The Frobenius anatomy of word meanings II: possessive relative pronouns

Sadrzadeh, M; Clark, S; Coecke, B

The Frobenius anatomy of word meanings II: possessive relative pronouns

Mehrnoosh Sadrzadeh
Queen Mary University of London
School of Electronic Eng. and Computer Science
mehrs@eecs.qmul.ac.uk

Stephen Clark
University of Cambridge
Computer Laboratory
stephen.clark@cl.cam.ac.uk

Bob Coecke
University of Oxford
Dept. of Computer Science
coecke@cs.ox.ac.uk

Abstract. Within the categorical compositional distributional model of meaning, we provide semantic interpretations for the subject and object roles of the possessive relative pronoun 'whose'. This is done in terms of Frobenius algebras over compact closed categories. These algebras and their diagrammatic language expose how meanings of words in relative clauses interact with each other. We show how our interpretation is related to Montague-style semantics and provide a truth-theoretic interpretation. We also show how vector spaces provide a concrete interpretation and provide preliminary corpus-based experimental evidence. In a prequel to this paper, we used similar methods and dealt with the case of subject and object relative pronouns.

Dedicated to Roy Dyckhoff on the occasion of his 63rd birthday.

1 Introduction

Mathematical linguistics is a field of Computational Linguistics that formalises and reasons about properties of natural language. Through the seminal work of Lambek [17], certain formal models of natural language have found connections to algebraic and categorical models of programming languages. In a nutshell, these more abstract models of both fields use the algebraic structure of residuated monoids. In the context of natural language, grammatical types are elements of the monoid, their juxtapositions are modelled by the monoid multiplications and grammatical reductions are modelled by the partial orders of the algebra. Relational or functional words, such as verbs, have implicative types and the residuals to the monoid multiplication are used to model them. Later, Lambek simplified these algebras and developed a structure he named a *pregroup*, wherein, there are no binary residual operations, but each element of the algebra has a left and a right residual [18]. Pregroups have been applied to formalising and reasoning about grammatical structures of different families of natural language, for recent examples see [20].

Vector spaces have been applied to formalise meanings of words in natural language, leading to models referred to as *distributional* [26]. In these models, the semantics of a word is a vector with coordinates based on the frequency of the co-occurrence of that word in the context of other words [8]. For instance, the word 'queen' often appears in the context of 'reign' and 'rule', so its meaning can be guessed from the meanings of 'reign' and 'rule', whereas the word 'carnivore' has a different meaning since it appears in the context of, for instance, 'animal' and 'meat'. This model has been successfully applied to natural language processing tasks, such as building automatic thesauri [7].

Vector spaces with linear maps and pregroups both have a compact closed categorical structure [22,16]. Based on this common structure, in previous work we provided an interpretation of pregroups in vector spaces and developed a vector space semantics for them [6,5]. This semantics extends the distributional models from words to sentences. It provides an abstract setting where one constructs meaning vectors for strings of words compositionally, based on their grammatical structure and the distributional vectors of the words within them. The theoretical predictions of a fragment of this setting were implemented and evaluated on language tasks, such as disambiguation, phrase similarity, and term/definition classification [10,11,14,13].

No matter how successful they have been in certain natural language processing tasks, distributional models cannot be used to build vectors for words whose meanings do not depend on the context. Among these words are logical words such as 'and, or',articles, such as 'a', 'the', and relative pronouns such as 'who, that, whose'. Our focus in this and a prequel paper [24] has been on relative pronouns. For instance, the word 'queen' can be described by the clause 'a woman who rules a country', and the word 'clown' by the clause 'a man who performs funny tricks'; making the relative pronoun 'who' appear in very different contexts. Because of this context-independent property, relative pronouns are often treated as 'noise' by distributional models. As a result, they are not taken into account when building vector representations for the clauses containing them. Even if they were (and as we will show in the last section of the paper) their vectors would be so dense that operating with them would be similar to operating with a vector consisting of 1's. Either of these solutions discards the vital information that is encoded in the structural semantics of a relative pronoun; the information that tells us how different parts of the clause are related to each other and which helps us derive the meaning of the clause and hence that of the word it is describing.

In a prequel to this paper [24], we developed a compositional distributional semantics for the subject and object relative pronouns 'who, which, that, whom'. Here we provide a semantics for the possessive relative pronoun 'whose', in its object and subject roles. This semantics does not depend on the context but rather on the structural roles of the pronouns. We use the general operations of a Frobenius algebra over a vector space [4] to model these structural roles. The computations of the algebra and vector space are depicted using a diagrammatic language [12,27] that depicts the interactions that happen among the words of a clause and produce the meaning of the overall compound. The diagrams visualise the role of the relative pronoun in passing the information of the head of the clause to the rest of the clause, acting on this information, copying, unifying, and even discarding it. Using these diagrams we show how possessive relative clauses can be decomposed to clauses with subject and object relative pronouns that contain a possessive predicate such as 'has'.

Further, we instantiate our mathematical constructions in a truth theoretic setting and show how the vector constructions also provide us with Montague-stye set theoretic semantics [23]. We also instantiate our model on vector spaces built from a corpus of real data and develop linear algebraic forms of the categorical morphisms. Finally, we provide a preliminary experiment by developing a toy dataset of words and their relative-clause descriptions and show how the cosine of the angle between the vectors of the words and the vectors of their descriptions can be used to assign the correct description to the word.

2 Related Work

As opposed to simple models such as that of [21], a compositional distributional model that does not ignore relative pronouns is that of [2]. In this line of work, the meaning vectors of phrases and sentences are computed based on their syntactic structure. Each word within a phrase or sentence has a grammatical type and based on this type a matrix or tensor is assigned to it, representing its meaning; the composition operator is a matrix multiplication, or its generalisation: tensor contraction. For instance, the meaning of an adjective noun phrase is computed by multiplying the matrix of an adjective with the vector of a noun. The meaning of a simple transitive sentence is computed by contracting the tensor of the verb with the vectors of object and subject. To some extent, this approach is the same as ours: words that have function types live in tensor spaces and the composition operator is composition of the linear maps corresponding to these tensors.

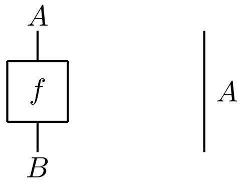
There are two differences. Firstly, this model estimates the tensors of words by doing regression on the co-occurrence vectors of their contexts. For instance, the matrix of 'red' is estimated from the adjective noun phrases such as 'red car', 'red carpet', 'red wine', etc. We, on the other hand, do not bind our method to any concrete construction and the concrete constructions that we have been using are very different from the above. However, the linear-regression constructions can also be embedded in our setting and provide the same results. There is a second more important difference, which shows itself in the developments of the present (and its prequel) paper on relative pronouns. The types that the models of [3,2] consider are purely syntactic and do not contain semantic information. The meaning of the relative pronoun 'which' is obtained by taking the intersection of the meaning of its head with the meaning of the rest of the clause. But in the absence of an intersection operator in vector spaces, the authors move to a simpler approach, where the meaning of 'which' is a function that inputs a verb phrase, e.g. 'eats meat' and outputs a modifies noun phrase, e.g. 'which eats meat'. This bypass is not necessary in our model. As a result, we do not need to build many-dimensional tensors for the relative pronouns and the two defects of data sparsity and computational power, mentioned in [2], are automatically overcome. These defects arise since, for example, the tensor of 'which' will have four dimensions; even in a vector space model where the dimensions are reduced this will cause a problem. For instance in a vector space with 300 dimensions, the tensor of 'whose' will have $300^4 = 81 \times 10^8$ (8.1. billion) dimensions. Compare this to our setting, where we do not need to build any concrete tensor for relative pronouns: the Frobenius operations allow us to encode their syntactic and semantic roles using simple operations on the meanings of the other words of the clause. Finally, the approach of [2] only treats the relative pronoun 'which', here we also deal with the possessive pronoun 'whose'.

3 Compact Closed Categories, Diagrams, Examples

In order to ground the constructions that will provide us with meanings of relative pronouns, we need to discuss the theory of categories and in particular the definition and operations of compact closed categories. Theory of categories is a mathematical theory that abstracts away from the concrete details of structures and relates them to each other via the high-level properties that they hold. It was this theoretical tool that enabled us to relate the grammatical structures of sentences to vector semantics in a compositional way. Frobenius algebras are operators that can be applied to certain objects within compact closed categories. These will provide our setting with extra expressive powers and will help us embed the features that are required to model relative pronouns. In a prequel to this paper [24], we worked with purely formal definitions. In this paper, we explain these in an informal way. We also use a diagrammatic calculus to depict them; these should help the reader follow the explanations easier. The diagrams will also help us depict the computations necessary for providing meaning for the role of relative pronouns within linguistic compounds.

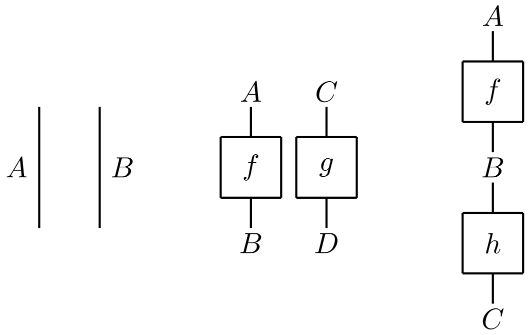
The theory of categories is the study of abstract mathematical structures referred to by *categories*. Categories have *objects* and *morphisms*. Examples of objects are sets, elements of a set, groups or elements of a group. In the context of linguistics, they are usually taken to be grammatical types of words. Morphisms map objects to each other, they might denote a way of relating objects to each other or transforming an object to another. For instance, if the objects of a category are sets, the morphisms can be functions or relations between the sets. In the context of linguistics, they denote the grammatical reductions between the types.

If we denote the objects of a category by $A, B, C, ...$, the morphisms will be denoted by $f\colon A \to B$, $g\colon B \to C$. The morphisms can compose with each other, that is whenever we have morphisms f and g defined as above, we also have a morphisms $h\colon A \to C$, and h is a composition of f and g, denoted by $g \circ f$. There is also a special morphism called *identity* that transforms an object to itself; it is denoted by $1_A\colon A \to A$. In the context of sets, this can be the morphism that maps the elements of a set to other elements of the same set. Diagrammatically, the objects and the identity morphisms are depicted by lines. All other morphisms are depicted by boxes. For instance a morphism $f\colon A \to B$ and an object A and its identity arrow $1_A\colon A \to A$ are depicted as follows:



One can define operations on the objects. For example, if objects are sets, one can take their intersection, union, or Cartesian product. In the context of linguistics, one needs to juxtapose the grammatical types to obtain the type of a juxtaposition of words. The abstract form of the juxtaposition operation is called a monoidal tensor and is denoted by $A \otimes B$. So if A is the grammatical type of the word w_1, e.g. 'red' and B is the grammatical type of the word w_2, e.g. 'car', then $A \otimes B$ is the type of the string of words $w_1 w_2$, that is 'red car'. Such operations usually have a unit, for the case of union, the unit is the empty set, denoted by $\emptyset$, since we have $A \cup \emptyset = A$. In the case of juxtaposition, the unit is the empty type, denoted by 1, since we have that the juxtaposition of a type with an empty type is the original type, that is $A \otimes I = A$. A category that has a monoidal tensor with a unit (and which satisfies certain other equations, which we will not present here), is called a *monoidal closed category.*

Diagrammatically, the tensor products of the objects and morphisms are depicted by juxtaposing their diagrams side by side, whereas compositions of morphisms are depicted by putting one on top of the other, for instance the object $A \otimes B$, and the morphisms $f \otimes g$ and $f \circ h$, for $f \colon A \to B, g \colon C \to D$, and $h \colon B \to C$ are depicted as follows:



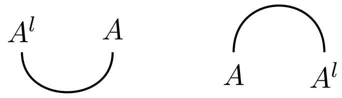
A category is called *compact closed*, whenever it is monoidal closed and moreover, its objects can cancel each other out and generate each other. Cancelation means that there is way of transforming the tensor of certain objects to the unit of the tensor. Generation means that there is a way of transforming the unit to the tensor of certain objects. To make this property formal, we assign to each object A, an object denoted by A^r, referred to by the right adjoint of A, and another object denoted by A^l, referred to by the left adjoint of A. The morphism that transforms $A \otimes A^r$ to I is referred to by ϵ^r_A and the morphism that transforms $A^l \otimes A$ to I is referred to by ϵ^l_A. These are denoted as follows:

$$\textbf{Cancelations :} \qquad A \otimes A^r \xrightarrow{\epsilon^r_A} I \qquad A^l \otimes A \xrightarrow{\epsilon^l_A} I$$

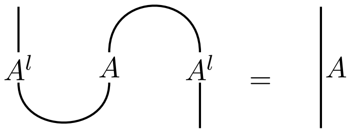
The morphism that transforms I to $A^r \otimes A$ is referred to by η^r_A and the morphism that transforms I to $A^l \otimes A$ is referred to by η^l_A. These are denoted as follows:

$$\textbf{Generations}: \qquad I \xrightarrow{\eta^r_A} A^r \otimes A \qquad I \xrightarrow{\eta^l_A} A \otimes A^l$$

The ϵ maps are depicted by cups, and the η maps by caps. For instance, the diagrams for $\epsilon^l \colon A^l \otimes A \to I$ and $\eta \colon I \to A \otimes A^l$ are as follows:



The morphisms of a compact closed category satisfy certain formal equations. The most important one consists of four equations referred to by *yanking*. One of the instances of this property is $(\epsilon^l \otimes 1_A) \circ (1_A \otimes \eta^l) = 1_A$, depicted below:



Roughly speaking, yanking expresses the fact that a cancelation followed by a generation (or the other way around) is the same as doing nothing, that is, the same as having an identity morphism.

3.1 A compact closed category for grammar

Theory of categories was first applied to the analysis of grammatical structure of language in [17]. The categories presented and studied there were monoidal closed. The first application of compact closed categories to grammar was in [19]. The argument behind the passage from monoidal to compact was mainly simplicity. Monoidal closed categories have two other operations (other than the tensor) and all of these three operations are needed when analysing grammar. In compact closed categories, the applications of these two other operations are modelled by the cancelation and generation morphisms.

The compact closed category of grammar is called a *pregroup*. The objects of this category are grammatical types. The morphisms of this category are grammatical reductions. The objects are denoted by $p, q, r, \cdots$, and the morphisms are denoted by partial orders such as $p \leq q$. The partial order morphisms compose with each other as follows: whenever we have $p \leq q$ and $q \leq r$, we also have that $p \leq r$. This is because partial ordering is a transitive relation. The tensor product of the category denotes the juxtaposition of types. So $p \otimes q$ denotes the juxtaposition of type p with type q. This can be the type of a two word phrase '$w_1 w_2$', where w_1 has type p and w_2 has type q.

Since the category is compact, each type p has a right adjoint p^r and a left adjoint p^l. This means that we have the following cancelation and generation morphisms:

$$\begin{array}{lll}
\textbf{Cancelation} & \epsilon_p^r\colon\ p \otimes p^r \leq 1 & \epsilon_p^l\colon\ p^l \otimes p \leq 1 \\
\textbf{Generation} & \eta_p^r\colon\ 1 \leq p^r \otimes p & \eta_p^l\colon\ 1 \leq p \otimes p^l
\end{array}$$

The linguistic motivation behind the adjoint types is as follows. The vocabulary of a language consists of two kinds of words: the ones that are atomic such as nouns and the ones that are relational such as adjectives and verbs. The relational types input the atomic types as their arguments and then modify them. The relational types have adjoints in them and the adjoint types represent their arguments. So when such types are juxtaposed with their arguments (in the right order), they will cancel each other out and produce a new modified type. The left and right labels of adjunction denote the order with which a relational type needs to be juxtaposed with its arguments. This order depends on the grammar and differs from language to language. For instance, in English the adjective 'red' occurs before the noun 'car'. So the adjective has type $n \otimes n^l$, for n the type of a noun. This means that it needs an argument of type n and it has to be to the left of its argument. The grammatical reduction of an adjective-noun phrase 'red car' is denoted by the following morphism:

$$\begin{array}{ccl}
\text{red} & \text{car} & \\
(n \otimes n^l) \otimes & n & \leq n \otimes 1 = n
\end{array}$$

Here we are transforming the n in the type of the adjective to itself and we are cancelling the n^r with n in the type of the noun. So the above grammatical reduction corresponds to the morphism $1_n \otimes \epsilon_n^l$. As another example consider a simple intransitive sentence such as 'men sneeze'. Here the noun 'men' has type n and the intransitive verb 'sneeze' has type $n^r \otimes s$. This type means that, according to the grammar of English, an intransitive verb inputs a noun (for simplicity we assign the same type n to nouns and noun phrases) as its subject and has to be the right of that noun. After inputing the subject and modifying it, the verb will produce a sentence, denoted by the type s. This grammatical reduction is denoted by the morphism $\epsilon_n^r \otimes 1_s$ obtained as follows:

$$\begin{array}{ccl} \text{men} & \text{sneeze} & \\ n & \otimes\,(n^r \otimes s) & \leq 1 \otimes s = s \end{array}$$

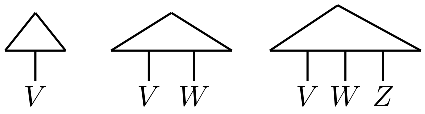
Finally, consider a simple transitive sentence such as 'men like cats'. Here, the transitive verb 'like' needs two arguments of type n and has to be to the right of one and the left of the other. After inputting these nouns, it will modify them and produce a sentence, thus, it has type $n^r \otimes s \otimes n^l$. The morphisms corresponding to the grammatical reduction of a transitive sentence is $\epsilon_n^r \otimes 1_s \otimes \epsilon_n^l$ and is obtained as follows:

$$\begin{array}{cccl} \text{men} & \text{like} & \text{cats} & \\ n & \otimes\,(n^r \otimes s \otimes n^l) \otimes & n^l & \leq 1 \otimes s \otimes 1 = s \end{array}$$

3.2 A compact closed category for meaning

Distributional models of meaning represent meanings of words by vectors. These vectors live in a finite dimensional vector space with a fixed set of basis vectors. Such vector spaces also form a compact closed category. In this category, vector spaces $V, W, ...$ are objects and linear mappings between them $f\colon V \to W$ are morphisms. The composition of morphisms is the composition of linear maps. The tensor product is the tensor product of vector spaces $V \otimes W$, whose unit is the scalar field of the vector spaces; in our case this is the field of reals $\mathbb{R}$. The left and right adjoints are the same, that is we have $V^l \cong V^r \cong V^*$, where V^* is the dual space of V. Since the basis vectors of these vector spaces are fixed, we have an isomorphism between V^* and V, that is $V^* \cong V$.

Diagrammatically speaking, vector spaces are lines and vectors within them are triangles. Each triangle has a number of strings emanating from it. This number denotes the tensor rank of the vector, for instance, $\overrightarrow{v} \in V$, $\overrightarrow{v'} \in V \otimes W$, and $\overrightarrow{v''} \in V \otimes W \otimes Z$ are depicted as follows:



Given a basis $\{\overrightarrow{r}_i\}_i$ for a vector space V, the two cancelation maps become isomorphic to one, since we have $V^* \otimes V \cong V \otimes V^* \cong V \otimes V$. This map is as follows:

$$\epsilon_V\colon\ V \otimes V \to \mathbb{R}$$

Concretely, the application of this map to vectors $\overrightarrow{v}, \overrightarrow{w}$ from V is taking their inner product, which provides us with a number in $\mathbb{R}$, defined as follows:

$$\epsilon_V(\overrightarrow{v} \otimes \overrightarrow{w}) = \langle \overrightarrow{v} \mid \overrightarrow{w} \rangle$$

Representing vectors by their linear expansions, that is $\overrightarrow{v} = \sum_i c_i \overrightarrow{r}_i$, $\overrightarrow{w} = \sum_j c_j \overrightarrow{r}_j$, the above becomes equivalent to the following:

$$c_{ij} \sum_{ij} \langle \overrightarrow{r}_i \mid \overrightarrow{w}_j \rangle$$

For the same reason as above, the two generation maps also become isomorphic to the following one:

$$\eta\colon \ \mathbb{R} \to V \otimes V$$

Concretely, the application of this map to a real number $k \in \mathbb{R}$ produces a vector in $V \otimes V$ defined as follows

$$\eta(k) = k \sum_i \overrightarrow{r}_i \otimes \overrightarrow{r}_i$$

For an example, take V to be a two dimensional space with the basis $\{\overrightarrow{r}_1, \overrightarrow{r}_2\}$. An example of the generation map in this space is as follows:

$$\eta(k) = k(\overrightarrow{r}_1 \otimes \overrightarrow{r}_1 \ + \ \overrightarrow{r}_2 \otimes \overrightarrow{r}_2)$$

Because the basis vectors of V are fixed, the above can be equivalently written in the form of a 2×2 matrix as follows:

$$\eta(k) \ = \ \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$$

4 Frobenius Algebras, Diagrams, and Examples

Again we refer the reader for a formal definition and references to the prequel paper [24]. Informally speaking, some of the objects of a compact closed category might have special properties, referred to by *copying* and *uncopying*. Note that these maps are only defined on certain objects of the category and the family of these objects need not coincide with the family of all the objects of the category. Although the general definition of Frobenius algebras is over any compact closed category, for the purpose of this paper, we will consider the case of vector spaces.

The *copying* property is an expression of the fact that there is a linear way of transforming a certain vector space V to the vector space $V \otimes V$. The linear map corresponding to this transformation is denoted as follows:

$$\Delta\colon V \to V \otimes V$$

Concretely, it acts on a vector $\overrightarrow{v} = \sum_i c_i \overrightarrow{r}_i$ of V as follows:

$$\Delta(\overrightarrow{v}) = \sum_i c_i \overrightarrow{r}_i \otimes \overrightarrow{r}_i$$

So for a two dimensional space with basis vectors $\{\overrightarrow{r}_1, \overrightarrow{r}_2\}$, we have:

$$\Delta(\overrightarrow{v}) = \Delta(c_1\overrightarrow{r}_1 + c_2\overrightarrow{r}_2) = c_1(\overrightarrow{r}_1 \otimes \overrightarrow{r}_1) \ + \ c_2(\overrightarrow{r}_2 \otimes \overrightarrow{r}_2)$$

Using the matrix notation, the above can be equivalently written as:

$$\Delta \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix}$$

The copying map has a unit ι of the type $V \to \mathbb{R}$, which transforms a certain vector space to the scalar field. Concretely, it sends a vector to the sum of its co-ordinates. So in general for $\overrightarrow{v} \in V$, we have:

$$\iota(\overrightarrow{v}) = \iota(\sum_i c_i \overrightarrow{r}_i) = \sum_i c_i$$

An example consider $\iota(\overrightarrow{v}) = \iota(c_1\overrightarrow{r}_1 + c_2\overrightarrow{r}_2) = c_1 + c_2$.

