HAL open science

From Sentence to Concept

Anne Preller

▶ **To cite this version:**

Anne Preller. From Sentence to Concept. E. Grefenstette, C. Heunen, and M. Sadrzadeh. Categorical Information Flow in Physics and Linguistics, Oxford University Press, pp.247–271, 2012. lirmm-00635866v2

HAL Id: lirmm-00635866

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00635866v2

Submitted on 22 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Sentence to Concept*

Anne Preller
Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier †

Abstract

The compositional functional logical models of natural language are recast as compact closed categories. Composition is based on the geometrical representation of information flow characteristic for these categories. The functional logical interpretation of (strings of) words is carried over to projectors in a finite tensor product of 2-dimensional spaces such that the truth of a sentence is equivalent to the truth of the corresponding projector.

Examples include sentences with compound noun phrases involving quantifiers, adjectives and negation.

Keywords: Compact closed categories, quantum logic, concept spaces, two-sorted first order logic, compositional semantics, pregroup grammars, proof graphs, compound noun-phrases

1 Introduction

The present work attempts to relate two semantic representations of natural language, the functional logical models and the distributional vector models. The former deals with individuals and their properties, the latter with concepts and how they can be approximated.

Montague semantics and similar functional logical models for natural language are extensional and compositional. Meaningful expressions designate individuals, sets of individuals, functions from and to (sets of) individuals, truth-value functions and so on. The meaning of a grammatical string of words is computed from the meanings of the constituents using functional application or composition. This semantics requires prior grammatical analysis where every word contributes to the meaning, including 'noise' like negation, determiners, quantifiers, relative pronouns , etc.

The semantic vector models are based on the principle that the content of a word is measured in relation to the content of other words. They handle probabilistic estimations of concepts. Words, with the exclusion of 'noise', are represented by vectors in a finite dimensional space over the field of real numbers.

*chapter 9 of *Categorical Information Flow in Physics and Linguistics*, pp. 249 -271, Oxford University Press 2012

†E-mail: preller@lirmm.fr

Frequency counts of co-occurrences with other words determine the coordinates of a word. Semantic vector models excel in detecting similarity of words. They confound opposites.

Compositionality of vector semantics remains an open question and is subject of intensive research.

One approach to compositionality is quantum logic on the lattice of projectors of Hilbert spaces, see [Rijsbergen, 2004] for an overview oriented towards information retrieval or [Widdows, 2004] for a discussion of geometric properties of meaning. There is, however, no general algorithm that transforms a string of word into a vector respecting the logic. Another approach is composition of vectors by the tensor product, [Smolensky, 1988] invoking computational principles of cognitive science, or [Clark and Pulman, 2007] and [Clark et al., 2008] using syntactical analysis. Again, 'noise' is not included in composition.

The present work outlines a method that takes into account the logical content of 'noise' and transforms the compositional extensional representation into a conceptual representation. Both representations are based on biproduct dagger compact closed categories.

On one hand, a *concept space*, that is to say a tensor product of two-dimensional spaces, hosts both the words (concepts) and their probabilistic approximations. Concept spaces are the linguistic pendent to compound systems in quantum mechanics.

On the other hand, the logical functional representations of (strings of) words are also recast as vectors. These vectors are, roughly speaking, the names of the functions representing the words. Their construction involves syntactical analysis.

The examples presented here are analysed via pregroup grammars, [Lambek, 1999], based on compact bilinear logic, [Lambek, 1993]. The pregroup calculus is a simplification of the syntactic calculus by the same author, [Lambek, 1958]. Compact bilinear logic 'compacts' the higher order of categorial grammars into second order logic with general models, or equivalently, into two-sorted first order logic, [Benthem and Doets, 1983]. Moreover, the category of types and proofs of compact bilinear logic is the free compact *2*-category , [Preller and Lambek, 2007].

Categorical semantics in compact 2-categories for pregroup grammars was first proposed in [Preller, 2005], reformulated in [Preller, 2007] in terms of functions in two-sorted first order logic. This reformulation rests on the fact that sets and two-sorted functions form a compact closed category. The embedding of the category of two-sorted functions in the category of semimodules over a real interval, [Preller and Sadrzadeh, 2011], establishes the connection to semantic vector models.

The formulation of functional logic in a biproduct dagger compact closed category has been chosen to facilitate a comparison with quantum logic. It is based on [Abramsky and Coecke, 2004], casting quantum mechanics in the abstract setting of a biproduct dagger compact closed category. The result is an embedding of functional two-sorted first order logic into the lattice of projectors of concept spaces.

Section (2) introduces the semantical and syntactical categories. The category of two-sorted functions follows in Section 3 with its two-sorted first order logic. An embedding transfers them to an arbitrary bicategory dagger compact closed category.

The algorithm in Section 4, constructing meanings of strings from meanings of words, is based on syntactical analysis. Examples from natural language provide the graphs depicting the computation of the meaning by 'information flow'.

Concept spaces and the logical properties of their intrinsic projectors are investigated in Section 5. Subsection 5.1 deals with propositional logic and Subsection 5.2 with predicate logic. The truth preserving one-to-one correspondence between predicates on and intrinsic projectors of concept spaces is the subject of 5.3. This correspondence is used in Section 5.4 to compute the meaning of strings directly in concept spaces and to view arbitrary word vectors as a probabilistic approximation of concepts.

2 Notations, basic properties

Natural language processing involves both syntactical analysis and logical representation. Both can be formulated in the language of compact bicategories, also known as non-symmetric star autonomous categories.

Throughout this paper, the *syntactical category* is the compact bicategory freely generated by a 'basic' category. It is not symmetric.

The *semantic category* $\mathcal{C}$ is any biproduct dagger compact closed category in which all objects have a chosen finite basis, for example the category $\mathcal{RI}$ of free semimodules generated by finite sets over the lattice of the real interval $[0, 1]$.

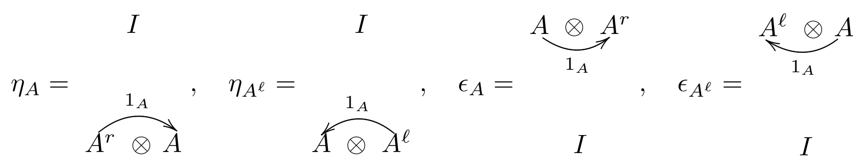
2.1 The syntactical category

The syntactical category $\mathcal{C}\mathit{2}(\mathcal{B})$ is the free compact bicategory generated by a category $\mathcal{B}$. It is notationally convenient to replace the canonical associativity and unit isomorphisms by identities, for example $A \otimes (B \otimes C) = (A \otimes B) \otimes C$, $A \otimes I = A = I \otimes A$. Strictly speaking, the bicategory is treated like a *2*-category.

Saying that a bicategory is compact means that every 1-cell A has a left adjoint A^ℓ and a right adjoint A^r. Let $\eta_A : I \longrightarrow A^r \otimes A$ be the unit and $\epsilon_A : A \otimes A^r \to I$ the counit for the right adjoint. Then $A \simeq A^{\ell r}$ is a right adjoint to A^ℓ so that $\eta_{A^\ell} : I \longrightarrow A \otimes A^\ell$ and $\epsilon_{A^\ell} : A^\ell \otimes A \longrightarrow I$ act as unit and counit for the left adjunction of A^ℓ to A.

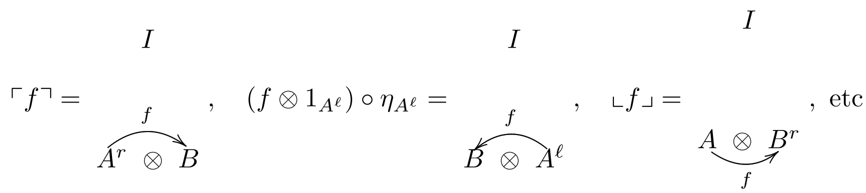
Starting with any 1-cell A that is an object of $\mathcal{B}$, one obtains the *iterated right* adjoints $A^\ell, A^{\ell\ell}, A^{\ell\ell\ell}, \ldots$ and the *iterated left* adjoints $A^r, A^{rr}, A^{rrr}, \ldots$ of A, but no mixed adjoints, because $A^{\ell r}$ and $A^{r\ell}$ are both isomorphic to A.

The *morphisms*, i.e. the 2-cells of $\mathcal{C}\mathit{2}(\mathcal{B})$, are represented by graphs where the vertices are objects of $\mathcal{B}$ and the oriented links are labelled by morphisms of $\mathcal{B}$. Examples are



NOTE: *graphs display the domain of the morphism above, the codomain below.*

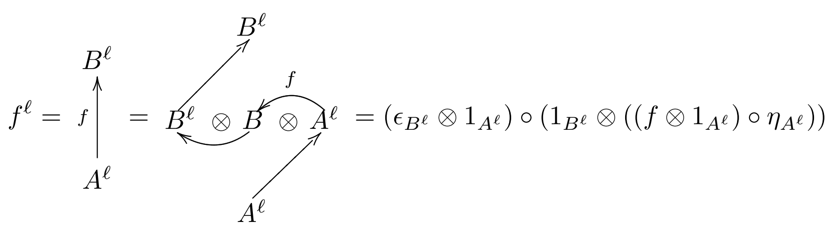
In the case where the label is an identity, it is in general omitted. An arbitrary $f : A \longrightarrow B \in \mathcal{B}$ also creates labels for links, for example



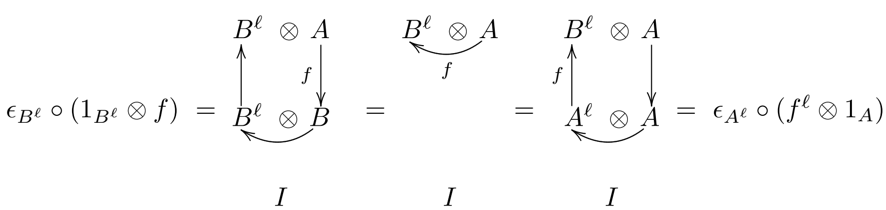
NOTE: *The labels of the links are morphisms of $\mathcal{B}$. Stripping the tail of the link of its adjoints, we obtain the domain of the label in $\mathcal{B}$. Similarly, the head without the adjoints is the codomain of the label.*

Composition of morphism is computed by connecting the graphs at the joint interface and walking paths, picking up and composing the labels in the order in which they are encountered.

Here are few examples involving $f : A \longrightarrow B$, $g : B \longrightarrow C$

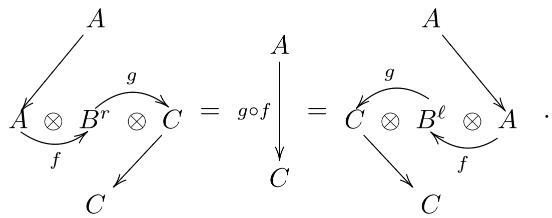


RECALL: *The domain of the morphism f^ℓ is the top line in the graph, the codomain is the bottom line, i.e. $f^\ell : B^\ell \longrightarrow A^\ell$.*



An equality of graphs is far easier to compute than the equality of the corresponding algebraic expressions. For example, the equality $(\llcorner f \lrcorner \otimes 1_C) \circ (1_A \otimes$

$\ulcorner g \urcorner) = g \circ f = (1_C \otimes (\epsilon_{B^\ell} \circ (1_{B^\ell} \otimes f))) \circ (((g \otimes 1_{B^\ell}) \circ \eta_{B^\ell}) \otimes 1_A) : A \longrightarrow C$ is proved thus



NOTE: *Links do not cross in the graphs of the syntactical category.*

The benefit of orienting and labelling links will become evident through the examples of natural language processing in Section 4.1.

