From Sentence to Concept, a Linguistic Quantum Logic

Anne Preller

FROM SENTENCE TO CONCEPT, A LINGUISTIC QUANTUM LOGIC

ANNE PRELLER

Abstract. The category of semi-modules over the lattice of a real interval I serves as a common frame for extensional logical semantics and conceptual vector semantics of natural language. The vector lattice of an I-space is embedded in its projector lattice. Words are represented by an extensional vector given in a pregroup lexicon and a conceptual vector that lives in a tensor product of 2-dimensional spaces where each 2-dimensional space stand for a 'basic' concept. Syntactical analysis defines the extensional vector of a grammatical string compositionally. The latter is translated to the concept vector of the string such that the truth of a sentence is equivalent to the truth of a concept. Examples include compound noun phrases with adjectives, quantifiers, negation and relative pronouns.

keywords: *natural language processing, semantic vector models, logic for information retrieval systems, concept spaces, compositional semantics, two-sorted first order logic, 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 the relations between them.

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 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 words like negation, determiners, quantifiers, relative pronouns. Properties of individuals are expressed by truth-value functions.

The semantic vector models of Information Retrieval Systems do not require syntactic analysis and are neither extensional nor compositional. They are based on the principle that the content of a word is measured in relation to the content of other words. Words are represented by vectors of a finite dimensional space over the field of real numbers. Among others, [15] proposes quantum logic defined on the lattice of projectors of Hilbert spaces for reasoning with vectors. Quantum logic has only propositional logical connectives, which suffice when dealing with systems or properties of sets. Various authors propose to represent a string of words by the tensor product of the word vectors, among them [16] and [5], to make vector space semantics compositional. The former is based

supported by LIRMM/CNRS

on computational principles of cognitive science, the latter proposes syntactical analysis as a computing tool of vector representation.

The contribution presented here outlines a common mathematical frame for both semantics and a method transforming the extensional representation into a conceptual representation. The common frame, the category of semi-modules over the semi-ring of a real interval I, is the linguistic counterpart to the category of complex Hilbert space in quantum logic.

Grammatical strings of words may be viewed as the pendents to the experiments of quantum mechanics. Text makes truth-statements about the individuals of the system and it provides a statistical description of the state of the system. Hence a word gives rise to two vectors. One standing for the extension, the other one for the concept. Syntactical analysis defines a linear map that can be applied to the tensor product of the extensional vectors. The result is the extensional vector of the string, which simplifies to an expression involving linear maps and vectors only. The concept vector is derived from the structure of the extensional vector. In the case of a sentence, the extensional vector is true if and only if the concept vector is true. This equivalence shows that 'Montague semantics' and 'Capulet semantics', see [9], are complementary rather than opposites.

Syntactical analysis consists in a derivation by a pregroup grammar. Such a derivation has a canonical interpretation in any compact closed category, in particularly in a category of linear maps and vector spaces. Pregroup grammars, [8], are a simplification of the syntactic calculus, [7], and are based on compact bilinear logic. Roughly speaking, 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, [3], [2]. Moreover, the category of types and proofs of compact bilinear logic is the free compact *2*-category , [13]. Categorical semantics in compact 2-categories for pregroup grammars were first proposed in [10], reformulated in [12] in terms of two-sorted first order functional logic. Two-sorted functions and sets as well as vector spaces and linear maps form compact closed categories with an embedding of the former into the latter, [14].

The concept vectors of strings live in the so-called *concept space*, a tensor product of two-dimensional semi-modules over a fixed real interval I. The concept space is the linguistic pendent to the compound system of quantum mechanics, a tensor product of two-dimensional Hilbert spaces. The analogy with quantum logic stems from the categorical foundation of quantum mechanics given in [1]. Indeed, the category of semi-modules over the real lattice and the category of complex Hilbert spaces are both symmetric compact closed categories. The notions based on the same mathematical definition in one and the other category are analogue to each other.