In the context of linguistics one can think of copying as a way of being able to dispatch the information of a vector space to two vector spaces (and by analogy also for the vectors within these vector spaces). For instance, it might be needed to input the information expressed in a noun to a relative pronoun and to the verb of the main clause. In this case, and as we will see in more detail later on, we will copy the vector of the noun and pass a copy to the relative pronoun and another copy to the verb. In other words, we dispatch the information of the noun to the relative pronoun and to the verb. The linguistic application of the unit ι is that sometimes one needs to discard the information of a word or part of a word, in which case ι is applied to the vector of that word. An example is again the case of relative clauses, where the relative pronoun inputs the type of sentence s from the verb but has to discard it as the output of a relative clause is a noun, rather than a sentence.

The *uncopying* map expresses the fact that there is a linear way of transforming the tensor product of a certain vector space V with itself to V. That is we have the following linear map:

$$\mu \colon V \otimes V \to V$$

Concretely, the application of μ on vectors in V is defined as follows:

$$\mu(\overrightarrow{v} \otimes \overrightarrow{w}) = \mu(\sum_i c_i \overrightarrow{r}_i \otimes \sum_j c_j \overrightarrow{r}_j) = \sum_i c_i c_j \delta_{ij} \overrightarrow{r}_j$$

The notation $\delta_{ij}\overrightarrow{r}_i$ is defined as follows:

$$\delta_{ij}\overrightarrow{r}_j = \begin{cases} \overrightarrow{r}_i & i = j \\ 0 & i \neq j \end{cases}$$

As an example we have

$$\mu(\overrightarrow{v} \otimes \overrightarrow{w}) = \mu((c_1\overrightarrow{r}_1 + c_2\overrightarrow{r}_2) \otimes (c_3\overrightarrow{r}_3 + c_4\overrightarrow{r}_4)) = c_1c_3\overrightarrow{r}_1 + c_2c_4\overrightarrow{r}_2$$

Using the matrix notation, the above can be written as follows

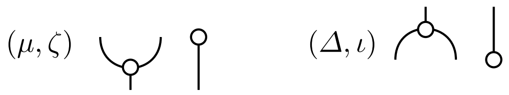
$$\mu\left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \otimes \begin{pmatrix} c_3 \\ c_4 \end{pmatrix}\right) = \mu\begin{pmatrix} c_1c_3 & c_1c_4 \\ c_2c_3 & c_2c_4 \end{pmatrix} = \begin{pmatrix} c_1c_3 \\ c_2c_4 \end{pmatrix}$$

The above is equal to the point wise multiplication of the two input vectors, that is $\overrightarrow{v} \odot \overrightarrow{w}$. The uncopying map has a unit $\zeta \colon \mathbb{R} \to V$, which transforms a real number k to a vector whose all co-ordinates are k. That is, we have

$$\zeta(k) = \sum_i k\overrightarrow{r}_i$$

An example consider $\zeta(k) = k\overrightarrow{r}_1 + k\overrightarrow{r}_2$.

The diagrammatic forms of the Froebnius morphisms are as follows:

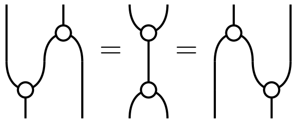


In the context of linguistics, the uncopying map can be thought of as a way of merging or combing information. In this case, the information of two vectors from a vector space V can be merged into the information of one vector in V. For instance, after a relative pronoun has made a copy of the vector of a word, and these copies are dispatched to various parts of the clause, there is a need to put the modified information together, in other words, merge them, to obtain one single vector as the output of the relative clause. We will see examples of this feature in the proceeding sections. The ζ map is used to create a vector for a word that has been dropped from a phrase. For instance the relative pronoun 'that' is usually dropped, as is the case in the clause 'dogs I saw yesterday'. The original form of this phrase is 'dogs that I saw yesterday'. In such cases, the ζ map enables us to generate a vector for the dropped pronoun.

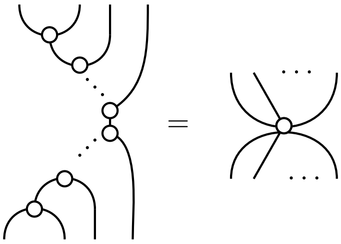
The above linear maps satisfy a number of equations (e.g. commutativity and specialty), the major of which is the *Frobenius* condition with the following formal form:

$$(\mu \otimes 1_V) \circ (1_V \otimes \mathit{\Delta}) \;=\; \mathit{\Delta} \circ \mu \;=\; (1_V \otimes \mu) \circ (\mathit{\Delta} \otimes 1_V)$$

The diagrammatic form this property is as follows:



Informally speaking, this property says that if one has the tensor of a vector space with itself, that is $V \otimes V$, then one can keep the first space and copy the second space followed by then uncopying the first V with the first output of copying and keeping the second output (or do these operations in the opposite order, that is copy the first V and then uncopy the second output with V), the result is the same as an uncopy followed by a copy. This property allows us to copy and uncopy in an alternate way and many times, and at the end being able to merge all the operations into a big uncopying and copying, which performs all of the previous operations at once. Diagrammatically, we have the following:



This property is sometimes referred to by the *spider* property. In the context of linguistics, this property expresses the fact that if we have two words (within the same vector space) and we dispatch the information of the second word and then merge the first output with the first word, this will result in the same pair of words obtained from the operation of first merging the information of the original two words, then dispatching them into two other words.

5 From Grammar to Meaning

Since both pregroup grammars and vector spaces are compact closed, there exists a structure-preserving map between the two $F\colon \mathit{Preg} \to \mathit{FVect}$. This map and its mathematical properties were developed and discussed in previous work [6,15,5]. Here, we review its main properties, which include assigning to each atomic grammatical type a vector space as follows:

$$F(1) = \mathbb{R} \qquad F(n) = N \qquad F(s) = S$$

Naturally, the unit of juxtaposition is mapped to the unit of tensor product in vector spaces. We map the atomic type n to a vector space N and the atomic type s to a vector space S. These vector spaces can be built in different ways. In the sections that follow, we present two instantiations for them.

A juxtaposition of types is mapped to the tensor product of the vector spaces assigned to each type, that is we have:

$$F(p \otimes q) = F(p) \otimes F(q)$$

The two left and right adoints of each grammatical type are sent to the dual of the vector spaces assigned to the original type, that is we have:

$$F(p^l) = F(p^r) = F(p)^* \cong F(p)$$

As an example, consider the grammatical type of a transitive verb, that is $n^r \otimes s \otimes n^l$, then its vector space assignment is computed as follows:

$$F(n^r \otimes s \otimes n^l) = F(n^r) \otimes F(s) \otimes F(n^l) = F(n) \otimes F(s) \otimes F(n) = N \otimes S \otimes N$$

This assignment means that the meaning vector of a transitive verb is a vector in the tensor space $N \otimes S \otimes N$.

The grammatical reductions, i.e. the partial order morphisms of a pregroup, are mapped to linear maps. That is, a partial $p \leq q$ in a pregroup, is mapped to a linear map $f_{\leq} \colon F(p) \to F(q)$ in vector spaces. The cancelation ϵ and generation η maps of a pregroup are assigned to cancelation and generation maps of a vector space.

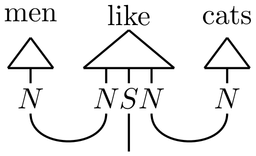
As an example, recall the grammatical reduction of a transitive sentence. This reduction is interpreted as follows in vector spaces:

$$F(\epsilon^r_n \otimes 1_s \otimes \epsilon^r_n) = F(\epsilon^r_n) \otimes F(1_s) \otimes F(\epsilon^l_n) = F(\epsilon_n) \otimes F(1_s) \otimes F(\epsilon_n) = \epsilon_N \otimes 1_S \otimes \epsilon_N$$

To obtain a vector meaning for a string of words, we apply the vector space interpretation of the grammatical reduction of the string to the vectors of the words within the string. For instance, the vector meaning of the sentence ‘men like cats’ is as follows:

$$F(\epsilon_n^r \otimes 1_s \otimes \epsilon_n^l)(\overrightarrow{men} \otimes \overrightarrow{like} \otimes \overrightarrow{cats})$$

This meaning is depictable as follows:



As you can see, depending on their types, the distributional meanings of the words are either atomic vectors or linear maps. For instance, the distributional meaning of ‘men’ is a vector in N, whereas the distributional meaning of ‘love’ is in the space $N \otimes S \otimes N$, which is equivalent to a linear map from $N \otimes N$ to S.

6 Modelling Possessive Relative Pronouns

In this section, we use the categorical constructions introduced in the previous sections and present a model for the possessive relative pronoun in its subject and object roles. We start with the pregroup representations of the grammatical types of these pronouns and show how a relative clause containing them reduces. We then develop a categorical semantics for these using the cancelation and generation maps of compact closed categories and the morphisms of Frobenius algebras. In a nutshell, the cancelation map ϵ models the application of the semantics of one word to another; the generation map η, passes information among the words by bridging the intermediate words; and the Frobenius operations dispatch and combine the noun vectors and discard the sentence vectors. The end product of this model is a compositional vector representation for the meanings of possessive relative clauses.

The possessive relative pronoun ‘whose’ can occur in a subject or object position, hence producing subject and object possessive clauses. The general forms of these clauses are as follows:

Poss Subj: Possessor whose Subject Verb Object
Poss Obj: Possessor whose Object Subject Verb

An example for each of the above cases is as follows:

Poss Subj: ‘author whose book entertained John’
Poss Obj: ‘author whose book John read’

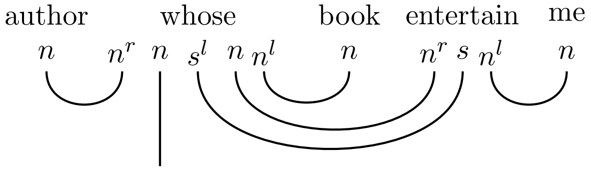
Informally speaking, the head of such a clause is a possessor who owns an item, which is either the subject or the object of the rest of the clause. For instance, in the subject example above, the possessor head noun ‘author’ owns a book which has entertained John and in the object case, John has read this book. In a manner of speaking, we are describing the head of the clause through his possession. For instance, in the above clauses we are describing an ‘author’ through his ‘book’.

The syntactic role of the relative pronoun ‘whose’ is to relate the head of the clause to the rest of the clause by first inputting the modified subject or object of the rest of the clause, then inputting the ‘owner’ of this noun, and finally letting the head of the clause be modified by these owners and outputting their modified versions. For instance, in the subject example above, ‘whose’ first inputs the subject of ‘book entertained John’, that is, books which have been modified by the verb phrase ‘entertained John’. Then it inputs the owners of these books, and finally modifies ‘author’ by them, outputting the authors whose books entertained John. Set theoretically speaking, ‘whose’ is to choose from the set of authors, the ones that have books which have entertained John, in the first example, and the ones that have books which John read, in the second example.