2.2 The semantic category

The general definitions and properties of biproduct dagger compact closed categories can be found in [Selinger, 2007] or [Abramsky and Coecke, 2004]. Two semantic categories are specially tailored to natural language semantics, namely the category $\mathit{2}\mathcal{SF}$ of two-sorted functions, Subsection 3.1, and the category $\mathcal{RI}$ of free semimodules over the lattice of the real interval $[0,1]$ generated by finite sets.

Its importance to natural language processing resides in the fact that semantic vector models interpret words as vectors the coordinates of which are obtained by frequency counts of co-occurrences in context-windows. Without loss of generality, one may assume that the coordinates belong to the real interval $[0,1]$.

Recall that the linear order on the real numbers in $[0,1]$ induces a distributive and implication-complemented lattice structure on $[0,1]$, namely

$$\begin{gathered} \alpha \vee \beta = \max\{\alpha, \beta\} \text{ and } \alpha \wedge \beta = \min\{\alpha, \beta\} \\ \alpha \to \beta = \max\{\gamma \in I : \ \alpha \wedge \gamma \leq \beta\} \\ \neg\alpha = \alpha \to 0\,. \end{gathered}$$

This lattice is not Boolean, because $\neg\neg\alpha = 1 \neq \alpha$ for $0 < \alpha < 1$. The subset $\{0,1\}$, however, forms a Boolean algebra.

The lattice operations define a semiring structure on $[0,1]$ with neutral element 0 and unit 1 by

$$\alpha + \beta = \alpha \vee \beta \qquad \alpha \cdot \beta = \alpha \wedge \beta\,.$$

Basic properties of biproduct dagger compact closed categories

Objects of an arbitrary biproduct dagger compact closed category are called *spaces*, morphisms *linear maps*. A morphism $v : I \longrightarrow V$, where I is the unit of the tensor product, is called a *vector of* V. Write $v \in V$ for vectors $v : I \longrightarrow V$ and $f(v) \in W$ for $f \circ v : I \longrightarrow W$, where $f : V \longrightarrow W$.

Vectors $b_1, \ldots, b_n$ of V form a *basis* of V if every vector of V can be written in a unique way as a linear combination of the vectors $b_1, \ldots, b_n$. A space is n-dimensional if it has a basis of cardinality n. The dimension is unique. A space with chosen basis $B = b_1, \ldots, b_n$ is denoted V_B.

All spaces are assumed to be finite dimensional from now on.

Linear maps identify with matrices such that multiplication of matrices corresponds to composition of maps. Indeed, let $A = a_1, \ldots, a_m$ and $B = b_1, \ldots, b_n$. A linear map $f : V_A \longrightarrow V_B$ is determined by its values on the basis vectors $a_1, \ldots, a_m \in A$ and is characterized by the matrix (ϕ_{ij}) where $\phi_{ij} \in I$ is the i-th coordinate of $f(a_j) = \sum_{i=1}^{n} \phi_{ij} b_i$. Such matrices can be identified with vectors in $V_A \otimes W_B$. The basis vectors of the latter are $a_j \otimes b_i$, for $j = 1, \ldots, m$ and $i = 1, \ldots, n$.

The *inner product* $\langle v|w\rangle$ of vectors $v = \sum_{j=1}^{m} \alpha_j a_j$ and $u = \sum_{j=1}^{m} \beta_j a_j$ in $\mathcal{RI}$ is given by

$$\langle v|u\rangle = \langle \sum_{j=1}^{m} \alpha_j a_j | \sum_{j=1}^{m} \beta_j a_j \rangle = \sum_{j=1}^{m} \alpha_j^{\dagger} \beta_j \, .$$

Vectors are *orthogonal* if $\langle v|u\rangle = 0$. In the case of $\mathcal{RI}$, we have $\alpha = \alpha^{\dagger}$ for all scalars $\alpha \in [0,1]$. Hence vectors with coordinates in $[0,1]$ are orthogonal in $\mathcal{RI}$ exactly when they are orthogonal in the category of Hilbert spaces.

The category of semimodules $\mathcal{RI}$ is a biproduct dagger compact closed category with monoidal unit $I = [0,1]$. Every object V of $\mathcal{RI}$ has a unique finite basis A, which we express by $V = V_A$. It is its own adjoint, $V_A = V_A^*$. The unit $\eta_{V_A} : I \longrightarrow V_A \otimes V_A$ and counit $\epsilon_{V_A} : V_A \otimes V_A \to I$ of the adjunction are given by

$$\eta_{V_A}(1) = \sum_{a \in A} a \otimes a \qquad \epsilon_{V_A}(a \otimes b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{else} \end{cases} = \langle a|b\rangle \, .$$

The name and coname of $f : V_A \longrightarrow V_B$ are defined by

$$\begin{array}{l} \ulcorner f \urcorner(1) = \sum_{a \in A} a \otimes f(a) \\ \llcorner f \lrcorner(a \otimes b) = \langle f(a)|b\rangle \text{ for } a \in A,\, b \in B \, . \end{array}$$

By definition, $V_A = V_A^{\dagger}$. The adjoint of $f : V_A \longrightarrow V_B$ is the morphism $f^* = f^{\dagger}$ induced by the transpose of the matrix of f.

The logic of vectors

Definition 1 (**Boolean vector**). *Let $\mathcal{C}$ be any semantic category, $0 = 0_{0I} : 0 \to I$, $1 = 1_I : I \to I$ and V_B any space in $\mathcal{C}$. A vector $v = \sum_{i=1}^{n} \alpha_i b_i \in V_B$ is* Boolean *if $\alpha_i \in \{0,1\}$, for $i = 1, \ldots, n$.*

The connectives $\neg$, $\wedge$ etc. are operators on the set $\{0,1\}$ satisfying

$$\neg 0 = 1, \neg 1 = 0 \text{ and } 1 \wedge 1 = 1, 0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0 \text{ etc.}$$

They lift to the Boolean vectors, where they are defined coordinate by coordinate

$$\neg \sum_{i=1}^{n} \alpha_i b_i = \sum_{i=1}^{n} (\neg \alpha_i) b_i ,\ (\sum_i \alpha_i b_i) \wedge (\sum_j \beta_j b_j) = \sum_i (\alpha_i \wedge \beta_i) b_i \text{ etc.} \tag{1}$$

and induce a partial order by the postulate

$$v \leq w \text{ if and only if } v \wedge w = v\,.$$

We have

1. the Boolean vectors together with the logical vector connectives form a Boolean algebra
2. every Boolean v defines a unique subset $K \subseteq B$ such that $v = \sum_{b \in K} b$ and vice versa
3. the *null vector* $\overrightarrow{0}$ (with coordinates all equal to 0) is the smallest and the *full vector* $\overrightarrow{1}$ (with coordinates all equal to 1) the largest vector.

Lemma 1. *If V_B is a space of $\mathcal{RI}$ the equalities (1) define a distributive, implication complemented lattice structure on V_B such that the following equivalences hold*

$$\neg\neg v = v \Longleftrightarrow v \vee \neg v = \overrightarrow{1} \Longleftrightarrow \textit{ the coordinates of } v \textit{ are } 0 \textit{ or } 1\,. \tag{2}$$

Moreover, vector conjunction is the linear map $\wedge : V_B \otimes V_B \longrightarrow V_B$ defined on the basis vectors by

$$\wedge(b \otimes b) = b, \quad \wedge(b \otimes b') = \overrightarrow{0},\ \textit{for } b \neq b' \in B\,.$$

The logic of projectors

Let $\mathcal{C}$ be any semantic category and E an n-dimensional space with chosen basis $B = b_1, \ldots, b_n$.

Recall that a morphism $p : E \longrightarrow E$ is a *projector* if it is idempotent and self-adjoint

$$p \circ p = p, \quad p^\dagger = p\,.$$

In $\mathcal{RI}$, the latter equality means that the matrix of p is symmetric.

A projector p of a Hilbert space determines a subspace, namely the set of vectors invariant under p

$$E_p = \{w \,:\, w = p(w)\} = p(E)\,.$$

These subspaces are in one-to-one correspondence with the projectors. Hence the quantum connectives are defined on the set of projectors/subspaces, see [Rijsbergen, 2004], by

$$\neg E_p = E_p^\perp,\ E_p \vee E_q = E_p + E_q,\ E_p \wedge E_q = E_{p \circ q},$$
$$E_p \to E_q = \{u \,:\, q(p(u)) = p(u)\}\,.$$

The quantum connectives induce a not necessarily distributive lattice structure on the set of projectors such that $p \leq q$ is equivalent to $p \wedge q = q$. Thinking of projectors as propositions and of 1_E as the true proposition, the equality $p \rightarrow q = 1_E$ is read as 'p implies q'.

This approach is not possible in an arbitrary semantic category. Any two-dimensional space $V_{\{a_1,a_2\}}$ of $\mathcal{RI}$ has subspaces that are not image of any projector. An example is the span of the vectors $u = \alpha a_1$, $v = \beta a_2$, $w = \gamma a_1 + \beta a_2$, where $0 < \beta < \alpha < \gamma \leq 1$. We need a property that connects subspaces and projectors in an arbitrary semantic category.

Definition 2 (**Intrinsic morphism**). *A linear map of $\mathcal{C}$ is* intrinsic *if it sends every basis vector to a basis vector or to the null vector.*

Intrinsic linear maps are closed under composition.

Lemma 2. *A projector $p : V_B \longrightarrow V_B$ is intrinsic if and only if*

$$p(b_i) = b_i \text{ or } p(b_i) = \overrightarrow{0}\,, \text{ for } i = 1, \ldots, n\,. \tag{3}$$

Proof. Let p be an intrinsic projector and (π_{ij}) its matrix. This matrix is symmetric, because p is self-adjoint and the entries π_{ij} are 0 or 1.