The main result of Section 2 says that the lattice of vectors of an I-semi-module is homomorphically embedded into its lattice of projectors. The propositional connectives of the latter are defined like in Hilbert spaces and have the same geometrical representation.

Section 3 refines these properties in the case of a concept space, i.e. a tensor product of two-dimensional spaces, where each two-dimensional space represents a basic concept. The lattice of projectors includes the free Boolean algebra generated by the set of basic concepts. The concept vector of a word identifies with a projector in the Boolean sublattice. Hence the propositional logic of projectors representing concepts is classical.

In Section 4, a first link between concepts and extensions is made by using the basis vectors of a distributional vector model as the basic concepts of a concept space. As the factor spaces of a concept space are two-dimensional, each factor allocates two states to the basic concept, one dimension for 'concept present', the other one for 'concept absent'. Thus, the basic concepts behave like the key-words of a classification system leading to a definition of satisfaction of a concept by individuals respectively sets of individuals.

Section 5 recalls pregroup grammars and presents the (proof-)graphs of compact bilinear logic used in syntax and semantics of pregroup grammars. The 'noise' words ignored in the probabilistic approach now have a logical content. They are responsible for the structure of the extensional vector and ultimately of the concept vector of a grammatical string of words.

Section 6 includes a proof that concept vector of a sentence is true in quantum logic if and only if its extensional vector true in functional logic. The proof is based on an explicit construction of the concept vector of the string, depending on the syntactical structure and the logical content of 'noise' words. The examples are sentences with compound noun-phrases involving adjectives, quantifiers, negation and relative pronouns.

§2. Vector spaces over an interval of real numbers. Distributional models represent words by 'semantic' vectors of a finite dimensional real space, i.e. vectors the coordinates of which are obtained by a frequency count in text-windows. Without loss of generality, one may assume that they belong to the real interval $I = [0,1]$. The linear order of the real numbers induces a semi-ring structure over I. Hence semantic vectors belong to a semi-module over the lattice of *scalars* I.

2.1. The lattice structure of I. The order on $I = [0,1]$ induces a distributive and implication-complemented lattice structure on I as usual, namely

$$\begin{gathered} \alpha \vee \beta = \max\{\alpha, \beta\} \text{ and } \alpha \wedge \beta = \min\{\alpha, \beta\} \\ \alpha \to \beta = \max\{\gamma \in I : \alpha \wedge \gamma \leq \beta\} \\ \neg\alpha = \alpha \to 0\,. \end{gathered} \tag{1}$$

Note that

$$\alpha \to \beta = \begin{cases} 1 & \text{if } \alpha \leq \beta \\ \beta & \text{if } \alpha > \beta \end{cases} \quad \text{and} \quad \alpha \to 0 = \begin{cases} 0 & \text{if } \alpha \neq 0 \\ 1 & \text{if } \alpha = 0 \end{cases} \tag{2}$$

and

$$\begin{aligned} &\neg\,\alpha = 0 \text{ if and only if } \alpha \neq 0 \\ &\neg\,\alpha = 1 \text{ if and only if } \alpha = 0 \\ &\neg\,\neg\,\alpha = 1 \text{ if and only if } \alpha \neq 0 \\ &\alpha \wedge \neg\alpha = 0 \end{aligned} \tag{3}$$

In particular,

$$(4)\qquad \begin{array}{l} \neg\neg\alpha \neq \alpha \text{ if } 0 < \alpha < 1 \\ \neg 0 = 1,\ \neg 1 = 0,\ \neg\neg 0 = 0,\ \neg\neg 1 = 1 \\ \neg\,\alpha \vee \beta = \begin{cases} 1 & \text{if } \alpha = 0 \\ \beta & \text{if } \alpha \neq 0 \end{cases} \end{array}$$

Hence $\alpha \to \beta$ and $\neg\,\alpha \vee \beta$ are not identical in general. The following, however, holds for all $\alpha, \beta \in I$

$$\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$$
$$\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$$

Though the lattice operations $\vee,\ \wedge,\ \to,\ \neg$ on I are not Boolean, their restrictions to the sublattice $\{0, 1\}$ are Boolean.