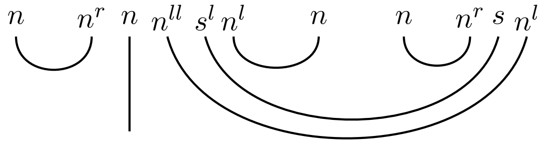
The pregroup types of these pronouns are as follows:

Poss Subj: $n^r n s^l n n^l$ **Poss Obj:** $n^r n n^{ll} s^l n^l$

The grammatical reduction of the subject case is as follows:



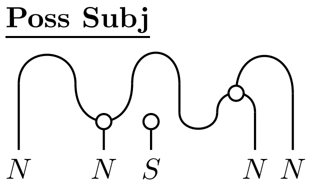
The grammatical reduction of the object case is as follows:

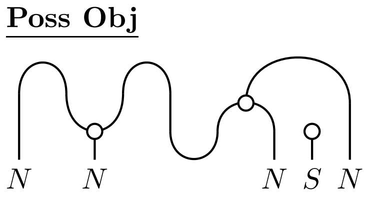


The vector spaces in which the meaning vectors of ‘whose’ live, are obtained as follows:

$$F(n^r n s^l n n^l) = N \otimes N \otimes S \otimes N \otimes N$$
$$F(n^r n n^{ll} s^l n^l) = N \otimes N \otimes N \otimes S \otimes N$$

Other than passing the information around, these pronouns also act on the subject/object of the clause to establish the ownership relationship between them and the possessor. We denote this action by a ‘’s’-labeled box with the type $N \to N$; it takes the subject/object as input then outputs its owner. The remaining structure of the pronoun duplicates the information of the subject/object and passes a copy to the verb, then unifies the possessor with the owners of the modified subject/object. These processes are depicted below:

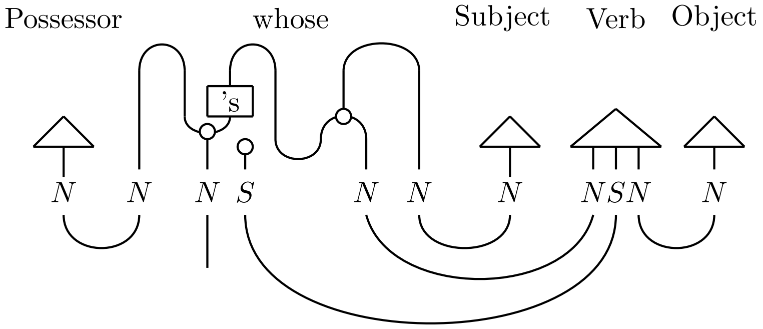




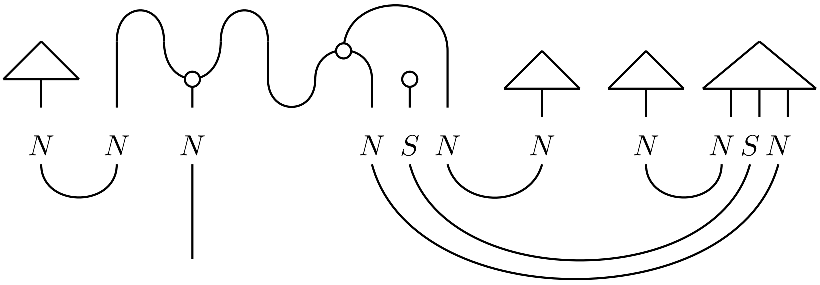
The above diagrams correspond to the following morphisms:

$$(1_N \otimes \mu_N \otimes \zeta_S \otimes \epsilon_N \otimes \varDelta_N \otimes 1_N) \circ (1_{N\otimes N} \otimes \overline{\text{'s}} \otimes 1_{N\otimes N\otimes N}) \circ (\eta_N \otimes \eta_N \otimes \eta_N)$$
$$(1_N \otimes \mu_N \otimes \epsilon_N \otimes \varDelta_N \otimes \zeta_S \otimes 1_N) \circ \big(1_{N\otimes N} \otimes \overline{\text{'s}} \otimes 1_{N\otimes N\otimes N}\big) \circ (\eta_N \otimes \eta_N \otimes \eta_N)$$

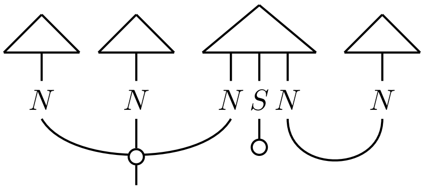
The diagrams of the meanings of these clauses visualise the above processes in the clause. For instance, consider the diagram of the subject case:



The pronoun ‘whose’ inputs the information of the subject and outputs its owners after applying the $\boxed{\text{'s}}$ to it. This information is unified with the possessor via a μ map, then a copy of it is passed to the verb via a Δ map and outputted after the verb has acted on it. The flow of information happens via the three η maps. A similar process takes place in the object case, as depicted below:



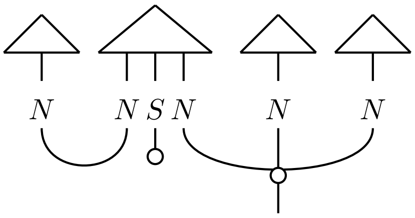
The actions of the above processes are summarised in their normal forms. Consider the case of the subject clause, normalised below:



Here, the η and ϵ maps are yanked and the result is displayed in a more clear way: the verb acts on the subject and object, but does not return a sentence, as usual, since its sentence dimension is discarded by 'whose'. Instead, the information of the subject, after the verb has acted on it, is inputted to $\boxed{\text{'s}}$ then unified with the information of the possessor. The meaning of the clause is the result of this unification. The above diagram corresponds to the following categorical morphisms:

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \circ (1_N \otimes \mu_N \otimes \iota_S \otimes \epsilon_N) \left(\overrightarrow{\text{Poss}} \otimes \overrightarrow{\text{Sub}} \otimes \overrightarrow{\text{Verb}} \otimes \overrightarrow{\text{Obj}}\right)$$

The normal form of the object clause, describing a similar process, is as follows:



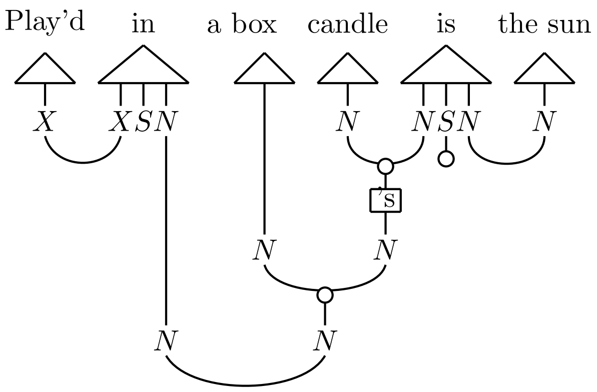
The above corresponds to the following morphism:

$$\mu_N \circ \left(\overline{\text{'s}} \otimes 1_N\right) \circ (\epsilon_N \otimes \iota_S \otimes \mu_N \otimes 1_N) \left(\overrightarrow{\text{Sub}} \otimes \overrightarrow{\text{Verb}} \otimes \overrightarrow{\text{Obj}} \otimes \overrightarrow{\text{Poss}}\right)$$

As an example of the occurrence of a relative clause in a sentence, consider the third verse of the translation of quatrain (XLVI) of Omar Khayyam (11th century Persian poet and mathematician) by Fitzgerald [9]. The full quatrain is as follows (the choice of capital letters is by Fitzgerald):

For in and out, above, about, below
Tis nothing but a magic shadow-show
Play'd in a box ***whose*** *candle is the sun,*
Round which *we Phantom Figures come and go.*

In 'Play'd in a box **whose** candle is the sun', we have the possessive pronoun 'whose' in a subject role. It is modifying the noun 'candle' which is the subject of the verb 'is' by the possessor noun 'a box'. This part is analysed in exactly the same way as presented in section 6. That is, the noun phrases 'a box' , 'candle', and 'the sun' have type n, whose has type $n^r n s^l n^l$, and the predicate 'is' has type $n^r s n^l$. The possessive clause 'a box whose candle is the sun', which has type n, is then used by the preposition 'in' to modify the verb phrase 'play'd'. For this part we may analyse 'play'd' as a verb phrase $n^r s$ and hence the preposition 'in' will have type $(n^r s)^r s n^l$. Or one can argue that 'play'd' is an abbreviated sentence of type s whose original sentence was something like 'it is played'. In this case, the preposition 'in' will have type $s^r s n^l$. In either case, the general grammatical reduction is the same: 'in' inputs a verb phrase or a sentence on the left and a noun (which is the possessive clause) on the right; it then modifies the former with the latter and outputs a sentence. The normal form of the compact closed meaning of this verse can then be depicted as follows, where X can be either $F(n^r)$ or $F(n^r s)$, representing either of the discussed cases:



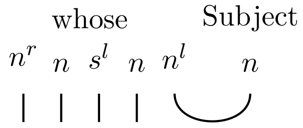
It is apparent that the process is very compositional. One can plug in the meanings of different parts of the phrases together to obtain a meaning for the full sentence.

7 Decomposing Whose

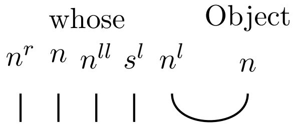
In [19], Lambek suggests that the type of the compounds ‘whose Subject’ and ‘whose Object’ in the subject and object possessive clauses should be the same as the type of the relative pronoun ‘that’ in its subject and object roles, respectively. These types are as follows:

Subj: $n^r n s^l n$ **Obj:** $n^r n n^{ll} s^l$

We observe that this reduction is indeed the case in our setting, as shown in the corresponding syntactic computations, depicted as follows:



$$(n^r n s^l n n^l) n \leq n^r n s^l n$$



$$(n^r n n^{ll} s^l n^l) n \leq n^r n n^{ll} s^l$$

This way of looking at the type of ‘whose’ suggests that any possessive relative clause can be seen as a relative clause without the actual possessive pronoun ‘whose’. This is possible by a combination of two relative pronouns and the predicate ‘has’, as follows:

Poss Subj: Possessor that has Subject that Verb Object.
Poss Obj: Possessor that has Object that Subject Verb.

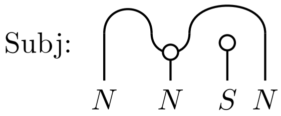
For instance, for our above examples we would have:

> ‘author that has a book that entertained John’
> ‘author that has a book that John read’

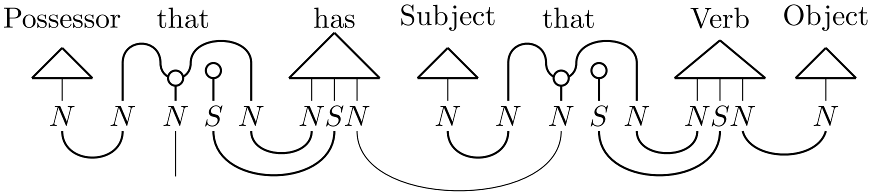
We verify that above suggestion is correct by showing that the possessive relative clauses and their non-whose version have the same meanings.

Proposition 1. *The clause ‘Possessor that has Subject that Verb Object’ has the same vector space meaning as the clause ‘Possessor whose Subject Verb Object’.*

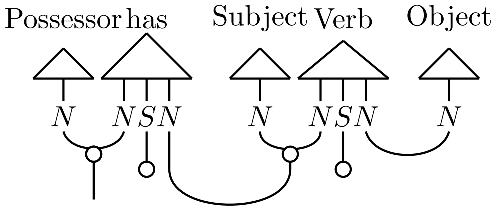
Proof. The meaning of the subject relative pronoun, as developed in previous work [24], is depicted as follows:



Hence, the meaning of the clause ‘Possessor that has Subject that Verb Object’ is computed as follows:



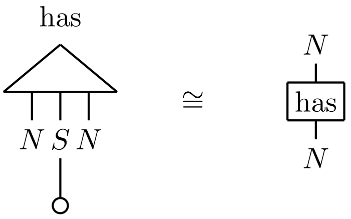
This normalises to the following:



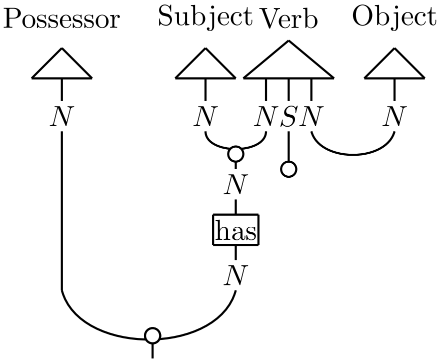
The vector space meaning of the application of the unit of the Frobenius algebra, that is the ι map, on the predicate ‘has’ is as computed as follows:

$$(1_N\otimes\iota_S\otimes 1_N)(\overline{\text{has}}) := \sum_{ijk} C_{ijk}\overrightarrow{n}_i\otimes\iota(\overrightarrow{s}_j)\otimes\overrightarrow{n}_k = \sum_{ijk} C_{ijk}\overrightarrow{n}_i\otimes\overrightarrow{n}_k = \sum_{ik} C_{ik}\overrightarrow{n}_i\otimes\overrightarrow{n}_k$$

This is an element of the space $N \otimes N$, which in our vector space setting is isomorphic to the set of linear maps from N to N. Pictorially, we have:



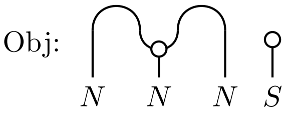
As a result, the above simplifies further to the following:



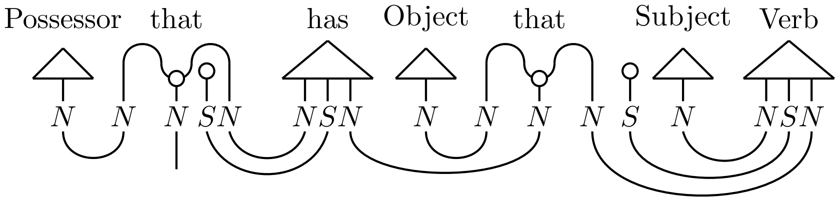
This is the same as the meaning of the phrase 'Possessor whose Subject Verb Object'.