We must show that $p(b_k) = b_l$ implies $k = l$. From $p(b_k) = b_l$ follows $p(b_l) = b_l$, because p is idempotent. The latter equality implies $\pi_{ll} = 1$ and $\pi_{il} = 0$ for $i \neq l$. Moreover, $p(b_k) = b_l$ implies $\pi_{lk} = 1$ and $\pi_{ik} = 0$ for $i \neq l$. By symmetry, $\pi_{kl} = 1$. Hence $k = l$. □

Hence, every intrinsic projector has the form $\sum_{k \in K} |b_k\rangle\langle b_k|$ where $K \subseteq \{1, \ldots, n\}$, where $|b_i\rangle\langle b_i|$ maps b_i to itself and every other basis vector to the null vector. The composition $q \circ p$ of intrinsic projectors $p : V_B \longrightarrow V_B$ and $q : V_B \longrightarrow V_B$ is again an intrinsic projector satisfying

$$(q \circ p)(x) = q(x) \wedge p(x), \text{ for all } x \in B\,.$$

Intrinsic projectors are in one-to-one correspondence with Boolean vectors. Indeed, let $v = \alpha_1 b_1 + \cdots + \alpha_n b_n$ be a vector of V_B. Define the linear map $p_v : V_B \longrightarrow V_B$ by its values on the basis vectors

$$p_v(b_i) = \alpha_i b_i, \text{ for } i = 1, \ldots, n\,. \tag{4}$$

If v is Boolean some obvious properties are

1. p_v is an intrinsic projector
2. $p_v(w) = v \wedge w$ for every Boolean vector w; in particular $p_v(\overrightarrow{1}) = v$
3. the map $v \mapsto p_v$ is one-to-one
4. $p_{\overrightarrow{1}} = 1_{V_B}$

Lemma 3. *For every intrinsic projector p there is a Boolean vector v such that $p = p_v$. For every Boolean vector v, the subspace E_{p_v} of vectors invariant under p_v coincides with the subspace generated by the basis vectors b_i satisfying $b_i \leq v$. Every subspace generated by a subset of basis vectors is the invariance space of an intrinsic projector.*

Proof. Let p be an intrinsic projector. Define

$$K = \{k : p(b_k) = b_k \,\&\, 1 \le k \le n\} \text{ and } v = \sum_{k \in K} b_k \,.$$

Then $p(b_i) = p_v(b_i)$, for $i = 1, \dots, n$. Hence the map $v \mapsto p_v$ is onto the set of intrinsic projectors. Moreover, E_p is generated by the set of basis vectors $\{b_k : k \in K\}$ and $b_i \le v$ if and only if $i \in K$. □

Theorem 1. *The map $v \mapsto p_v$ is a negation preserving lattice isomorphism $\mathcal{H}$ from the Boolean vectors onto the intrinsic projectors of V_B.*

Proof. Writing $E_v = E_{p_v}$, we must prove that

$$E_v{}^{\perp} = E_{\neg v},\; E_v \vee E_w = E_{v \vee w},\; p_v \circ p_w = p_{v \wedge w},\; E_v \to E_w = E_{v \to w}\,. \tag{5}$$

The first three equalities follow immediately from Property 2. and Property 4. above.

To prove the last equality, let $v = \sum_{i=1}^n \alpha_i b_i$, $w = \sum_{i=1}^n \beta_i b_i$ be Boolean vectors and $u = \sum_{i=1}^n \gamma_i b_i$ be any vector. By definition, $v \to w = \sum_{i=1}^n (\neg\alpha_i \vee \beta_i) b_i$. On one hand,

$$\begin{aligned} u \in E_{v \to w} \quad &\Leftrightarrow u = p_{v \to w}(u) \\ &\Leftrightarrow \gamma_i = (\neg\alpha_i \vee \beta_i)\gamma_i, \text{ for } i = 1, \dots, n\,. \end{aligned}$$

On the other hand, recall that $E_v \to E_w = \{u : p_w(p_v(u)) = p_v(u)\}$. Hence,

$$u \in E_v \to E_w \Leftrightarrow \alpha_i \gamma_i = \alpha_i \beta_i \gamma_i \text{ for } i = 1, \dots, n\,.$$

The equalities $\gamma_i = (\neg\alpha_i \vee \beta_i)\gamma_i$ and $\alpha_i\gamma_i = \alpha_i\beta_i\gamma_i$ both hold trivially if $\alpha_i = 0$ or $\beta_i = 1$. In the case where $\alpha_i = 1$ and $\beta_i = 0$ they are both equivalent to the equality $\gamma_i = 0$. □

It follows that the sublattice of intrinsic projectors is Boolean. Moreover, intrinsic projectors are monotone increasing on Boolean vectors.

The linear map defined by Equalities (4) is a projector in $\mathcal{RI}$ for an arbitrary vector and Equalities (5) hold for all vectors.

3 Two-sorted first order logic in compact closed categories

The relevance of two-sorted first order logic for natural language resides in the fact that it is equivalent to second order logic with general models, see [Benthem and Doets, 1983], and a wide spread belief that second order logic suffices for natural language semantics.

3.1 The category of two-sorted functions

Two-sorted functions are tailored to natural language, because they accept elements (sort 1) as well as sets (sort 2) as arguments. In a similar way, verbs accept both singulars and plurals. Functions in two-sorted first order logic were first used in [Preller, 2007]. The presentation below follows [Preller and Sadrzadeh, 2011].

Definition 3 (**Two-sorted function**). *A function $f : A \longrightarrow B$ is* two-sorted *if it maps elements and subsets of A to elements or subsets of B and satisfies*

$$\begin{array}{rcl} f(\{a\}) &=& f(a) \text{ for } a \in A \\ f(\emptyset) &=& \emptyset \\ f(X \cup Y) &=& f(X) \cup f(Y) \text{ for } X, Y \subseteq A\,. \end{array} \tag{6}$$

Obviously, a two-sorted function defined on a finite set is determined by its values on the elements. Examples are the *two-sorted identity* 1_A and the *two-sorted diagonal* 2_A defined by

$$1_A(a) = a, \text{ for } a \in A \qquad 2_A(a) = \langle a, a \rangle, \text{ for } a \in A\,.$$

Lemma 4. *The category $\mathcal{2SF}$ of finite sets and two-sorted functions is a dagger biproduct compact closed category.*

Proof. The biproduct is the disjoint union of sets. Hence $V_A = A$ for every finite set A.

The monoidal structure is given by the cartesian product of sets. A two-sorted notation for the Cartesian product brings the same notational advantages as the tensor product

$$\begin{array}{ll} a \times_2 b & = \langle a, b \rangle \\ a \times_2 B & = \{a\} \times B \\ A \times_2 b & = A \times \{b\} \\ A \times_2 B & = A \times B\,. \end{array}$$

The two-sorted product $f \times_2 g : A \times_2 C \longrightarrow B \times_2 D$ for $f : A \longrightarrow B$ and $g : C \to D$ is determined by it values on the elements of $A \times_2 C$, namely

$$(f \times_2 g)(a \times_2 b) = f(a) \times_2 g(b), \text{ for } a \in A, b \in C\,.$$

The monoidal unit is a distinguished singleton set $I = \{a_0\}$.

Every object is its own adjoint, $A = A^* = A^\dagger$, the unit $\eta_A : I \longrightarrow A \times A$ and counit $\epsilon_A : A \times A \to I$ of the adjunction are given by

$$\eta_A(a_0) = \{a \times_2 a \,:\, a \in A\} \qquad \epsilon_A(a \times_2 b) = \begin{cases} a_0 & \text{if } a = b \\ \emptyset & \text{else} \end{cases}$$

The name, coname, dual and dagger of $f : A \longrightarrow B$ are

$$\begin{array}{l} \ulcorner f \urcorner(a_0) = \{a \times_2 b \,:\, f(a) = b \text{ or } b \in f(a),\, a \in A,\, b \in B\} \\ \llcorner f \lrcorner(a \times_2 b) = \begin{cases} a_0 & \text{if } f(a) = b \text{ or } b \in f(a) \\ \emptyset & \text{else} \end{cases} \\ f^*(b) = \{a \in A \,:\, f(a) = b \text{ or } b \in f(a)\} = f^\dagger(b)\,. \end{array}$$

□

The category $\mathcal{2SF}$ has an abundance of projectors. Here is one, which is not a projector in the category of Hilbert spaces.

Example 1. *A two-sorted projector that is not intrinsic.*

The two-sorted function $p : \{a, b\} \longrightarrow \{a, b\}$ defined by

$$p(a) = \{a, b\} \text{ and } p(b) = \{a, b\}$$

is a projector in $\mathcal{2SF}$. The corresponding matrix is

$$(\pi_{ij}) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} .$$

The same matrix induces an endomorphism in any semantic category. This endomorphism is again a projector in $\mathcal{RI}$, but not in general when addition is not idempotent.

Luckily, the semantics of natural language only involves projectors that live in every semantic category, namely the intrinsic projectors. The characterization (3) of intrinsic projectors is equivalent in $\mathcal{2SF}$ to

$$p(Y) = \{x \in B \,:\, p(x) = x\} \cap Y, \text{ for every } Y \subseteq B\,. \tag{7}$$

3.2 The embedding

Given a set $A = \{a_1, \ldots, a_m\}$, the map $\mathcal{J}_A$ is defined for elements $a \in A$ and subsets $X \subseteq A$ thus

$$\mathcal{J}_A(a) = a \in V_A, \quad \mathcal{J}_A(X) = \sum_{a \in X} a \in V_A\,. \tag{8}$$

The map $\mathcal{J}_A$ has an inverse $\mathcal{J}_A^{-1}$ that sends a sum of distinct basis vectors in V_A to the corresponding subset of A, i.e. for $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$

$$\mathcal{J}_A^{-1}(\sum_{j=1,\ldots,k} a_{i_j}) = \{a_{i_1}, \ldots, a_{i_k}\}\ . \tag{9}$$

The following holds for every Boolean vector $v \in V_A$ and every $X \subseteq A$

$$\begin{aligned} \mathcal{J}_A^{-1} \circ \mathcal{J}_A(X) &= X \\ \mathcal{J}_A \circ \mathcal{J}_A^{-1}(v) &= v\,. \end{aligned}$$

In fact, $\mathcal{J}_A$ is an isomorphism of Boolean algebras that maps subsets of A onto Boolean vectors of V_A. Indeed, $\mathcal{J}_A$ commutes with the logical connectives

$$\begin{aligned} \mathcal{J}_A(X \cup Y) &= \mathcal{J}_A(X) \vee \mathcal{J}_A(Y) \\ \mathcal{J}_A(X \cap Y) &= \mathcal{J}_A(X) \wedge \mathcal{J}_A(Y)\ , \text{ for } X, Y \subseteq A \\ \mathcal{J}_A(A \setminus X) &= \neg \mathcal{J}_A(X) \end{aligned}$$

Finally, for any finite set A let

$$\mathcal{J}(A) = V_A$$

and for any two-sorted function $f : A \longrightarrow B$ let $\mathcal{J}(f) : V_A \longrightarrow V_B$ be the linear map satisfying

$$\mathcal{J}(f)(a)=\mathcal{J}_B(f(a)) \text{ for } a \in A\,. \tag{10}$$

The restriction of $\mathcal{J}(f) : V_A \longrightarrow V_B$ to Boolean vectors is a Boolean homomorphism in $\mathcal{RI}$, because vector disjunction coincides with vector addition in this category.

Definitions (8) -(10) make sense in the category of finite dimensional Hilbert spaces as well. The maps $\mathcal{J}_A$ are Boolean isomorphisms, but the linear map $\mathcal{J}(f)$ is not a Boolean homomorphism when restricted to Boolean vectors.

The following *Embedding Lemma* is one of the reasons why $\mathcal{RI}$ is especially appropriate for natural language semantics.

Lemma 5 (Embedding Lemma)**.** *The map $\mathcal{J} : 2\mathcal{SF} \to \mathcal{RI}$ is a one-to-one functor that preserves the biproduct dagger compact closed structures and the logical connectives.*

The restriction of $\mathcal{J}$ to the subcategory of intrinsic maps is a functor in an arbitrary semantic category.