2.2. Semi-modules over I. The lattice operations define a semi-ring structure on $I = [0, 1]$ with neutral element 0 and multiplicative unit 1 by

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

Call *I-space* any semi-module over I. Many of the definitions familiar from vector spaces over the field of real numbers carry over to I-spaces. In particular, $b_1, \dots, b_n \subset V$ is a *basis* of V if every vector can be written in a unique way as a linear combination of the vectors $b_1, \dots, b_n$. An I-space is n-dimensional if it has a basis of cardinality n. The dimension is unique. In fact, a stronger property holds.

LEMMA 1. *Every I-space has at most one finite set of basis vectors.*

PROOF. Assume that each of the strings of m vectors $a_1, \dots, a_m$ and n vectors $b_1, \dots, b_n$ forms a basis of V. Then there are scalars β_{il} and α_{lj} such that

$$a_i = \sum_{l=1}^{n} \beta_{il} b_l \text{ and } b_l = \sum_{j=1}^{m} \alpha_{lj} a_j,\ 1 \leq i \leq m,\ 1 \leq l \leq n\,.$$

Hence, for for each i

$$a_i = \sum_{l=1}^{n} \beta_{il} (\sum_{j=1}^{m} \alpha_{lj} a_j) = \sum_{j=1}^{m} (\sum_{l=1}^{n} \beta_{il} \alpha_{lj}) a_j\,.$$

It follows that $\sum_{l=1}^{n} \beta_{il}\alpha_{li} = 1$ and that $\sum_{l=1}^{n} \beta_{il}\alpha_{lj} = 0$ whenever $j \neq i$. The latter equality implies $\beta_{il}\alpha_{lj} = 0$ for $1 \leq l \leq n$, provided $j \neq i$, and the former that there is an l_i such that $\beta_{il_i} = \alpha_{l_i i} = 1$. Hence $\alpha_{l_i j} = 0$ whenever $j \neq i$, because $\beta_{il_i} = 1$ and the equality $\beta_{il_i}\alpha_{l_i j} = 0$ holds for $j \neq i$. Therefore $b_{l_i} = a_i$. The map $i \mapsto l_i$ is necessarily one-one, because $a_1, \dots, a_m$ are different from one another. Therefore, $\{a_1, \dots, a_m\} \subseteq \{b_1, \dots, b_n\}$. A similar proof, starting with b_l shows that $\{b_1, \dots, b_n\} \subseteq \{a_1, \dots, a_m\}$. ⊣

All I-spaces are assumed to be finite dimensional from now on. Hence, the vectors of an I-space form a super-cube I^n in the n-dimensional real Hilbert space.

The tensor product, $v \otimes w$ or $|v\rangle\langle w|$, the inner product, $\langle v|w\rangle$, and orthogonality are defined as usual. In particular, if V and W are I-spaces of basis $\{a_1, \dots, a_m\}$ and $\{b_1, \dots, b_n\}$ respectively then $\{a_i \otimes b_j, 1 \leq i \leq m,\ 1 \leq j \leq n\}$ is a basis of

$V \otimes W$. Moreover, two vectors are orthogonal in an I-space exactly when they are orthogonal in the corresponding real Hilbert space.

Vectors are partially ordered by the product order of I for which the *null vector* $\overrightarrow{0}$ is the smallest and the *full vector* $\overrightarrow{1}$ the largest element. Accordingly, the logical connectives $\neg, \wedge, \vee, \rightarrow$ defined on I 'lift' to the vectors in V, that is to say *vector negation*, *vector conjunction*, *vector disjunction* and *vector implication* are defined coordinate by coordinate $\neg \sum_{i=1}^{n} \alpha_i b_i = \sum_{i=1}^{n} (\neg \alpha_i) b_i$, $(\sum_i \alpha_i b_i) \wedge (\sum_j \beta_j b_j) = \sum_i (\alpha_i \wedge \beta_i) b_i$ etc.