Proposition 2. *The clause 'Possessor that has Object that Subject Verb' has the same vector space meaning as the clause 'Possessor whose Object Subject Verb'.*

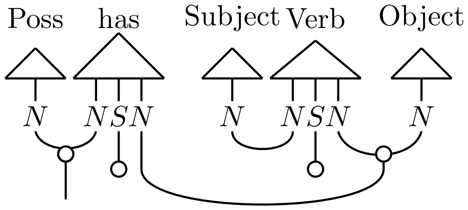
Proof. The meaning of the object relative pronoun as developed in previous work [24], is depicted as follows:



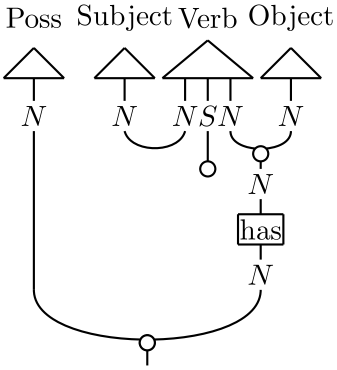
Hence, the meaning of the clause 'Possessor that has Object that Subject Verb' is computed as follows:



The above normalises to the following:



Using the equivalence result of the previous proposition on the 'has' predicate, the above further normalises to the following:



This is the same as the meaning of the phrase 'Possessor whose Object Subject Verb'.

8 Truth Theoretic Instantiations

In this section, we provide a truth-theoretic instantiation for our model. This instantiation is designed only as a theoretical example and to show that vector spaces can be used to recast set theoretic Montague-style semantics [23].

Take N to be the vector space spanned by a set of individuals $\{\overrightarrow{n}_i\}_i$ that are mutually orthogonal. For example, $\overrightarrow{n}_1$ represents the individual Mary, $\overrightarrow{n}_{25}$ represents Roger the dog, $\overrightarrow{n}_{10}$ represents John, and so on. A sum of basis vectors in this space represents a common noun; e.g. $\overrightarrow{man} = \sum_i \overrightarrow{n}_i$, where i ranges over the basis vectors denoting men. We take S to be the one dimensional space spanned by the single vector $\overrightarrow{1}$. We set the unit vector spanning S to represent the truth value 1 and the zero vector to represent the truth value 0.

Since the sentence space is the real line, we also have access to an infinite set of numbers (now represented by a vector, e.g. number 2 by vector $\overrightarrow{2}$, number 3 by vector $\overrightarrow{3}$ and so on). We interpret these numbers as *truth weights*. In a very loose sense, the real line can be seen as a fuzzy set in the style of fuzzy logic [29]. The intuitive reading of the truth of the predicates over the real interval is that they apply to their arguments not in an absolute sense, but with a certain weight or degree. In other words, whereas in the world of sets, a relation (representing a predicate) is either true or false about an element of a set, in our world, we have weighted relations and a predicate may hold about an element with a certain weight or degree.

A transitive verb w, which is a vector in the space $N \otimes S \otimes N$, is represented as follows:

$$\overline{w} := \sum_{ij} \overrightarrow{n}_i \otimes (\alpha_{ij}\overrightarrow{1}) \otimes \overrightarrow{n}_j \quad \text{if } \overrightarrow{n}_i \ w\text{'s } \overrightarrow{n}_j \text{ with degree } \alpha_{ij}, \text{ for all } i, j$$

This means that a verb may apply to the subject and object not in an absolute way, that is to say, not only that either they are related via the relation represented by the verb or not, but that they are related with a certain degree. For instance, the subject John and the object Mary might not have an absolute love relationship with each other, but might love each other to a certain extent, that is to say, with a certain degree.

Further, since S is one-dimensional with its only basis vector being $\overrightarrow{1}$, the transitive verb can be represented by the following element of $N \otimes N$:

$$\sum_{ij} \alpha_{ij} \overrightarrow{n}_i \otimes \overrightarrow{n}_j \quad \text{if} \quad \overrightarrow{n}_i \ w\text{'s } \overrightarrow{n}_j \text{ with degree } \alpha_{ij}$$

Restricting to either $\alpha_{ij} = 1$ or $\alpha_{ij} = 0$ provides a 0/1 meaning, i.e. either $\overrightarrow{n}_i$ w's $\overrightarrow{n}_j$ or not. Letting α_{ij} range over the interval $[0, 1]$ enables us to represent degrees as well as limiting cases of truth and falsity. For example, the verb "love", denoted by $\overline{love}$, is represented by $\sum_{ij} \alpha_{ij} \overrightarrow{n}_i \otimes$

$\overrightarrow{n}_j$ if $\overrightarrow{n}_i$ loves $\overrightarrow{n}_j$with degree α_{ij}. If we take α_{ij} to be 1 or 0, from the above we obtain $\sum_{(i,j)\in R_{love}} \overrightarrow{n}_i \otimes \overrightarrow{n}_j$, where R_{love} is the set of all pairs (i,j) such that $\overrightarrow{n}_i$ loves $\overrightarrow{n}_j$. Note that, with this definition, the sentence space has already been discarded, and so for this instantiation the ι map, which is the part of the relative pronoun interpretation designed to discard the relative clause after it has acted on the head noun, is not required.

The ownership morphism which is the map $\overline{\text{'s}}\colon N \to N$ is represented as follows:

$$\overline{\text{'s}}(\overrightarrow{n}_{h'}) = \sum_{h''\in O} \overrightarrow{n}_{h''} \quad \text{whenever } \overrightarrow{n}_{h''} \text{ is an owner of } \overrightarrow{n}_{h'}$$

For common nouns $\overrightarrow{\text{Subject}} = \sum_{k\in K} \overrightarrow{n}_k$, $\overrightarrow{\text{Object}} = \sum_{l\in L} \overrightarrow{n}_l$, and $\overrightarrow{\text{Possessor}} = \sum_{h\in P} \overrightarrow{n}_h$, where k, l, and h range over the sets of basis vectors representing the respective common nouns, the truth-theoretic meaning of a noun phrase modified by a possessive subject relative clause is computed as follows:

$$\overrightarrow{\text{Possessor whose Subject Verb Object}} :=$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \circ (1_N \otimes \mu_N \otimes \epsilon_N) \left(\overrightarrow{\text{Poss}} \otimes \overrightarrow{\text{Sub}} \otimes \overrightarrow{\text{Verb}} \otimes \overrightarrow{\text{Obj}}\right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \circ (1_N \otimes \mu_N \otimes \epsilon_N) \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{k\in K} \overrightarrow{n}_k \otimes (\sum_{ij} \alpha_{ij} \overrightarrow{n}_i \otimes \overrightarrow{n}_j) \otimes \sum_{l\in L} \overrightarrow{n}_l\right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{ij,k\in K,l\in L} \alpha_{ij} \mu_N(\overrightarrow{n}_k \otimes \overrightarrow{n}_i) \otimes \epsilon_N(\overrightarrow{n}_j \otimes \overrightarrow{n}_l)\right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{ij,k\in K,l\in L} \alpha_{ij} \delta_{ki} \overrightarrow{n}_i \delta_{jl}\right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{k\in K,l\in L} \alpha_{kl} \overrightarrow{n}_k\right) =$$

$$\mu_N \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{k\in K,l\in L} \alpha_{kl}\ \overline{\text{'s}}(\overrightarrow{n}_k)\right) = \mu_N \left(\sum_{h\in P} \overrightarrow{n}_h \otimes \sum_{k'\in O,l\in L} \alpha_{k'l}\ \overrightarrow{n}_{k'}\right)$$

$$= \sum_{h\in P,k'\in O,l\in L} \alpha_{k'l}\ \mu_N(\overrightarrow{n}_h \otimes \overrightarrow{n_{k'}}) = \sum_{h\in P,k'\in O,l\in L} \alpha_{k'l} \delta_{hk'} \overrightarrow{n}_{k'} = \sum_{h\in P,l\in L} \alpha_{hl} \overrightarrow{n}_h$$

The result is as it should be: the sum of the possessor individuals who own the subject individuals weighted by the degree with which the subjects have acted on the object individuals via the verb. A similar computation, with the difference that the μ and ϵ maps are swapped and the ownership morphism $\overline{\text{'s}}$ acts on the object, provides the truth-theoretic semantics of the possessive object relative clause, which will end up being $\sum_{h\in P,k\in K}\alpha_{hk}\overrightarrow{n}_h$.

As an example consider the possessive subject relative clause "authors whose books entertained John". Take N to be the vector space spanned by the set of all people, including authors, and objects, and books. The authors form a subspace on N whose basis vectors are denoted by $\overrightarrow{a}_h$, where h ranges over the set of authors denoted by P. The books form another subspace whose basis vectors are denoted by $\overrightarrow{b}_k$, where k ranges over the set of books denoted by K. Hence, "authors" and "books" are common nouns $\sum_{h\in P}\overrightarrow{a}_h$ and $\sum_{k\in K}\overrightarrow{b}_k$. The transitive verb "entertain" is defined by $\sum_{(i,j)\in R_{entertain}}\overrightarrow{b}_i\otimes\overrightarrow{n}_j$. Now set "John" to be the individual $\overrightarrow{n}_1$ and assume that $\overrightarrow{n}_8$ is another individual which is not necessarily an author or a book. The above common nouns, verb, and ownership relation are instantiated as follows:

$$
\begin{aligned}
\overrightarrow{\text{authors}} &= \overrightarrow{a}_5+\overrightarrow{a}_6+\overrightarrow{a}_7\\
\overrightarrow{\text{books}} &= \overrightarrow{b}_2+\overrightarrow{b}_3+\overrightarrow{b}_4\\
\overline{\text{entertain}} &= (\overrightarrow{b}_2\otimes\overrightarrow{n}_1)+(\overrightarrow{b}_3\otimes\overrightarrow{n}_1)+(\overrightarrow{b}_4\otimes\overrightarrow{n}_2)+(\overrightarrow{n}_5\otimes\overrightarrow{n}_2)\\
\overline{\text{'s}}(\overrightarrow{b}_2) &= \overrightarrow{a}_5+\overrightarrow{a}_6 \quad \overline{\text{'s}}(\overrightarrow{b}_3)=\overrightarrow{n}_2 \quad \overline{\text{'s}}(\overrightarrow{b}_b)=\overrightarrow{a}_8
\end{aligned}
$$