Proof. The proof is straight forward and essentially that given in [Preller and Sadrzadeh, 2011]. □

Only the equality $\mathcal{J}(g \circ f) = \mathcal{J}(g) \circ \mathcal{J}(f)$, which holds in $\mathcal{RI}$, does not hold in an arbitrary semantic category unless both f and g are intrinsic.

Indeed, let $f'(a) = f(a)$ if $f(a)$ is a set and $f'(a) = \{f(a)\}$ if $f(a)$ is an element. Define $g'(b)$ similarly and let $K = \bigcup_{b\in f'(a)} g'(b)$. Then on one hand,

$$(g \circ f)(a) = \bigcup_{b\in f'(a)} \bigcup_{c\in g'(b)} \{c\} = \bigcup_{c\in K} \{c\} \text{ and therefore } \mathcal{J}(g \circ f)(a) = \sum_{c\in K} c\,.$$

On the other hand, $\mathcal{J}(f)(a) = \mathcal{J}_B(f(a)) = \sum_{b\in f'(a)} b$. Hence

$$\mathcal{J}(g) \circ \mathcal{J}(f)(a) = \sum_{b\in f'(a)} \mathcal{J}(g)(b) = \sum_{b\in f'(a)} \sum_{c\in g'(b)} c = \sum_{c\in K} c\,. \tag{11}$$

The rightmost equality of (11) holds in $\mathcal{RI}$, because vector addition is idempotent. This is not the case in an arbitrary semantic category, Example 1, unless f and g are intrinsic. If they are intrinsic, however, the sets $f'(a)$ and $g'(b)$ are either empty or singleton sets and the equalities (11) above hold. Hence again $\mathcal{J}(g) \circ \mathcal{J}(f) = \mathcal{J}(g \circ f)$.

3.3 Two-sorted truth

Let S be a fixed two-dimensional space of the semantic category $\mathcal{C}$ with basis vectors $\top$ and $\bot$. Think of $\top$ as 'true' and of $\bot$ as 'false'. The Boolean vectors of S are called two-sorted *truth-values.* The basis vectors are of sort 1, the null-vector $\overrightarrow{0}$ and the full vector $\overrightarrow{1} = \top + \bot$ of sort 2. The null-vector stands for 'neither true nor false' and $\top + \bot$ for 'partly true and partly false'. Two-sorted truth-values reflect second order properties of natural language, Subsection 4.3 provides some examples.

The two-sorted connectives on S are linear maps, which are determined by their values on basis vectors. They are intrinsic and thus live in every compact closed category with biproducts.

The *two-sorted conjunction* $\mathtt{and}_S : S \otimes S \longrightarrow S$, the *two-sorted disjunction* $\mathtt{or}_S : S \otimes S \longrightarrow S$ and the *two-sorted negation* $\mathtt{not}_S : S \longrightarrow S$ are given by

$$\begin{gathered}\mathtt{and}_S(\top \otimes \top) = \top,\ \mathtt{and}_S(\bot \otimes \top) = \mathtt{and}_S(\top \otimes \bot) = \mathtt{and}_S(\bot \otimes \bot) = \bot \\ \mathtt{or}_S(\bot \otimes \bot) = \bot,\ \mathtt{or}_S(\bot \otimes \top) = \mathtt{or}_S(\top \otimes \bot) = \mathtt{or}_S(\top \otimes \top) = \top \\ \mathtt{not}_S(\top) = \bot,\ \mathtt{not}_S(\bot) = \top\,.\end{gathered}$$

The two-sorted connectives are Boolean operators on S, because

$$\begin{gathered}\mathtt{not}_S \circ \mathtt{not}_S = 1_S \\ \mathtt{or}_S \circ (\mathtt{not}_S \otimes \mathtt{not}_S) = \mathtt{not}_S \circ \mathtt{and}_S\,.\end{gathered} \tag{12}$$

Two-sorted negation coincides with vector negation on basis vectors, but they are not identical. In fact, $\mathtt{not}_S$ is the symmetry isomorphism that exchanges the two basis vectors of S, whereas vector negation is not even an endomorphism of S

$$\begin{gathered}\mathtt{not}_S(\top) = \neg\top,\ \mathtt{not}_S(\bot) = \neg\bot,\ \mathtt{not}_S(\overrightarrow{0}) = \overrightarrow{0}, \\ \text{whereas } \neg\overrightarrow{0} = \overrightarrow{1}\,.\end{gathered}$$

In general, the two-sorted connectives differ from the vector connectives, even on basis vectors. For example

$$\mathtt{and}_S(\bot \otimes \top) = \bot,\ \text{whereas } \bot \wedge \top = \overrightarrow{0}\,.$$

Natural language prefers the two-sorted connectives on S, a fact relected by the notation.

3.4 Two-sorted predicates

Let E be any object of $\mathcal{C}$, think of the basis vectors of E as 'individuals'. Abbreviate the n-fold tensor product of E by $E^n = E \otimes \ldots \otimes E$.

Definition 4 (**Two-sorted predicate**)**.** *A* two-sorted predicate *is an intrinsic morphism with codomain S.*

A two-sorted predicate on E *is a two-sorted predicate that maps basis vectors of E to basis vectors of S.*

*An n-*ary two-sorted predicate on E *is a* two-sorted predicate on E^n.

Lemma 6. *The n-ary predicates on E together with the two-sorted connectives form a Boolean algebra.*

More precisely, assume that $p : E^n \longrightarrow S$ and $r : E^n \longrightarrow S$ are n-ary two-sorted predicates on E. Then the linear maps

$$\mathtt{not}_S \circ p,\ \mathtt{and}_S \circ (p \otimes r),\ \mathtt{or}_S \circ (p \otimes r),\ \mathtt{and}_S \circ (p \otimes r) \circ 2_{E^n},\ \mathtt{or}_S \circ (p \otimes r) \circ 2_{E^n}$$

are again two-sorted predicates on E such that

$$\begin{aligned} &\mathtt{not}_S \circ \mathtt{not}_S \circ p = p \\ &\mathtt{not}_S \circ \mathtt{and}_S \circ (p \otimes r) = \mathtt{or}_S \circ ((\mathtt{not}_S \circ p) \otimes (\mathtt{not}_S \circ r)) \,. \end{aligned} \tag{13}$$

Proof. It suffices to check the equalities (13) for basis vectors, which follows from (12). □

Two-sorted predicates have two possible values for individuals. For sets, however, they have a wider range of values. The following lemma tells us when.

Lemma 7. [Fundamental Property] *Let $p : V_B \longrightarrow S$ be a two-sorted predicate on V_B and $A = \{b_{i_1}, \ldots, b_{i_k}\}$ a subset of k basis vectors. Then there is a non-negative integer $k_1 \leq k$ such that*

$$p(b_{i_1} + \cdots + b_{i_k}) = k_1 \cdot \top + (k - k_1) \cdot \bot$$

holds. In particular, the following equivalences hold in the categories $2\mathcal{SF}$ and $\mathcal{RI}$

$$\begin{aligned} &p(\textstyle\sum_{x \in A} x) = \overrightarrow{0} \Leftrightarrow A = \emptyset \\ &p(\textstyle\sum_{x \in A} x) = \top \Leftrightarrow p(x) = \top \text{ for all } x \in A \text{ and } A \neq \emptyset \\ &p(\textstyle\sum_{x \in A} x) = \bot \Leftrightarrow p(x) = \bot \text{ for all } x \in A \text{ and } A \neq \emptyset \\ &p(\textstyle\sum_{x \in A} x) = \overrightarrow{1} \Leftrightarrow p(x) = \top \text{ and } p(y) = \bot \text{ for some } x, y \in A \,. \end{aligned} \tag{14}$$

Proof. Let k_1 be the number of elements of the set $A_1 = \{x \in A : p(x) = \top\}$. Then $k - k_1$ is the number of elements of $A_2 = \{x \in A : p(x) = \bot\}$. By linearity, $p(\sum_{x \in A} x) = \sum_{x \in A_1} p(x) + \sum_{x \in A_2} p(x) = k_1 \cdot \top + (k - k_1) \cdot \bot$. The particular case follows because addition is idempotent. □

In Hilbert spaces, the value a two-sorted predicate assigns to the vector $\sum_{x \in A} x$ is the result of two 'counts'. One counts the elements of A for which the predicate is true, the other one those for which the predicate is false. It suffices to divide by the number of elements k in the set A, to obtain a frequency distribution. The step from two-sorted predicates to probabilities is minimal.

Corollary 1. *The following equivalences hold in the categories $2\mathcal{SF}$ and $\mathcal{RI}$, for any element x and any subset Y of B*

$$p(x) = \bot \Leftrightarrow p(x) \neq \top \Leftrightarrow \mathtt{not}_S(p(x)) = \top$$

$$p(Y) = \bot \Leftrightarrow \mathtt{not}_S(p(Y)) = \top \,. \tag{15}$$

In general, however, $p(Y) \neq \top$ does not imply $p(Y) = \bot$.

Proof. The equivalence $p(Y) = \bot \Leftrightarrow \mathtt{not}_S(p(Y)) = \top$ follows from the definition of the two-sorted negation.

By the fundamental property, $p(Y) = \bot$ is equivalent to $\forall_x(x \in Y \Rightarrow p(x) = \bot)$ for a non-empty set Y. Now assume that Y has two distinct elements x and y and that $p(x) = \top$ and $p(y) = \bot$. Then $p(Y) = \{\top, \bot\} = \overrightarrow{1} \neq \bot$. ☐

In the context of natural language, different font shapes help to distinguish the set $X \subseteq B$ from the vector $\mathcal{J}_B(X)$, namely italic font for the former and typewriter font for the latter. For example, if $Bank \subseteq B$

$$\mathtt{bank} = \sum_{x \in Bank} x = \mathcal{J}_B(Bank)\,.$$

The same applies when distinguishing a one-sorted predicate on B and the corresponding two-sorted predicate on V_B. For example, the one-sorted predicate *Rich* corresponding to $\mathtt{rich} : V_B \longrightarrow S$ satisfies

$$x \in Rich \Leftrightarrow Rich(x) \Leftrightarrow \mathtt{rich}(x) = \top, \text{ for all } x \in B\,.$$

Here are a few examples how two-sorted predicates work in an arbitrary semantic category.

Under the assumption that a noun designates a non-empty set of individuals, the following equivalences hold in $\mathcal{2SF}$ and $\mathcal{RI}$. The proofs are straight forward via the Fundamental Property (14)

Example 2. *The following are equivalent*

$$\mathtt{rich}(\mathtt{bank}) = \top$$
$$\forall x(x \in Bank \Rightarrow x \in Rich)\,.$$

Example 3. *The following are equivalent*

$$\mathtt{not}_S(\mathtt{rich}(\mathtt{bank})) = \top$$
$$\forall x(x \in Bank \;\Rightarrow\; x \notin Rich)\,.$$

Important: *The first order formula in Example 3 is not the negation of the first order formula in Example 2. But then, the equality* $\mathtt{rich}(\mathtt{bank}) = \bot$ *is not the negation of the equality* $\mathtt{rich}(\mathtt{bank}) = \top$*, because there are more than two truth values.*

4 Semantics via pregroup grammars

Let E be a finite dimensional space with basis B. Call its basis vectors *individuals*. Single basis vectors of E correspond to singulars and sums to plurals of natural language. The examples below concern unary predicates only. The generalization to ordered pairs, triples etc. of individuals is straight forward. The space S is the two-dimensional space of two-sorted truth introduced in 3.3.

