D., and Lambek, J. 2001a. An algebraic approach to Arabic sentence structure. *Linguistic Analysis* 31.

Bargelli, D., and Lambek, J. 2001b. An algebraic approach to French sentence structure. *Logical Aspects of Computational Linguistics*.

Bruza, P., and Widdows, D. 2007. Quantum information dynamics and open world science. AAAI Press. Proceedings of AAAI Spring Symposium on Quantum Interaction.

Buszkowski, W. 2001. Lambek grammars based on pregroups. *Logical Aspects of Computational Linguistics*.

Casadio, C., and Lambek, J. 2001. An algebraic analysis of clitic pronouns in Italian. *Logical Aspects of Computational Linguistics*.

Clark, S., and Pulman, S. 2007. Combining symbolic and distributional models of meaning. AAAI Press. Proceedings of AAAI Spring Symposium on Quantum Interaction.

Coecke, B. 2005a. Kindergarten quantum mechanics — lecture notes. In Khrennikov, A., ed., *Quantum Theory: Reconsiderations of the Foundations III*, 81–98. AIP Press. arXiv:quant-ph/0510032.

Coecke, B. 2005b. Quantum information-flow, concretely, and axiomatically. In *Proceedings of Quantum Informatics 2004, Proceedings of SPIE 5833*, 35–113. SPIE Publishing. arXiv:quant-ph/0506132.

Coecke, B. 2006. Introducing categories to the practicing physicist. In Sica, G., ed., *What is category theory?*, volume 30 of *Advanced Studies in Mathematics and Logic*. Polimetrica Publishing. 45–74. http://web.comlab.ox.ac.uk/oucl/work/bob.coecke/Cats.pdf.

Coecke, B. 2007. Automated quantum reasoning: Non-logic $\rightsquigarrow$ semi-logic $\rightsquigarrow$ hyper-logic. AAAI Press. Proceedings of AAAI Spring Symposium on Quantum Interaction.

Curran, J. R. 2004. *From Distributional to Semantic Similarity*. Ph.D. Dissertation, University of Edinburgh.

Dowty, D.; Wall, R.; and Peters, S. 1981. *Introduction to Montague Semantics*. Dordrecht.

Freyd, P., and Yetter, D. 1989. Braided compact closed categories with applications to low-dimensional topology. *Advances in Mathematics* 77:156–182.

Gazdar, G. 1996. Paradigm merger in natural language processing. In Milner, R., and Wand, I., eds., *Computing Tomorrow: Future Research Directions in Computer Science*. Cambridge University Press. 88–109.

Lambek, J. 1958. The mathematics of sentence structure. *American Mathematics Monthly* 65.

Lambek, J. 1999. Type grammar revisited. *Logical Aspects of Computational Linguistics* 1582.

Lambek, J. 2004. A computational algebraic approach to English grammar. *Syntax* 7:2.

Moortgat, M. 1997. Categorical type logics. In van Benthem, J., and ter Meulen, A., eds., *Handbook of Logic and Language*. Elsevier.

Preller, A. 2007. Towards discourse representation via pregroup grammars. *JoLLI*.

Sadrzadeh, M. 2006. Pregroup analysis of Persian sentences. In Casadio, C., and Lambek, J., eds., *Recent computational algebraic approaches to morphology and syntax (to appear)*. http://www.ecs.soton.ac.uk/∼ms6/PersPreGroup.pdf.

Sadrzadeh, M. 2007. High-level quantum structures in linguistics and multi-agent systems. AAAI Press. Proceedings of AAAI Spring Symposium on Quantum Interaction.

Schuetze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.

Selinger, P. 2007. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science* 170:139–163.

Smolensky, P., and Legendre, G. 2005. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar Vol. I: Cognitive Architecture Vol. II: Linguistic and Philosophical Implications*. MIT Press.