The vector corresponding to the meaning of "authors whose books entertained John" is computed as follows:

$$\overrightarrow{\text{authors whose books entertained John}} :=$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \circ (1_N \otimes \mu_N \otimes \epsilon_N) \left(\sum_{h \in P} \overrightarrow{a}_h \otimes \sum_{k \in K} \overrightarrow{b}_k \otimes (\sum_{(i,j) \in R_{entertain}} \overrightarrow{b}_i \otimes \overrightarrow{n}_j) \otimes \overrightarrow{n}_1 \right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h \in P} \overrightarrow{a}_h \otimes \sum_{k \in K, (i,j) \in R_{entertain}} \mu_N(\overrightarrow{b}_k \otimes \overrightarrow{b}_i) \epsilon_N(\overrightarrow{n}_j \otimes \overrightarrow{n}_1) \right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h \in P} \overrightarrow{a}_h \otimes \sum_{(i,j) \in R_{entertain}} \delta_{ki} \overrightarrow{b}_i \delta_{j1} \right) =$$

$$\mu_N \circ \left(1_N \otimes \overline{\text{'s}}\right) \left(\sum_{h \in P} \overrightarrow{a}_h \otimes (\overrightarrow{b}_2 + \overrightarrow{b}_3) \right) =$$

$$\mu_N \left(\sum_{h \in P} \overrightarrow{a}_h \otimes (\overline{\text{'s}}(\overrightarrow{b}_2) + \overline{\text{'s}}(\overrightarrow{b}_3)) \right) = \mu_N \left(\sum_{h \in P} \overrightarrow{a}_h \otimes (\overrightarrow{a}_5 + \overrightarrow{a}_6 + \overrightarrow{n}_2) \right) =$$

$$\sum_{h \in P} \mu_N(\overrightarrow{a}_h \otimes (\overrightarrow{a}_5 + \overrightarrow{a}_6 + \overrightarrow{n}_2)) = \overrightarrow{a}_5 + \overrightarrow{a}_6$$

As expected, the result is the sum of the author basis vectors who wrote books that entertained John.

The verb ‘entertain‘ can also have degrees of truth, for example instantiated as follows:

$$1/6(\overrightarrow{b}_2 \otimes \overrightarrow{n}_1) + 2/6(\overrightarrow{b}_3 \otimes \overrightarrow{n}_1) + 2/6(\overrightarrow{b}_4 \otimes \overrightarrow{n}_2) + 1/6(\overrightarrow{n}_5 \otimes \overrightarrow{n}_2)$$

In this case the result of the above phrase will be as follows:

$$1/6(\overrightarrow{a}_5 + \overrightarrow{a}_6)$$

Intuitively, this means that we are not just considering the set of authors whose books entertained John, but those elements of this set, that is those authors, whose books have entertained John with a certain weight, namely 1/6. For this example, these are authors a_5 and a_6.

9 Embedding Predicate Semantics

In this section, we provide a set-theoretic interpretation for relative clauses according to the constructions discussed in [25]. In the prequel to this paper, we presented the case of subject and object clauses, here we extend that setting to possessive clauses. We start by fixing a universe of elements $\mathcal{U}$. A proper noun is an individual (i.e. an element) in this set; a common noun is the set of the individuals that have the property expressed by the common noun; hence common nouns are unary predicates over $\mathcal{U}$. An intransitive verb is the set of all individuals that are acted upon by the relationship expressed by the verb, hence these are unary predicates over $\mathcal{U}$. A transitive verb is the set of pairs of individuals that are related by the relationship expressed by the verb; hence these are binary predicates over $\mathcal{U} \times \mathcal{U}$. Here is an example for each case:

$$\begin{aligned}
[\![Mary]\!] &= \{u_1\} \\
[\![Author]\!] &= \{x \in \mathcal{U} \mid \text{"}x \text{ is an author"}\} \\
[\![Sleep]\!] &= \{x \in \mathcal{U} \mid \text{"}x \text{ sleeps"}\} \\
[\![Entertain]\!] &= \{(x,y) \in \mathcal{U} \times \mathcal{U} \mid \text{"}x \text{ entertains } y\text{"}\}
\end{aligned}$$

The subject and object relative clauses are interpreted as follows:

$$\{x \in \mathcal{U} \mid x \in [\![Subj]\!], (x,y) \in [\![Verb]\!], \text{ for all } y \in [\![Obj]\!]\}$$
$$\{y \in \mathcal{U} \mid y \in [\![Obj]\!], (x,y) \in [\![Verb]\!], \text{ for all } x \in [\![Subj]\!]\}$$

These clauses pick out individuals which belong to the intepretation of subject/object and which are in the relationship expressed by the verb with all elements of the interpretation of object/subject. Examples are ‘men who love Mary’ and ‘men whom cats love’, interpreted as follows, for $x \in \mathcal{U}$:

$$\{x \in \mathcal{U} \mid x \in [\![men]\!], (x,y) \in [\![Love]\!], \text{ for all } y \in [\![Mary]\!]\}$$
$$\{y \in \mathcal{U} \mid y \in [\![men]\!], (x,y) \in [\![Love]\!], \text{ for all } x \in [\![cat]\!]\}$$

The possessive relative clauses are interpreted as follows:

$$\{x \mid x \in [\![Poss]\!], (x,y) \in [\![Has]\!], \text{ for all } y \in [\![Subj]\!], (y,z) \in [\![Verb]\!], \text{ for all } z \in [\![Obj]\!]\}$$
$$\{x \mid x \in [\![Poss]\!], (x,y) \in [\![Has]\!], \text{ for all } y \in [\![Obj]\!], (z,y) \in [\![Verb]\!], \text{ for all } z \in [\![Subj]\!]\}$$

In the above, Has is the predicate which represents the ownership relation, defined as follows:

$$[\![Has]\!] = \{(x,y) \mid \text{"}x \text{ has } y\text{"}\}$$

For example, the clause ‘author whose book entertained John’ is interpreted as follows, where since $[\![John]\!]$ is a singleton we are able to abbreviate the interpretation:

$$\{x \mid x \in [\![Author]\!], (x,y) \in [\![Has]\!], \text{ for all } y \in [\![Book]\!], (y, [\![John]\!]) \in [\![Entertain]\!]\}$$

We obtain the vector space forms of the above predicate interpretations by developing a map from the category of sets and relations to the category of finite dimensional vector spaces with a fixed orthogonal basis and linear maps over them, that is a map with types as follows:

$$e\colon Rel \hookrightarrow FVect$$

This map turns a set T into a vector space V_T spanned by the elements of T and an element $t \in T$ into a basis vector $\overrightarrow{t}$ of V_T. A subset W of T is turned into the sum vector of its elements, i.e. $\sum_i \overrightarrow{w}_i$, where w_i's enumerate over elements of W. A relation $R \subseteq T \times T'$ is turned into a linear map from T to T', or equivalently as an element of the space $T \otimes T'$; we represent this element by the sum of its basis vectors $\sum_{ij} \overrightarrow{t}_i \otimes \overrightarrow{t'}_j$, here t_i's enumerate elements of T and t'_j's enumerate elements of T'.

The application of a relation $R \subseteq T \times T'$ to its arguments is turned into the inner product of the vector space forms of the relation and the arguments. In our sum representation, $R(a)$ when $a \in T$ and $R^{-1}(b)$ when $b \in T'$ are turned into the following:

$$\sum_{ij} \langle a \mid \overrightarrow{t}_i \rangle \overrightarrow{t'}_j \qquad\qquad \sum_{ij} \overrightarrow{t}_i \langle \overrightarrow{t'}_j \mid b \rangle$$

To check wether an element t is in the set T, we use the Frobenius operation μ on T, that is $\mu\colon T \otimes T \to T$, as follows:

$$t \in T \quad \text{whenever} \quad \mu(\overrightarrow{t}, \sum_i \overrightarrow{t}_i) = \overrightarrow{t}$$

The intersection of two sets T, T' is computed by generalising the above via the Frobenius μ operation on the whole universe, that is $\mu\colon \mathcal{U} \otimes \mathcal{U} \to \mathcal{U}$, or on a set containing both of these sets, e.g. $T \cup T'$, that is $\mu\colon (T \cup T') \otimes (T \cup T') \to (T \cup T')$. In either case, for $\sum_i \overrightarrow{t}_i$ and $\sum_j \overrightarrow{t'}_j$ representations of T and T' we obtain the following for the intersection:

$$T \cap T' \;:=\; \mu(\sum_i \overrightarrow{t}_i, \sum_j \overrightarrow{t'}_j)$$

We are now ready to show that the vector space forms of the predicate interpretations of relative clauses, developed along side the constructions described above, provide us with the same semantics as the truth-theoretic vector space semantics developed in Section 8.

Proposition 3. *The map e, described above, provides a 0/1 truth-theoretic interpretation for possessive relative clauses.*

Proof. The first step is to instantiate the proper and common nouns, the intransitive and transitive verbs. For this, we set the universe $\mathcal{U}$ to be the individuals representing the nouns of the language, then the map e instantiates as follows:

- The universe $\mathcal{U}$ becomes the vector space $V_{\mathcal{U}}$, spanned by its elements. So an element $u_i \in \mathcal{U}$ becomes a basis vector $\overrightarrow{u}_i$ of $V_{\mathcal{U}}$.
- A proper noun $a \in \mathcal{U}$ becomes a basis vector $\overrightarrow{u}_a$ of $V_{\mathcal{U}}$.
- A common noun $c \subseteq \mathcal{U}$ becomes a subspace of $V_{\mathcal{U}}$, represented by the sum of the representations of its elements $\overrightarrow{u}_i$, that is
$$c \hookrightarrow \overrightarrow{c} \quad :: \quad \sum_i \overrightarrow{u}_i \quad \text{for all } u_i \in c$$
- A transitive verb $w \subseteq \mathcal{U} \times \mathcal{U}$ becomes a linear map $\overline{w}$ in $V_{\mathcal{U}} \otimes V_{\mathcal{U}}$, represented by the sum of its basis vectors $\overrightarrow{u}_i, \overrightarrow{u}_j$, which in this case are pairs of its elements (u_i, u_j), that is
$$w \hookrightarrow \overline{w} \quad :: \quad \sum_{ij} \overrightarrow{u}_i \otimes \overrightarrow{u}_j \quad \text{for all } (u_i, u_j) \in w$$

The next step is to use the above definitions and develop the vector space forms of the predicate interpretations of the relative clauses. To do so, we turn these interpretations into intersections of subsets and check the membership relation via the μ map. As for the application of the predicates to the interpretations of their arguments, we use their relational images on subsets. That is, for $R \subseteq \mathcal{U} \times \mathcal{U}$ and $T \subseteq \mathcal{U}$, we work with $R[T]$, defined by $R[T] = \{t' \in \mathcal{U} \mid \ (t, t') \in R \text{ for } t \in T\}$. This form of application has an implicit universal quantification: it consists of all the elements of $\mathcal{U}$ that are related to some element of T. Hence, the intersection forms take care of the quantification present in the predicate semantics of the relative clauses without having to explicitly use it.