4.1 The computation of meanings

Like every other categorial grammar, a pregroup grammar is given by a lexicon and a calculus. The pregroup calculus is *compact bilinear logic* where proofs identify with morphisms in the free compact bicategory $\mathcal{C2}(\mathcal{B})$ generated by a partially ordered set $\mathcal{B}$.

The 1-cells are called *types* and the tensor product is written as concatenation. Hence, a *type* is a string of simple types, where a *simple type* is either an element of $x, y, \dots \in \mathcal{B}$ or an iterated adjoint $x^\ell, y^\ell, \dots, x^{\ell\ell}, y^{\ell\ell}, \dots, x^r, y^r, \dots, x^{rr}, y^{rr}, \dots$ etc. The types $x, y, \dots \in \mathcal{B}$ are called *basic types.* They stand for grammatical notions.

As a consequence, every functor from $\mathcal{B}$ into a semantic category $\mathcal{C}$ extends into a functor from $\mathcal{C2}(\mathcal{B})$ to $\mathcal{C}$ preserving the structure of compact bicategories.

A *lexicon* is a finite list of entries. An *entry* is a triple $w : T :: m$, where w is a word, T a type and m a meaning expression.

This description differs from the original one in [Lambek, 1999]. There, only pregroup 'dictionaries' are considered where the entries are pairs $w : T$, without meaning expressions. The latter must be added explicitly, because the functional semantics of categorial grammars with higher order types has been lost by the pregroup types.

The *meaning* in the entry is a formal expression $m : I \longrightarrow V$ in the language of compact closed categories or, equivalently, a string of two-sorted functions. It depends functionally on the pair $w : T$.

Consider the following entries

$$\begin{array}{ll} no\ : \boldsymbol{s}\boldsymbol{s}^\ell\boldsymbol{n}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\mathtt{no}}} S \otimes S^* \otimes E \otimes E^* & some : \boldsymbol{n}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\mathtt{some}}} E \otimes E^* \\ are : \boldsymbol{n}_2{}^r\boldsymbol{s}\boldsymbol{s}^\ell\bar{\boldsymbol{n}} :: I \xrightarrow{\overline{\mathtt{are}}} E^* \otimes S \otimes S^* \otimes E & big\ : \boldsymbol{c}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\mathtt{big}}} E \otimes E^* \\ and : \boldsymbol{s}^r\boldsymbol{s}\boldsymbol{s}^\ell :: I \xrightarrow{\overline{\mathtt{and_S}}} S^* \otimes S \otimes S^* & banks : \boldsymbol{c}_2 :: I \xrightarrow{\overline{\mathtt{bank}}} E \\ & rich\ : \bar{\boldsymbol{n}}^r\boldsymbol{s} :: I \xrightarrow{\overline{\mathtt{rich}}} E^* \otimes S \end{array}$$

The basic types $\boldsymbol{c}_2, \boldsymbol{n}_2, \bar{\boldsymbol{n}}, \boldsymbol{s}$, stand for 'plural count noun', 'plural noun phrase', 'dummy noun phrase' and 'sentence', in that order. Moreover, $\boldsymbol{c}_2 < \boldsymbol{n}_2$.

The properties of the meaning vector m in the entry $w : T :: m$ depend on the logical content of the word, given in due course, and on the type.

The lexicon defines a *canonical* functor from $\mathcal{B}$ into the compact closed category $\mathcal{C}$. For example, in the list of entries above, the basic type $\boldsymbol{s}$ is interpreted by S and each of the basic types $\boldsymbol{c}_2, \boldsymbol{n}_2, \bar{\boldsymbol{n}}$ by E. The canonical functor maps the inequality $\boldsymbol{c}_2 < \boldsymbol{n}_2$ to the identity 1_E and left and right adjoints of a type to 'the' adjoint space, because right and left adjoints may be identified in a symmetric monoidal category.

All meanings are presented in the form of names, up to a symmetry isomorphism that arranges the factors of the tensor product in the order given by the simple types. For example, $\overline{\mathtt{big}} = \sigma_{E^*,E} \circ \ulcorner big \urcorner : 1 \longrightarrow E \otimes E^*$ where $\mathtt{big} : E \longrightarrow E$.

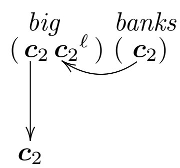
The meaning of a grammatical string involves both the meanings of the words and the syntactical analysis of the string. An ambiguous string with several parsings has several meanings. A non-grammatical string has no meaning.

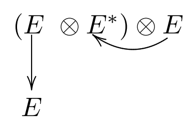
A string of words $w_1 \dots w_n$ is *grammatical* if there are entries $w_1 : T_1 :: m_1, \dots,\ w_n : T_n :: m_n$ and a basic type $\boldsymbol{b}$ such that

$$T_1 \dots T_n \vdash \boldsymbol{b}$$

is provable in compact bilinear logic. Otherwise said, if there is a morphism $f : T_1 \dots T_n \to \boldsymbol{b}$ in the syntactic category. Due to a theorem in [Lambek, 1999] the graph of the proof involves only underlinks and is called a *reduction*.

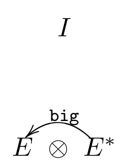
For example, the reduction corresponding to the graph on the left below analyses *big banks* as a plural noun-phrase. The graph on the right is the corresponding morphism in the semantic category.

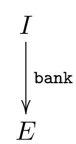
$1_{\boldsymbol{c}_2} \otimes \epsilon_{\boldsymbol{c}_2} =$ 

$r_1 = 1_E \otimes \epsilon_E =$ 

The reason why the syntactic category must not be symmetric is obvious in this example. The type $\boldsymbol{c}_2\boldsymbol{c}_2\boldsymbol{c}_2{}^{\ell}$ of the non-grammatical string *banks big* has no reduction to a basic type. If we had symmetry all order variants of a grammatical string would be grammatical.

The meaning vector of a lexical entry also identifies with a graph, for example

$I \xrightarrow{\overline{\texttt{big}}} E \otimes E^* =$ 

$\overline{\texttt{bank}} : I \longrightarrow E =$  .

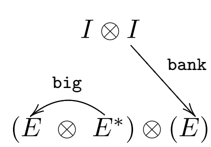
Again, the domain of the morphism is at the top of the graph, the codomain at the bottom.

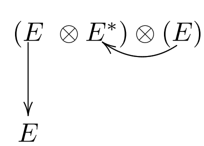
For a grammatical string $w_1 \dots w_n$, let $w_1 : T_1 :: m_1, \dots, w_n : T_n :: m_n$ be a choice of entries and r a reduction of $T_1 \dots T_n$ to a basic type. The corresponding *meaning* is

$$r \circ (m_1 \otimes \dots \otimes m_n)\,,$$

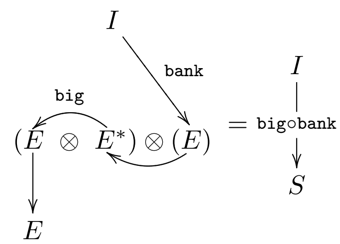
where now r denotes the linear map obtained by applying the canonical functor to the reduction.

The meaning of a string can be computed graphically. Connect the graphs at their joint interface and follow the paths from top to bottom picking up the labels along the way. For example, the graphs

$\overline{\texttt{big}} \otimes \overline{\texttt{bank}} =$ 

$r_1 =$ 

when connected at there joint interface compute to

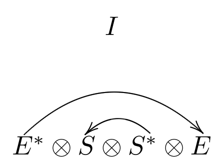
$r_1 \circ (\overline{\texttt{big}} \otimes \overline{\texttt{bank}}) =$ 

$= \texttt{big} \circ \texttt{bank}\,.$

The meaning vector $\overline{\texttt{are}} : I \longrightarrow E^* \otimes S \otimes S^* \otimes E$ is up to a symmetry isomorphism the name of the linear map $\texttt{are} : E \otimes S \longrightarrow E \otimes S$. The following postulate renders the *logical property* of the word *are*

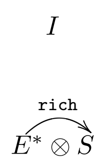
$$\texttt{are} = 1_E \otimes 1_S\,.$$

Hence the graph of $\overline{\texttt{are}}$ is

$\overline{\texttt{are}} =$ 

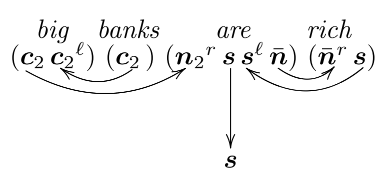
.

The lexicon assigns to the pair *rich* : $\bar{\boldsymbol{n}}^r \boldsymbol{s}$ a meaning vector $\overline{\texttt{rich}} : I \longrightarrow E^* \otimes S$. Its graph has the form

$\overline{\texttt{rich}} =$ 

, where $\texttt{rich} : E \longrightarrow S\,.$

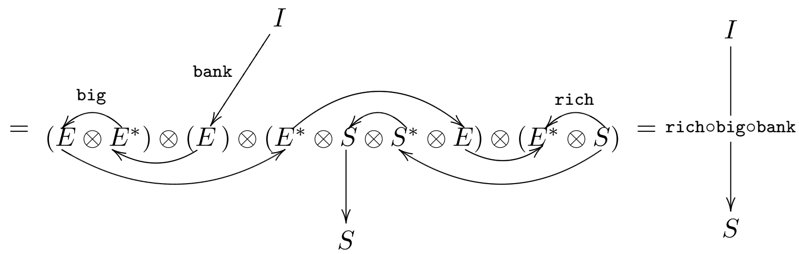
The reduction of the sentence *big banks are rich* is the graph

$r =$ 

.

Compute the meaning by composing the tensor product of the word vectors with the reduction

$$r \circ (\overline{\texttt{big}} \otimes \overline{\texttt{bank}} \otimes \overline{\texttt{are}} \otimes \overline{\texttt{rich}})$$

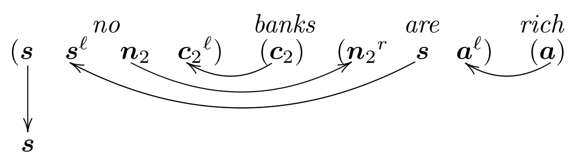


$$= \texttt{rich} \circ \texttt{big} \circ \texttt{bank}\,.$$

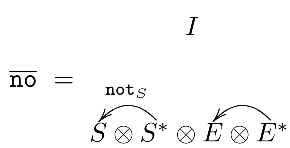
The sentence *All banks are rich* is computed by the same graph except that the label $\texttt{big}$ is replaced by the label $\texttt{all}$. A similar remark applies to the sentence *Some banks are rich.*

The last example concerns the computation of the meaning vector of the sentence *no banks are rich.*

- the reduction of the sentence is

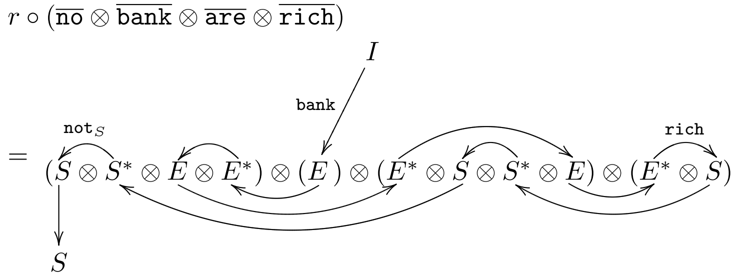


- the meaning vector $\overline{\texttt{no}} : I \longrightarrow S \otimes S^* \otimes E \otimes E^*$ is defined as the name of $\texttt{not}_S \otimes 1_E : S \otimes E \longrightarrow S \otimes E$ with the corresponding graph



- the meaning of the sentence *no banks are rich* is

$$r \circ (\overline{\texttt{no}} \otimes \overline{\texttt{bank}} \otimes \overline{\texttt{are}} \otimes \overline{\texttt{rich}})$$



$$= \texttt{not}_S \circ \texttt{rich} \circ \texttt{bank}\,.$$

'Walking graphs' makes the computation linear in the number of links. The latter is proportional to the length of the string of words, because the lexicon is finite and thus the length of its types is bounded.