The predicate interpretation of the possessive subject clause is equivalent to the following intersection form:

$$[\![Poss]\!] \cap [\![Has]\!]^{-1} \left[[\![Verb]\!]^{-1}[[\![Obj]\!]] \cap [\![Subj]\!]\right]$$

The vector space interpretation of the above is as follows:

$$V_{[\![Poss]\!] \cap [\![Has]\!]^{-1}[[\![Verb]\!]^{-1}[[\![Obj]\!]] \cap [\![Subj]\!]]} = \mu(V_{[\![Poss]\!]}, V_{[\![Has]\!]^{-1}[[\![Verb]\!]^{-1}[[\![Obj]\!]] \cap [\![Subj]\!]]})$$

For the right hand side, we first compute the following:

$$\begin{aligned} V_{[\![Verb]\!]^{-1}[[\![Obj]\!]] \cap [\![Sbj]\!]} &= \mu(V_{[\![Verb]\!]^{-1}[[\![Obj]\!]]}, V_{[\![Sbj]\!]}) \\ = \mu(\sum_{ij} \overrightarrow{n}_i \langle \overrightarrow{n}_j \mid V_{[\![Obj]\!]} \rangle, V_{[\![Sbj]\!]}) &= \mu(\sum_{ij} \overrightarrow{n}_i \langle \overrightarrow{n}_j \mid \sum_{l \in L} \overrightarrow{n}_l \rangle, \sum_{k \in K} \overrightarrow{n}_k) \\ &= \sum_{ij, k \in K, l \in L} \delta_{ki} \overrightarrow{n}_i \delta_{jl} \end{aligned}$$

This term is a sum of $\overrightarrow{n}_i$'s whenever $k = i$ and $j = l$ and provides us with subjects that have been modified by the verb phrase Verb-Obj. We denote these by $\sum_a \overrightarrow{n}_a$. Next, we insert this term in $V_{\llbracket Has\rrbracket^{-1}[\llbracket Verb\rrbracket^{-1}[\llbracket Obj\rrbracket]\cap\llbracket Subj\rrbracket]}$ and proceed the computation as follows:

$$V_{\llbracket Has\rrbracket^{-1}[\llbracket Verb\rrbracket^{-1}[\llbracket Obj\rrbracket]\cap\llbracket Subj\rrbracket]} = \sum_{h'h''} \overrightarrow{n}_{h'} \langle \overrightarrow{n}_{h''} \mid \sum_a \overrightarrow{n}_a \rangle = \sum_{h'h''} \overrightarrow{n}_{h'} \delta_{h''a}$$

This term is a sum of $\overrightarrow{n}_{h'}$'s whenever $h'' = a$, that is whenever the individuals that are possessed have been modified by the Verb-Obj phrase. It provides us with the owners of such individuals. We denote them by $\sum_b \overrightarrow{n}_b$. We insert this in $\mu(V_{\llbracket Poss\rrbracket}, V_{\llbracket Has\rrbracket^{-1}[\llbracket Verb\rrbracket^{-1}[\llbracket Obj\rrbracket]\cap\llbracket Subj\rrbracket]})$ and finish the computation as follows:

$$\mu(V_{\llbracket Poss\rrbracket}, V_{\llbracket Has\rrbracket^{-1}[\llbracket Verb\rrbracket^{-1}[\llbracket Obj\rrbracket]\cap\llbracket Subj\rrbracket]}) = \mu(\sum_{h\in P} \overrightarrow{n}_h, \sum_b \overrightarrow{n}_b) = \sum_{h\in P} \delta_{hb} \overrightarrow{n}_h$$

This is a sum of $\overrightarrow{n}_h$'s whenever $h = v$, it provides us with the possessors that own individuals that have been modified by Verb-Obj. This is the same term as the one obtained in Section 8 for the case when all α terms are equal to 1. It provides us with the 0/1 truth-theoretic semantics of the subject possessive clause.

The case of the object possessive clause is computed similarly. Here, we need to compute the vector space interpretation of the following intersection form:

$$\llbracket Poss\rrbracket \cap \llbracket Has\rrbracket^{-1} \left[\llbracket Verb\rrbracket[\llbracket Subj\rrbracket] \cap \llbracket Obj\rrbracket\right]$$

In this case, the above provides us with the possessors that own individuals that have been modified by Subj-Verb phrases, equal to the term obtained in Section 8 for the case when α terms are equal to 1, hence providing us with the 0/1 truth-theoretic semantics of the clause.

Finally, it is easy to see that the decomposion of 'whose', as done in Section 7, provides us with the same intersection forms. The intersection form of 'Possessor that has Subject that Verb Object' is computed as follows. First we compute the interpretation of 'Subject that Verb Object', as demonstrated in previous work[24], this is as follows:

$$\llbracket Sbj\rrbracket \cap \llbracket Verb\rrbracket^{-1}[\llbracket Obj\rrbracket]$$

Next, we compute the interpretation of 'Possessor that has X' and obtain:

$$\llbracket Pos\rrbracket \cap \llbracket Has\rrbracket^{-1}[\llbracket X\rrbracket]$$

Finally, we substitute $[\![Sbj]\!] \cap [\![Verb]\!]^{-1}[[\![Obj]\!]]$ for X in the above and obtain:

$$[\![Pos]\!] \cap [\![Has]\!]^{-1}[[\![Sbj]\!] \cap [\![Verb]\!]^{1}[[\![Obj]\!]]]$$

This is the same as the intersection form of the possessive subject form. A similar computation shows that the intersection form of 'Possessor that has Object that Subject Verb' is the same as that of possessive object form.

10 Concrete Instantiations

In the model of Grefenstette and Sadrzadeh (2011a) [10], the meaning of a verb is taken to be "the degree to which the verb relates properties of its subjects to properties of its objects". This degree is computed by forming the sum of the tensor products of the subjects and objects of the verb across a corpus, where w ranges over instances of the verb:

$$\overline{\text{verb}} = \sum_w (\overrightarrow{\text{sbj}} \otimes \overrightarrow{\text{obj}})_w$$

The above is a matrix in $N \otimes N$. Since the verbs of this model do not have a sentence dimension, no ι map will be needed in computing the meanings of the classes containing them.

It remains to find a concrete matrix interpretation for $\overline{\text{'s}}$. For instance, to compute the meaning vector of "woman whose husband died", we need to know the "owners" (in a manner of speaking) of the husbands who have died, to be able to modify the vector of the possessor "woman" with it. Hence, $\overline{\text{'s}}$ should be the linear map that inputs a noun phrase and returns their owners. One way to construct this map for a noun phrase X is to sum over the nouns that have occurred in "noun's X". That is, for X the subject or object of the relative clause, we have

$$\overline{\text{'s}}(X) := \sum_i (\overrightarrow{\text{noun}})_i \qquad \text{for each "noun's } X\text{" in the corpus}$$

The abstract vectors corresponding to these diagrams are similar to the truth-theoretic case, with the difference that the vectors are populated from corpora and the scalar weights for noun vectors are not necessarily 1 or 0. For possessor, subject, and object noun context vectors computed from a corpus as follows:

$$\overrightarrow{\text{Possessor}} = \sum_h C_h \overrightarrow{n}_h \qquad \overrightarrow{\text{Subject}} = \sum_k C_k \overrightarrow{n}_k \qquad \overrightarrow{\text{Object}} = \sum_l C_l \overrightarrow{n}_l$$

and the verb and $\overline{\text{'s}}$ linear maps as follows:

$$\overline{\text{Verb}} = \sum_{ij} C_{ij} \overrightarrow{n}_i \otimes \overrightarrow{n}_j \qquad \overline{\text{'s}}(X) = \sum_{h'} C_{h'} \overrightarrow{n}_{h'}$$

The concrete meaning of a noun phrase modified by a possessive subject relative clause is as follows:

$$\mu \circ (1_N \otimes \overline{\text{'s}}) \left(\sum_h C_h \overrightarrow{n}_h \otimes \sum_{kl} C_{kl} C_k C_l \overrightarrow{n}_k \right) = \mu \left(\sum_h C_h \overrightarrow{n}_h \otimes \overline{\text{'s}} \left(\sum_{kl} C_{kl} C_k C_l \overrightarrow{n}_k \right) \right)$$
$$= \mu \left(\sum_h C_h \overrightarrow{n}_h \otimes \sum_{h'} C_{h'} \overrightarrow{n}_{h'} \right) = \sum_{hh'} C_h C_{h'} \delta_{hh'} \overrightarrow{n}_h$$

The difference with the truth-theoretic case is that the degrees of the truth of the verb and the coordinates of the subject, object, and possessor vectors are now obtained from a corpus and result in $C_h C_{h'}$.

To see how the above vector represents the meaning of the modified noun phrase, recall from Section 4 that the application of the μ map to the tensor product of two vectors is their component-wise multiplication, that is the above is equivalent to the following:

$$\sum_h C_h \overrightarrow{n}_h \odot \sum_{h'} C_{h'} \overrightarrow{n}_{h'}$$

Note that the second term of the above is the subject modified by the verb-object phrase. Hence the above vector modifies the possessor with the subjects that have been themselves modified by the verb-object phrase via point-wise multiplication. A similar result holds for the possessive object relative clause case.

As an example, suppose that N has two dimensions with basis vectors $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$, and consider the noun phrase "men whose dog bites cats". Define the vectors of "dog", "men", and "cats" as follows:

$$\overrightarrow{\text{dog}} = d_1 \overrightarrow{n}_1 + d_2 \overrightarrow{n}_2 \qquad \overrightarrow{\text{men}} = m_1 \overrightarrow{n}_1 + m_2 \overrightarrow{n}_2 \qquad \overrightarrow{\text{cats}} = c_1 \overrightarrow{n}_1 + c_2 \overrightarrow{n}_2$$

and the matrix of "bites" by:

$$b_{11} \overrightarrow{n}_1 \otimes \overrightarrow{n}_2 + b_{12} \overrightarrow{n}_1 \otimes \overrightarrow{n}_2 + b_{21} \overrightarrow{n}_2 \otimes \overrightarrow{n}_1 + b_{22} \overrightarrow{n}_2 \otimes \overrightarrow{n}_2$$

Then the meaning of the clause becomes:

$$\overrightarrow{\text{men}} \odot \overline{\text{'s}} \left(\overrightarrow{dogs} \quad \odot \quad \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \times \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \right)$$

Assuming that the basis vectors of the noun space represent properties of nouns, the meaning of “men whose dog bites cats” is a vector representing the properties of men, which have been modified (via multiplication) by the properties of owners of dogs which bite cats. Put another way, those properties of men which overlap with properties of owner’s of dogs that bite cats get accentuated. Similarly, for the clause “men whose dogs cats bite” we obtain the following linear algebraic form:

$$\overrightarrow{\text{men}} \odot \overline{\text{'s}} \left(\overrightarrow{dogs} \quad \odot \quad \begin{pmatrix} c_1 \; c_2 \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \right)$$

Here we have to transpose the column vector of ”cats” into a row vector (in other words transposing it) to be able to matrix-multiply it with the matrix of “bites”. The resulting vector contains properties of men which overlap with properties of dogs that cats bite.

11 A Toy Experiment

For demonstration purposes and in order to get an idea on how the data from a large scale corpus responds to the abstract computational methods developed here, we implement some of our constructions on a corpus and do a small-scale experiment with a toy natural language processing task. The corpus we use is the British National Corpus, from which we create a vector space whose basis vectors are its 10,000 most occurring lemmas. Our context is a window of 5 words. For each noun we built a vector, whose coordinates are computed using the ratio of the probability of the occurrence of the word in the context of the basis word to the probability of the occurrence of the word overall. These parameters are chosen based on the success of our previous experiments [10,11,15,14,13].

Our task consists of building vectors for nouns, verbs, and relative pronouns. For nouns, we used their context vectors built as described above. For verbs, we used the model of [10], based on the context vectors of their subjects and objects built as above. For relative pronouns, we apply two methods, described further on below.