4.2 The logical content of words

The preceding examples mention the logical content of some words. More generally, the description of the logical content can be organized according to the type of the words. We postulate

1. *Any $f : E^n \longrightarrow S$ occurring in a lexical entry is an n-ary two-sorted predicate.*
2. *Any $f : E \longrightarrow E$ occurring in a lexical entry is an intrinsic projector.*
3. *Any $f : I \longrightarrow E$ occurring in a lexical entry is a Boolean vector.*

Under these postulates, adjectives in predicative position and intransitive verbs are unary two-sorted predicates. The meaning map $\mathtt{word} : E \longrightarrow E$ in $word : \boldsymbol{n}_i \boldsymbol{c}_i^\ell :: \overline{\mathtt{word}} : I \longrightarrow E \otimes E^*$ interpreting an adjective in attributive position or a determiner is an intrinsic projector. The universal determiner satisfies an even stronger property

$$\mathtt{all} = 1_E \,.$$

Recall that B is the canonical basis of E and that any subset Y of B is identified with a Boolean vector, by Equality (8). Assume that $\mathtt{word} : E \longrightarrow E$ is an intrinsic projector occurring in the lexicon. Use the abbreviation

$$Word = \mathtt{word}(B) \,.$$

Combine this abbreviation with Equality (7) to obtain the equality

$$\mathtt{word}(Y) = Word \cap Y, \text{ for } Y \subseteq B \,. \tag{16}$$

For example, $Bank = \mathtt{bank}(B)$ and $\mathtt{big}(Bank) = Big \cap Bank$

4.3 Examples

The examples below concern sentences and their meaning vectors in the categories $\mathcal{2SF}$ and $\mathcal{RI}$. Represent the truth of a sentence by the fact that the meaning vector computes to $\top$. Then show that this representation coincides with the 'usual' translation of the sentence in logic.

Example 4. *All banks are rich* / $\mathtt{rich}(\mathtt{all}(\mathtt{bank}))$

The following are equivalent

$$\forall x (x \in Bank \Rightarrow Rich(x))$$
$$\mathtt{rich}(\mathtt{all}(\mathtt{bank})) = \top \,.$$

Proof. Recall that $\mathtt{all} = 1_E$. Hence $\mathtt{rich}(\mathtt{all}(\mathtt{bank})) = \mathtt{rich}(\mathtt{bank})$. For the proof of the equivalence of $\mathtt{rich}(\mathtt{bank}) = \top$ with $\forall x (x \in Bank \Rightarrow Rich(x))$ confer to Example 2. ☐

Example 5. *Big banks are rich* / $\mathtt{rich}(\mathtt{big}(\mathtt{bank}))$

The following are equivalent

$$\forall x(x \in Big \cap Bank \Rightarrow Rich(x))$$
$$\texttt{rich}(\texttt{big}(\texttt{bank})) = \top\,.$$

Proof. Recall that $\texttt{bank} = \sum_{x \in Bank} x$ and therefore the vector $\texttt{big}(\texttt{bank})$ is identified with the set $\texttt{big}(Bank) = Big \cap Bank$ by (16). The equivalence now follows from the Fundamental Property (14). □

Example 6. *No banks are rich* / $\texttt{not}_S(\texttt{rich}(\texttt{bank}))$

The following are equivalent

$$\texttt{not}_S(\texttt{rich}(\texttt{bank})) = \top$$
$$\forall x(x \in Bank \;\Rightarrow\; x \notin Rich)$$

Proof. See Example 3. □

Example 7. *Some banks are rich/*$\texttt{rich}(\texttt{some}(\texttt{bank}))$

The implication

$$\texttt{rich}(\texttt{some}(\texttt{bank})) = \top \Rightarrow \exists_x(x \in Bank \,\&\, Rich(x))$$

holds, but its converse does not hold in general.

Proof. The equality $\texttt{rich}(\texttt{some}(\texttt{bank})) = \top$ implies $\texttt{some}(\texttt{bank}) \neq \emptyset$ and $\forall_x(x \in \texttt{some}(\texttt{bank}) \Rightarrow \texttt{rich}(x) = \top)$, by the Fundamental Property. The first order formula follows, because $\texttt{some}(\texttt{bank}) \subseteq \texttt{bank}$.

If the first order formula holds, take a witness $b \in Bank$ for which *Rich(b)* holds. Let $\texttt{some}_b$ be the intrinsic projector that maps b to itself and every other individual to the null vector. Clearly $\texttt{some}_b(\texttt{bank}) = b$ and $\texttt{rich}(b) = \top$. Hence,

$$\texttt{rich}(\texttt{some}_b(\texttt{bank})) = \top\,,$$

but this does not imply that the particular intrinsic projector $\texttt{some}_b$ coincides with the original $\texttt{some}$. □

Natural language confronts us with a problem. On one hand, the interpretation of '*some Y*' changes from occurrence to occurrence, like in *Some banks are rich and some banks are not rich.* On the other hand, '*some Y*' acts like a well determined reference set. In *Some banks are rich. They scare me*, the personal pronoun *they* refers to the set *some banks* of the preceding sentence.

The interpretation of $\texttt{some}$ given above may vary from occurrence to occurrence and in the same time it defines a set at each occurrence, which is available for later reference, e.g. $\texttt{they} = \texttt{some}(\texttt{bank})$.

The interpretation by a generalized quantifier, see [Barwise and Cooper, 2002], takes into account the change of meaning in different occurrences, but it does not construct the set to which the noun phrase refers.

5 Compositional semantics in concept spaces

Quantum logic stands for 'logic of projectors in a concept space' and *concept* for 'Boolean vector in a concept space'.

5.1 Classical propositional calculus in concept spaces

Let $P = \{p_1, \ldots, p_d\}$ be a non-empty set. Call *compound system* or *concept space* the tensor product

$$C(P) = C(p_1) \otimes \ldots \otimes C(p_d)\,,$$

where $C(p_i)$ is a 2-dimensional space with basis vectors $p_{i\top}, p_{i\bot}$, for $i = 1, \ldots, d$.

The space $C(p_i)$ is a 'basic variable' in quantum protocols and a 'basic concept' in semantics for natural language. For example, key-words of Roget's (or the speaker's mental) thesaurus provide sets of basic concepts.

Any basis vector b_f of $C(P)$ is a tensor product of basis vectors of the factors

$$b_f = f(1) \otimes \ldots \otimes f(d), \text{ where } f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\}\ .$$

Due to the fine-grained structure of the basis vectors, the Boolean algebra of intrinsic projectors of a concept space is isomorphic to the Boolean algebra freely generated by the set P. The rest of this subsection is devoted to the proof of this fact.

For every $i = 1, \ldots, d$, define the two so-called *primitive vectors*

$$\begin{array}{ll} \overrightarrow{p_i} & = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes p_{i\top} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \sum_{f, f(i) = p_{i\top}} b_f \\ \overrightarrow{\neg p_i} & = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes p_{i\bot} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \sum_{f, f(i) = p_{i\bot}} b_f\ . \end{array}$$

The two primitive vectors defined by $p_i \in P$ are orthogonal to each other. In fact, each is the negation of the other one

$$\neg\overrightarrow{p_i} = \overrightarrow{\neg p_i} \text{ and } \neg(\overrightarrow{\neg p_i}) = \overrightarrow{p_i}\ .$$

Every Boolean vector can be written as a disjunction of conjunctions of primitive vectors. Indeed, let $\{j_1, \ldots, j_k\}$ be a subset of $\{1, \ldots, d\}$. Assume that $g \in \prod_{i=1}^{d} \left\{p_{i\top}, p_{i\bot}, \overrightarrow{1}\right\}$ satisfies for $i = 1, \ldots, d$

$$g(i) \in \{p_{i\top}, p_{i\bot}\} \quad \text{if and only if } i \in \{j_1, \ldots, j_k\}\ . \tag{17}$$

The *partial choice vector* associated to g is

$$v_g = g(1) \otimes \ldots \otimes g(d)\,.$$

Lemma 8. *Every partial choice vector v_g is a conjunction of primitive vectors. In particular, every basis vector is a conjunction of primitive vectors.*

Proof. Assume that g satisfies(17). Let $q_{j_l} = g(j_l) \in \{p_{j_l\top}, p_{j_l\bot}\}$, $l = 1, \ldots, k$, and $G = \left\{ f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\} \ : \ f(j_l) = q_{j_l}, \text{ for } l = 1, \ldots, k \right\}$. Then

$$v_g = \sum_{f \in G} b_f = \overrightarrow{q_{j_1}} \wedge \cdots \wedge \overrightarrow{q_{j_k}} \,. \tag{18}$$

□

Theorem 2. *The free Boolean algebra generated by P is isomorphic to the lattice of intrinsic projectors of $C(P)$. The map $p \mapsto \overrightarrow{p}$ extends to an isomorphism $\mathcal{K}$ from $B(P)$ onto the lattice of Boolean vectors of $C(P)$.*

Proof. Partial choice vectors are Boolean vectors by (18). Hence, the map $p \mapsto \overrightarrow{p}$ extends to a unique Boolean homomorphism $\mathcal{K}$ from $B(P)$ into the Boolean algebra of Boolean vectors of $C(P)$. A classical theorem, [Halmos, 1974], states that the free Boolean algebra $B(P)$ is isomorphic to the set algebra generated by the following subsets of $\prod_{j=1}^{d} \{0, 1\}$

$$p_i \simeq \left\{ h \in \prod_{j=1}^{d} \{0,1\} \ : \ h(i) = 1 \right\}, \quad \neg p_i \simeq \left\{ h \in \prod_{j=1}^{d} \{0,1\} \ : \ h(i) = 0 \right\},$$

where i varies from 1 to d. Every singleton set $\{h\}$ can be written as a conjunction of subsets of the form p_i or $\neg p_i$. Therefore the homomorphism $\mathcal{K}$ maps $\{h\}$ to the corresponding basis vector in $C(P)$. It follows that the homomorphism $\mathcal{K}$ maps $B(P)$ onto the lattice of Boolean vectors. It is one-to-one, because every Boolean vector can be written uniquely as a sum of basis vectors.