The task we experiment with is an imaginary term/description classification task. It is based on the fact that relative clauses are often used to describe or provide extra information about words. For instance, the word ‘football’ may by described by the clause ‘game that boys like’ and the word ‘doll’ by the clause ‘toy that girls prefer’. For our experiment, we chose a set of words and manually described each of them by an appropriate relative clause. This data set is presented in the following table:

	Term	Description
1	emperor	person who rule empire
2	queen	woman who reign country
3	mammal	animal which give birth
4	plug	plastic which stop water
5	carnivore	animal who eat meat
6	vegetarian	person who prefer vegetable
7	doll	toy that girl prefer
8	football	game that boy like
9	skirt	garment that woman wear
10	widow	woman whose husband die
11	orphan	child whose parent die
12	teacher	person whose job educate children
13	comedian	artist whose joke entertain people
14	priest	clergy whose sermon people follow
15	commander	military whose order marine obey
16	clown	entertainer whose trick people enjoy

A preliminary goal would be to check how close the vector of the description is to the vector of its term. For instance, we obtained that the cosine of the angle between the vector of ‘football’ and the vector of its description was 0.61, and the cosine between the vector of ‘doll’ and its description was 0.50. As for the possessive cases, when we took $\overline{\text{’s}}$ to be the identity the cosine between ‘priest’ and its description was 0.51 and the cosine between ‘commander’ and its description was 0.40. These cosines decreased to 0.51 and 0.21 for the case when $\overline{\text{’s}}$ was computed by summing the possessors of the nouns ‘sermon‘ and ‘order’; this may be due to the rarity of the occurrence of the corresponding phrases, i.e. ‘noun’s sermon’ and ‘noun’s order’ in the corpus.

These cosines on their own may seem low and customary in the distributional models is not to consider them in isolation, but in relation to all the other cosines obtained in the experiment. So we follow previous work [15,14] and set the goal of our task to be classification; that is we compute for what percentage of the words, their vectors are closest to the vectors of their descriptions (a measure referred to by Mean Reciprocal Rank or MRR) and for what percentage of the descriptions, their vector are the closest to the vectors of their terms (a measure referred to by Accuracy).

For the MRR, we compared the vector of each term to the vectors of all the descriptions. From our 16 terms, the cosines of 12 of them were the closest to their own description; this was for the possessive model where $\overline{\text{’s}}$ was taken to be identity. The terms whose descriptions were not the closest to them were ‘plug, carnivore, vegetarian’, and ‘clown’. For the second possessive model, this number decreased to 9, with the inclusion of ‘orphan, comedian’, and ‘teacher’ in the set of bad terms. However, when the datasets for the possessive and non-possessive clauses were considered separately, that is, terms with possessive-clause descriptions were only compared with each other and terms with non-possessive descriptions were also only compared with each other, this number increased to 6 out of 9 for the non-possessive cases and 6 out of 7 for the possessive cases. In the latter category, the only word which was not the closest to its own description was ‘clown’. In the former category, the words whose descriptions were not the closest to their own were ‘plug, carnivore’, and ‘vegetarian’. In all of these cases, however, the correct description was the second closest to the word. Below are some of our term/description cosines (in the full dataset) for four of the good terms and two of the bad terms; this was in the first possessive model (the numbers for the second possessive model are slightly lower).

Term	Description	Cosine
football	game that boys like	0.62
	woman who reigns country	0.24
	toy that girls prefer	0.18
	woman whose husband died	0.18
priest	clergy whose sermon people follow	0.53
	woman who reigns country	0.45
	woman whose husband died	0.37
	person who rules empire	0.35
plug	toy that girls prefer	0.24
	plastic which stops water	0.22
	woman who reigns country	0.17
	game that boys like	0.17
clown	woman who reigns country	0.28
	toy that girls prefer	0.24
	woman whose husband died	0.24
	game that boys like	0.24

To have a comparison baseline, we also experimented with a multiplicative and an additive model. In these models, we built vectors for the relative clauses by multiplying/summing the vectors of the words in them in two ways: (1) treating the relative pronoun as noise and not considering it in the computation, (2) treating the relative pronoun as any other word and considering its context vector in the computation. As expected, the results turned out not to be significantly different (as the vectors of relative pronouns were dense and computing by them was similar to computing with the vector consisting of all 1's). For instance, the cosine for 'football' was 0.39 when the pronoun was not considered and 0.37 when it was considered; for 'clown', these numbers were 0.11 and 0.10. These models did bad on some of the good results of the Frobenius model, for example in the multiplicative model, the closest description to the term 'mammal' was 'animal that eats meat'. At the same time, they performed better on some of the bad terms of the Frobenius model, for example the description of 'plug' in the multiplicative model became the closest to it. The Mean Reciprocal Rank and Accuracy of the cosines between all the clauses (possessive and non-possessive) are presented in the following table. Overall, the Frobenius model with $\overline{\text{'s}} = Id$ did the best. All the models did better than the baseline, which was the vector of the head noun of the description, e.g. 'woman' for 'queen', 'animal' for 'mammal', 'plastic' for 'plug' and so on.

	Base	**Frob** $\overline{\text{'s}} = Id$	**Frob** $\overline{\text{'s}} = \sum_i (\overrightarrow{\text{noun}_1})_i$	**Mult** wo Rel. Pr.	**Mult** w Rel. Pr.	**Add** w/wo Rel. Pr.
MRR	0.70	0.82	0.71	0.78	0.76	0.75
Acc	0.56	0.75	0.56	0.62	0.62	0.62

The above numbers are not real representatives of the performance of the models; as the dataset is small and hand-made. Extending this task to a real one on a large automatically built dataset and doing more involved statistical analysis on the results requires a different venue; it constitutes work-in-progress.

12 Conclusion and Future Directions

In this paper, we reviewed the constructions of compact closed categories and Frobenius algebras and their diagrammatic calculi in an informal way. We also reviewed how compact closed categories can be applied to reason about the grammatical structure of language and the distributional meanings of the words and sentences of language. We then used the constructions of a Frobenius algebra over a vector space to extend the existing model to one that can also reason about possessive relative clauses. The Frobenius algebraic structure of the possessive relative pronouns show how the information of the head of the clause flows through the relative pronoun to the rest of the clause and how it interacts with the other words to produce a meaning for the whole clause. We instantiated these abstract constructions in a truth-theoretic model and also in a concrete vector space model built from a real corpus. For the former, we showed how this semantics coincides with the predicate semantics of these clauses and for the latter, we have presented a toy experiment and discussed a possible application to a natural language processing task. In a prequel to this paper, we used similar methods to reason about the subject and object relative pronouns.

For future work we aim to extend the toy example to a full blown example with a data set built from a corpus and investigate the role of our constructions. In this paper, our sentence space was the real line and we loosely interpreted the intermediate values of the unit interval as degrees or weights of truth. These values are reminiscent of the truth values in fuzzy logics. Developing a formal connections with fuzzy logic constitutes a possible future direction.

13 Acknowledgements

We would like to thank Dimitri Kartsaklis for his invaluable help in implementation and statistical analysis of the toy experiment. We would also like to thank Laura Rimell for her very helpful comments on an earlier draft of this paper. Stephen Clark was supported by ERC Starting Grant DisCoTex(30692). Bob Coecke and Stephen Clark are supported by EPSRC Grant EP/I037512/1. Mehrnoosh Sadrzadeh is supported by an EPSRC CAF grant EP/J002607/1.

References

1. S. Abramsky and B. Coecke. 2004. *A categorical semantics of quantum protocols.* Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 415–425. arXiv:quant-ph/0402130.
2. M. Baroni, R. Bernardi and R. Zamparelli. 2013. *Frege in Space: A Program for. Compositional Distributional. Semantics.* Linguistic Issues in Language Technologies (LiLT), A. Zaenen, B. Webber and M. Palmer (eds.). CSLI Publications. to appear.
3. M. Baroni and R. Zamparelli. 2010. *Nouns are vectors, adjectives are matrices.* Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP).
4. B. Coecke, D. Pavlovic and J. Vicary. 2008. *A new description of orthogonal bases.* Mathematical Structures in Computer Science **1**.

5. B. Coecke, E. Grefenstette, M. Sadrzadeh. 2013. *Lambek vs Lambek: Functorial Vector Space Semantics and String Diagrams for Lambek Calculus.* Annals of Pure and Applied Logic, to appear.
6. B. Coecke, M. Sadrzadeh and S. Clark. 2010. *Mathematical Foundations for Distributed Compositional Model of Meaning.* Lambek Festschrift. Linguistic Analysis **36**, 345–384. J. van Benthem, M. Moortgat and W. Buszkowski (eds.).
7. J. Curran. 2004. *From Distributional to Semantic Similarity.* PhD Thesis, University of Edinburgh.
8. J. R. Firth. 1957. *A synopsis of linguistic theory 1930-1955.* Studies in Linguistic Analysis.
9. E. Fitzgerald. 1859. *The Rubaiyat of Omar Khayyam.* Illustrated by A. Nath Tagore. Published by L. B. Hill, London.
10. E. Grefenstette and M. Sadrzadeh. 2011. *Experimental Support for a Categorical Compositional Distributional Model of Meaning.* Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP).
11. E. Grefenstette and M. Sadrzadeh. 2011. *Experimenting with Transitive Verbs in a DisCoCat.* Proceedings of Workshop on Geometrical models of natural language semantics (GEMS).
12. A. Joyal and R. Street. 1991. *The Geometry of Tensor Calculus I.* Advances in Mathematics, Vol. 88.
13. D. Kartsaklis and M. Sadrzadeh. 2013. *Prior Disambiguation of Word Tensors for Constructing Sentence Vectors.* Conference on Empirical Methods in Natural Language Processing (EMNLP).
14. D. Kartsaklis, M. Sadrzadeh, and S. Pulman. 2012. *A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments.* The 24th International International Conference on Computational Linguistics (CoLing).
15. D. Kartsaklis, M. Sadrzadeh, S. Pulman and B. Coecke. Compact Closed Categories and Frobenius Algebras for Natural Language Meaning. *Logic & Algebraic Structures in Quantum Computing & Information.* J. Chubb, A. Eskandarian, and V. Harizanov (eds.), Association for Symbolic Logic Lecture Notes in Logic, Cambridge University Press, to appear.
16. G. M. Kelly and M. L. Laplaza. 1980. *Coherence for compact closed categories.* Journal of Pure and Applied Algebra **19**, 193–213.
17. J. Lambek. 1958, *The mathematics of sentence structure*, American Mathematics Monthly 65.
18. J. Lambek. 1999, *Type grammar revisited*, Logical Aspects of Computational Linguistics 1582, A. Lecomte et al. (eds.), Springer LNAI.
19. J. Lambek. 2008. *From Word to Sentence.* Polimetrica, Milan.
20. J. Lambek and C. Casadio. 2006. *Computational algebraic approaches to natural language.* Polimetrica, Milan.
21. J. Mitchell and M. Lapata. 2008. *Vector-based models of semantic composition.* Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 236–244.
22. A. Preller and J. Lambek. 2007. *Free compact 2-categories.* Mathematical Structures in Computer Science **17**, 309–340.
23. R. Montague. 1974. *English as a formal language.* Formal Philosophy, 189–223.
24. M. Sadrzadeh, S. Clark, B. Coecke. 2013. *The Frobenius anatomy of word meanings I: subject and object relative pronouns.* arXive 1404.5278
25. I. A. Sag. 1997. *English Relative Clause Constructions.* JOURNAL OF LINGUISTICS, 431–484.

26. H. Schütze. 1998. *Automatic Word Sense Discrimination*. Computational Linguistics **24**, 97–123.
27. P. Selinger, 2010, *A survey of graphical languages for monoidal categories*, In: *New structures for physics*, B. Coecke (ed.), 275–337, Lecture Notes in Physics, Springer.
28. M. Steedman. 1996. *Surface Structure and Interpretation.* Linguistic Inquiry, Monograph Thirty. MIT Press.
29. L. .A. Zadeh. 1965. *Fuzzy sets.* Information and Control **8**, 338–353.