Compose $\mathcal{K}$ with the isomorphism $\mathcal{H}$ of Theorem 1 to obtain the isomorphism $\mathcal{H} \circ \mathcal{K}$ onto the lattice of intrinsic projectors. □

By Theorem 2, the *primitive projectors* $p_{\overrightarrow{p_i}}$ and $p_{\neg\overrightarrow{p_i}}$ play an important role in the lattice of projectors of $C(P)$:

1. Every intrinsic projector is a finite disjunction of finite conjunctions of primitive projectors
2. The lattice of intrinsic projectors in a compound system is the classical propositional calculus modulo equiderivability.
3. One can use *induction on the complexity of propositions* for defining and proving properties of Boolean vectors/intrinsic projectors.

Propositional complexity creates a somewhat unusual hierarchy on subspaces. Recall that $com(p_i) = 0$, $com(\neg p) = 1 + com(p)$, and $com(p \wedge q) = 1 + max(com(p), com(q))$. The primitive subspace $E_{\overrightarrow{p_i}}$ has complexity 0 and $E_{\neg\overrightarrow{p_i}}$ has complexity 1, but both have dimension 2^{d-1}. The one-dimensional subspace generated by a single basis vector b_f has complexity $d-1$ if $f(i) = p_i$ for $i = 1, \ldots, d$, and complexity d otherwise.

5.2 Concept spaces and two-sorted truth

A *classification system* consists of

1. a set B (of individuals, pairs of individuals etc.)
2. a set $P = \{p_1, \ldots, p_d\}$ (of properties)
3. a relation $\models \subseteq B \times P$

Read $x \models p$ as 'x satisfies p'.

Extend the relation $\models$ to arbitrary concepts in $C(P) = C(P_1) \otimes \ldots \otimes C(P_d)$ for every individual $x \in B$ using induction on the complexity of concepts

$$\begin{array}{ll} x \models \overrightarrow{p_i} & \text{if and only if } \; x \models p_i \\ x \models \neg v & \text{if and only if } \; x \not\models v \\ x \models v \wedge w & \text{if and only if } \; x \models v \;\&\; x \models w \\ x \models v \vee w & \text{if and only if } \; \models v \text{ or } x \models w\,. \end{array}$$

Clearly, either $x \models v$ or $x \models \neg v$ holds for every individual $x \in B$ and every concept $v \in C(P)$.

Extend satisfaction to every non-empty subset Y of B and every concept v

$$Y \models v \text{ if and only if } x \models v \text{ for all } x \in Y\,. \tag{19}$$

Read $Y \models v$ as 'Y has property v in general'.

Note that $Y \not\models v$ and $Y \not\models \neg v$ may hold simultaneously. It suffices that Y has an element satisfying v and another one that does not satisfy v.

The satisfaction system induces a representation of (sets of) individuals by concepts in $C(P)$. For any $x \in B$ let

$$\begin{array}{l} q(x)_i = p_{i\top} \text{ if } x \models p_i \\ q(x)_i = p_{i\bot} \text{ if } x \not\models p_i\,, \end{array}$$

for $i = 1, \ldots, d$. This choice determines a basis vector, the *concept v_x internalizing x*,

$$v_x = q(x)_1 \otimes \ldots \otimes q(x)_d\,.$$

For any non-empty subset $Y \subseteq B$ define the *concept v_Y internalizing Y*

$$v_Y = \sum_{x \in Y} v_x\,.$$

Different individuals may be internalized by the same basis vector. This means that they are indiscernible by the properties listed in P.

Lemma 9. *For any concept $c \in C(P)$ and any individual $x \in B$*

$$x \models c \textit{ if and only if } \; v_x \le c\,. \tag{20}$$

In particular, for any basis vector $b_f \in C(P)$

$$x \models b_f \textit{ if and only if } v_x = b_f\,. \tag{21}$$

For $Y \neq \emptyset$

$$Y \models c \textit{ if and only if } \; v_Y \le c\,. \tag{22}$$

Proof. Show (20), the equivalence concerning individuals, by induction on the propositional complexity of c. Equivalence (21) is a particular case of (20).

The equivalence concerning sets, (22), now follows from the equivalence for individuals. □

One consequence of the lemma above is that satisfaction in a classification system coincides with the conditional logic for Boolean vectors/projectors. Indeed, the inequality $v \leq w$ is equivalent to $v \to w = \overrightarrow{1}$, where the full vector $\overrightarrow{1}$ stands for 'true'.

Another consequence is that the concept v_x is the best possible description of the individual x in the classification system and the same holds for v_Y and the set Y.

5.3 Intrinsic projectors and two-sorted predicates

Let E be a space with basis B. One can think of any satisfaction system $(B, P = \{p_1, \ldots, p_d\}, \models)$ as a model of the language generated by P. It suffices to think of $p \in P$ as the two-sorted predicate $p(\,.\,) : E \longrightarrow S$ satisfying

$$p(x) = \begin{cases} \top & \text{if } x \models p \\ \bot & \text{if } x \not\models p \end{cases}, \text{ for all } x \in B\,.$$

Recall that $x \models v$ is equivalent to $v_x \leq v$. The expressiveness of the logic remains unchanged if the individuals in B are replaced by the basis vectors internalizing them. Indeed, individuals x, y for which $v_x = v_y$ are indiscernible in the logic.

The compound system $C(P)$ is endowed with a *canonical satisfaction system*, namely

$$\begin{array}{rcl} P & = & \{p_1, \ldots, p_d\} \\ B & = & \left\{ b_f \;:\; f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\} \right\} \\ x \models p & \Leftrightarrow & x \leq \overrightarrow{p},\ \text{for all } x \in B,\, p \in P\,. \end{array}$$

In the canonical satisfaction system, a basis vector can be both an individual and a concept. More generally, every Boolean vector is both a set of individuals and a concept.

Given $p \in P$, define a two-sorted predicate $\mathcal{L}(\overrightarrow{p})$ on $C(P)$ by stipulating

$$\begin{array}{ll} \mathcal{L}(\overrightarrow{p})(x) = \top & \Leftrightarrow x \leq \overrightarrow{p} \\ \mathcal{L}(\overrightarrow{p})p(x) = \bot & \Leftrightarrow x \not\leq \overrightarrow{p} \end{array}, \text{ for all } x \in B\,. \tag{23}$$

Theorem 3. [Definability of predicates] *The map $\overrightarrow{p} \mapsto \mathcal{L}(\overrightarrow{p})$ extends to a Boolean isomorphism from the lattice of concepts of $C(P)$ onto the Boolean algebra of two-sorted predicates on $C(P)$ satisfying*

$$\begin{array}{ll} \mathcal{L}(\neg v) & = \mathtt{not}_S \circ \mathcal{L}(v) \\ \mathcal{L}(v \wedge w) & = \mathtt{and}_S \circ (\mathcal{L}(v) \otimes \mathcal{L}(w)) \circ 2_E\,. \end{array} \tag{24}$$

Moreover, if K is a non-empty subset of k basis vectors and $w = \sum_{x \in K}$ the following equivalences hold

$$\begin{array}{lcl} \mathcal{L}(v)(w) = k \cdot \top & \Leftrightarrow & w \leq v \\ \mathcal{L}(v)(w) = k \cdot \bot & \Leftrightarrow & w \leq \neg v\,. \end{array} \tag{25}$$

Proof. The extension of $\mathcal{L}$ to all Boolean vectors such that (24) holds is guaranteed by Theorem 2 .

Next, prove (25) in the particular case where K consists of a single basis vector, i.e. prove that

$$\begin{array}{lcl} \mathcal{L}(v)(x) = \top & \Leftrightarrow & x \leq v \\ \mathcal{L}(v)(x) = \bot & \Leftrightarrow & x \leq \neg v \end{array}\,, \text{ for all } x \in B\,. \tag{26}$$

Use induction on the propositional complexity of v . If the complexity is 0 then $v = \overrightarrow{p}$, for some $p \in P$. The two equivalences of (26) hold for $\overrightarrow{p}$ by (23) and the fact that $x \leq \neg \overrightarrow{p}$ if and only if $x \not\leq \overrightarrow{p}$.

For the induction step, assume that (26) holds for v . Recall that $\mathtt{not}_S$ is the symmetry isomorphism that exchanges the two basis vectors $\top$ and $\bot$. Thus

$$\mathcal{L}(\neg v)(x) = \mathtt{not}_S \circ \mathcal{L}(v)(x) = \top \Leftrightarrow \mathcal{L}(v)(x) = \bot\,.$$

The righthand equality above is equivalent to $x \leq \neg v$ by induction hypothesis. The equivalence $\mathcal{L}(\neg v)(x) = \top \Leftrightarrow \; x \leq \neg v$ follows.

Next, assume that (26) holds for the concepts v and w . Then

$$\mathcal{L}(v \wedge w)(x) = \mathtt{and}_S \circ (\mathcal{L}(v)(x) \otimes \mathcal{L}(w)(x))\,.$$

Therefore, $\mathcal{L}(v \wedge w)(x) = \top$ holds exactly if both $\mathcal{L}(v)(x) = \top$ and $\mathcal{L}(w)(x) = \top$ hold, by definition of $\mathtt{and}_S$. The latter two equalities are equivalent to $x \leq v$ and $x \leq w$ by induction hypothesis, and to $x \leq v \wedge w$ by the definition of vector conjunction. This terminates the proof that the first equivalence of (26) holds for $v \wedge w$. The proof of the second equivalence is similar. Hence the particular case (26) holds for all Boolean vectors.

Next, (26) implies that $\mathcal{L}$ is one-to-one. Indeed, if v and w are different Boolean vectors there is a basis vector x such that $x \leq v$ and $x \leq \neg w$.

Next, for showing that $\mathcal{L}$ is onto, assume that $r : C(P) \longrightarrow S$ is an arbitrary two-sorted predicate on $C(P)$. Let $K = \{x \in B \, : \, r(x) = \top\}$ and $v = \sum_{x \in K} x$. Then $r(x) = \bot$ for all basis vectors $x \notin K$, because $\bot$ is the only other possible value of r for a basis vector. Thus $r(x) = \top$ if and only if $x \leq v$ and $r(x) = \bot$ if and only if $x \leq \sum_{x \in B \setminus K} x = \neg v$. The equality $r = \mathcal{L}(v)$ follows by (26).

To show (25) in the general case, let K be a non-empty subset of k basis vectors and $w = \sum_{x \in K}$. Then $\mathcal{L}(v)(w) = k \cdot \top$ if and only if $v(x) = \top$ for all $x \in K$. The latter is equivalent to $w \leq v$, because of (26). The proof of the second equivalence of (25) is similar. □

A succinct summary of the theorem above says that every predicate on $C(P)$ can be expressed as a Boolean combination of the predicates $\mathcal{L}(\overrightarrow{p})$ for $p \in P$.

The fact that the space of indivuals is the concept space $C(P)$ is essential here. Assume that a and b are distinct individuals of some space E that indiscernible by the properties $p_i \in P$. Then the two-sorted predicate r on E that maps x to $\top$ if and only if $x = b$ is not definable by a concept of $C(P)$.

Note that the homomorphism $\mathcal{L}$ maps the full vector $\overrightarrow{1} \in C(P)$ to the predicate that is 'everywhere' true, where 'everywhere' means 'for all basis vectors'. Switching from Boolean vectors to intrinsic projectors, identify the intrinsic projector p_v with the two-sorted predicate $\mathcal{L}(v)$. In particular, $1_{C(P)} = p_{\overrightarrow{1}}$ identifies with the predicate that is 'everywhere' true. The equalities (24), recast in terms of projectors, connect predicate logic and projector logic thus

$$\mathtt{not}_S \circ \mathcal{L}(v) = p_v^{\perp}, \quad \mathtt{and}_S \circ (\mathcal{L}(v) \otimes \mathcal{L}(w)) \circ 2_{C(P)} = p_v \circ p_w .$$

Theorems 1 and 3 bring a new understanding to projectors in a compound system $C(P)$. The slogans 'negation is orthogonality', 'conjunction is composition of projectors' can be extended to 'every grammatical string corresponds to a projector such that predicate logic becomes quantum logic'. It suffices to make $E = C(P)$ in Subsection 4.1 to compute the projector.

Thus, the sample sentences of Section (4.3) have two interpretations in $C(P)$. One is a two-sorted predicate and the other one a projector. Assuming that we evaluate the former in $\mathcal{2SF}$ or $\mathcal{RI}$, we have the following equivalences concerning the two interpretations

All banks are rich / $\mathtt{rich}(\mathtt{all}(\mathtt{bank}))$ / $p_{\mathtt{bank}} \to p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{all}(\mathtt{bank})) = \top \Leftrightarrow (p_{\mathtt{bank}} \to p_{\mathtt{rich}}) = 1_{C(P)} .$$

Big banks are rich / $\mathtt{rich}(\mathtt{big}(\mathtt{bank}))$ / $p_{\mathtt{big}} \circ p_{\mathtt{bank}} \to p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{big}(\mathtt{bank})) = \top \Leftrightarrow (p_{\mathtt{big}} \circ p_{\mathtt{bank}} \to p_{\mathtt{rich}}) = 1_{C(P)} .$$

No banks are rich / $\mathtt{not}_S(\mathtt{rich}(\mathtt{bank}))$ / $p_{\mathtt{bank}} \to p_{\mathtt{rich}}^{\perp}$

$$\mathtt{not}_S(\mathtt{rich}(\mathtt{bank})) = \top \Leftrightarrow (p_{\mathtt{bank}} \to p_{\mathtt{rich}}^{\perp}) = 1_{C(P)} .$$

Some banks are rich / $\mathtt{rich}(\mathtt{some}(\mathtt{bank}))$ / $p_{\mathtt{some(bank)}} \to p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{some}(\mathtt{bank})) = \top \Leftrightarrow (p_{\mathtt{some(bank)}} \to p_{\mathtt{rich}}) = 1_{C(P)} .$$

This means that projector equalities translate to quantified formulas of two-sorted first order logic. Otherwise said, quantum logic includes two-sorted first order logic. For example,

$$(p_{\mathtt{bank}} \to p_{\mathtt{rich}}^{\perp}) = 1_{C(P)} \Leftrightarrow \forall x (x \in \mathit{Bank} \Rightarrow x \notin \mathit{Rich}) .$$

These equivalences concern the categories $\mathcal{2SF}$ or $\mathcal{RI}$, where predicates cannot count but only assert. The next subsection deals with Hilbert spaces, where predicate count the elements that satisfy them.

5.4 States in a concept space

In this subsection, $\mathcal{C}$ is the category of finite dimensional real Hilbert spaces.

A satisfaction relation requires a yes or no answer for every individual and every basic property p_i. For practical reasons such a precise information may not be available. Instead, real numbers $\alpha_{iY} \in [0,1]$ are available representing the probability that an arbitrary individual in Y has property p_i, $i = 1, \ldots, d$.

Let $0 \le \alpha_{iY} \le 1$ and $\beta_{iY} = 1 - \alpha_{iY}$ for $i = 1, \ldots, d$. The set Y is represented in $C(p_1) \otimes \ldots \otimes C(p_d)$ by its *state vector*

$$\mu_Y = (\alpha_{1Y} p_{1\top} + \beta_{1Y} p_{1\perp}) \otimes \ldots \otimes (\alpha_{dY} p_{d\top} + \beta_{dY} p_{d\perp}).$$

Lemma 10. *The coordinates of μ_Y define a probability on the event space $B(P)$ generated by the $\overrightarrow{p_i}$'s. Moreover, α_i is equal to the sum of the coordinates of $\overrightarrow{p_i} \wedge \mu_Y$ and β_i to the sum of the coordinates of $\neg\overrightarrow{p_i} \wedge \mu_Y$.*

Proof. Let γ_f be the coordinate of μ_Y for $f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\perp}\}$. Then $\overrightarrow{p_i} \wedge \mu_Y = \sum_{f,\, f(i)=p_{i\top}} \gamma_f b_f$. Hence the assertions follow from the equalities

$$\sum_f \gamma_f = 1, \quad \alpha_i = \sum_{f(i)=p_{i\top}} \gamma_f, \quad \beta_i = \sum_{f(i)=p_{i\perp}} \gamma_f.$$

Prove the first equality by induction on d. The case $d = 1$ is trivial. For the induction step, let $d' = d - 1$, $P' = \{1, \ldots, d'\}$ and

$$\mu'_Y = (\alpha_{1Y} p_{1\top} + \beta_{1Y} p_{1\perp}) \otimes \ldots \otimes (\alpha_{d'Y} p_{d'\top} + \beta_{d'Y} p_{d'\perp}).$$

Let δ_g be the coordinate of μ'_Y in $C(P')$, i.e. $\mu'_Y = \sum_{g \in \prod_{i=1}^{d'} \{p_{i\top}, p_{i\perp}\}} \delta_g b_g$. Then $\sum_g \delta_g = 1$ by induction hypothesis. We have

$$\mu_Y = \mu'_Y \otimes (\alpha_{dY} p_{d\top} + \beta_{dY} p_{d\perp}) = \sum_g \delta_g \alpha_{dY} (b_g \otimes p_{d\top}) + \sum_g \delta_g \beta_{dY} (b_g \otimes p_{d\perp}).$$

This finishes the proof, because for every basis vector $b_f \in C(P)$ there is a unique $g \in \prod_{i=1}^{d-1} \{p_{i\top}, p_{i\perp}\}$ such that either $b_f = b_g \otimes p_{d\top}$ or $b_f = b_g \otimes p_{d\perp}$. $\square$

The projector $p_{\overrightarrow{p_i}}$ of $C(P)$ maps the state vector μ_Y to a vector $p_{\overrightarrow{p_i}}(\mu_Y) = \overrightarrow{p_i} \wedge \mu_Y$, the coordinates of which sum up to α_i. Hence the projector $p_{\overrightarrow{p_i}}$, equivalently, the two-sorted predicate $\mathcal{L}(\overrightarrow{p_i})$, returns for μ_Y the probability that an arbitrary individual in Y satisfies p_i.

Return to vector semantics in information retrieval systems. Choose a set $P = p_1, \ldots, p_d$ of basic properties, for example the most frequent words in a (set of) document(s). Represent words by vectors in the d-dimensional space V_P, where the coordinate γ_i of word w is the frequency of co-occurrence with p_i. The projection onto the one-dimensional subspace of V_P generated by p_i is the vector $\gamma_i p_i$.

The scalar γ_i may be interpreted as the similarity of the word with p_i, but not as the probability that an arbitrary individual designated by w has property p_i, because positive and negative occurrences like *some banks are safe*, *some banks are not safe* contribute both to γ_i. 'Reasoning by probability' based on frequency counts requires a distinction between positive and negative occurrences.

6 Conclusion

New in this approach is that two separate notions of truth, one for concepts and one for sentences, are handled formally inside a single mathematical frame with a resulting equivalence of the two representations. The geometrical properties of quantum logic and the functional application of logic are preserved.

On a technical level, both the tensor product and syntactical analysis intervene when composing meanings.

Many interesting questions have not been addressed. For example, biproducts of concept spaces are necessary to handle predicates of an arbitrary arity simultaneously. Ambiguous words as well live in a biproduct of different concept spaces. Disambiguation by context uses the probability that the meaning factors through one branch rather than the other of the biproduct.

The most challenging questions belong to the probabilistic approach to natural language semantics and its relation to compositionality. How to distinguish between opposites? (The usual probabilistic approach confounds them.) How to capture the intuitive interaction of statistical learning of concepts and their logical use?

References

Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pages 415–425, 2004.

Jon Barwise and Robin Cooper. *Formal Semantics: the essential readings*, chapter Generalized Quantifiers and Natural Language, pages 76–125. Wiley-Blackwell, 2002.

Johan van Benthem and Kees Doets. *Handbook of Philosophical Logic*, chapter Higher-Order Logic, pages 275–329. Reidel Publishing Company, Dordrecht, 1983.

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

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. A compositional distributional model of meaning. In W. Lawless P. Bruza and J. van Rijsbergen, editors, *Proceedings of Conference on Quantum Interactions*. University of Oxford, College Publications, 2008.

Paul Halmos. *Lectures on Boolean algebras.* Springer, 1974.

Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

Joachim Lambek. *Substructural Logics*, chapter From categorial grammar to bilinear logic, pages 207–237. Oxford University Press, 1993.

Joachim Lambek. Type grammar revisited. In Alain et al. Lecomte, editor, *Logical Aspects of Computational Linguistics*, volume 1582 of *LNAI*, pages 1–27, Heidelberg, 1999. Springer.

Anne Preller. Category theoretical semantics for pregroup grammars. In Philippe Blache and Edward Stabler, editors, *Logical Aspects of Computational Linguistics*, volume 3492 of *Lecture Notes in Artificial Intelligence*, pages 254–270, 2005.

Anne Preller. Toward discourse representation via pregroup grammars. *Journal of Logic, Language and Information*, 16:173–194, 2007. doi: http://dx.doi.org/10.1007/s10849-006-9033-y.

Anne Preller and Joachim Lambek. Free compact *2*-categories. *Mathematical Structures for Computer Sciences*, 17(1):1–32, 2007. doi: http://dx.doi.org/10.1017/S0960129506005901.

Anne Preller and Mehrnoosh Sadrzadeh. Semantic vector models and functional models for pregroup grammars. *Journal of Logic, Language and Information*, 20(4):419–423, 2011. doi: http://dx.doi.org/10.1007/s10849-011-9132-2.

C.J. van Rijsbergen. *The Geometry of Information Retrieval.* Cambridge University Press, 2004.

Peter Selinger. Dagger compact closed categories and completely positive maps (extended abstract). In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005*, pages 139–163, 2007. doi: http://dx.doi.org/10.1016/j.entcs.2006.12.018.

Paul Smolensky. Connectionism, constituency and language of thought. In Robert Cummins and Denise Dellarosa Cummins, editors, *Minds, Brains, and Computers*, pages 284–308, 1988.

Dominic Widdows. *Geometry and Meaning.* Number 172 in CSLI lecture notes. CSLI Publications, 2004.