LEMMA 2. *The logical connectives define a distributive, implication complemented lattice structure on V such that*

$$\begin{array}{rcl} \neg(v_1 \vee v_2) & = & \neg v_1 \wedge \neg v_2 \\ \neg(v_1 \wedge v_2) & = & \neg v_1 \vee \neg v_2 \\ v \wedge \neg v & = & \overrightarrow{0} \\ \neg v & = & v \rightarrow \overrightarrow{0} \\ \langle w | v \rangle = 0 & \Leftrightarrow & w \leq \neg v \\ u \leq v \rightarrow w & \Leftrightarrow & u \wedge v \leq w \\ v \rightarrow w = \overrightarrow{1} & \Leftrightarrow & v \leq w\,. \end{array} \tag{5}$$

In particular, vectors are orthogonal if and only if their conjunction is the null vector. Moreover, the following equivalences hold

$$\neg\neg v = v \Longleftrightarrow v \vee \neg v = \overrightarrow{1} \Longleftrightarrow \textit{the coordinates of } v \textit{ are } 0 \textit{ or } 1\,. \tag{6}$$

The vectors characterised by the equivalent properties (6) are called *intrinsic* vectors. Their logical properties are the same in I-spaces and in Hilbert spaces.

The definitions of subspaces, linear maps and their associated matrices also remain unchanged. Some of their familiar properties remain valid, others are new.

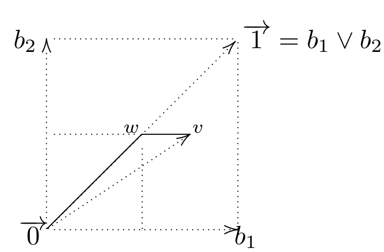
A property remaining valid is that a linear map is determined by its values on the basis vectors. An example is vector conjunction $\wedge : V \times V \rightarrow V$, which is bilinear by Lemma 2. It defines a unique linear map $\wedge : V \otimes V \rightarrow V$ satisfying for arbitrary basis vectors b_i, $b_j \in V$

$$b_i \wedge b_j = b_i \text{ if } i = j \text{ and } b_i \wedge b_j = \overrightarrow{0} \text{ if } i \neq j\,.$$

New properties are:
- All linear maps are monotone increasing.
- The the *span of* v, i.e. subspace generated by an arbitrary vector v,

$$G_v = \{\alpha\, v \,:\, \alpha \in I\}$$

is in general a broken line. It is a straight line if all non-zero coordinates of v are equal.



In the figure above, the broken solid line from $\overrightarrow{0}$ through w to v represents G_v, whereas G_w is the straight solid line $\overrightarrow{0}$ to w. Note that a vector with a straight span is a scalar multiple of an intrinsic vector.

The last two lemmas of this subsection are essential for connecting sentence semantics and vector models.

Lemma 3. *Every linear map commutes with disjunction. A linear map $f : V \to W$ that maps distinct basis vectors to orthogonal vectors commutes with conjunction.*

Proof. The first affirmation follows from linearity. For the proof of the second affirmation, note that $f(a_i) \wedge f(a_i) = f(a_i) = f(a_i \wedge a_i)$ holds because conjunction is idempotent. Next, assume that a_i and a_j are different basis vectors of V. Then $f(a_i \wedge a_j) = f(\overrightarrow{0}) = \overrightarrow{0}$, because the conjunction of distinct basis vectors is the null vector. Moreover, $f(a_i) \wedge f(a_j) = \overrightarrow{0}$, because $f(a_i)$ and $f(a_j)$ are orthogonal. General commutation follows from the linearity of f and distributivity of conjunctions over disjunctions. ⊣

Lemma 4. *A linear map commutes with negation if and only if it preserves the full vector $\overrightarrow{1}$ and maps distinct basis vectors to orthogonal vectors.*

Proof. Let $f : V \to W$ be a linear map, $a_1, \ldots, a_n$ the basis vectors of V, $b_1, \ldots, b_m$ those of W, $f(a_i) = \sum_{j=1}^m \beta_{ji} b_j$, $v = \sum_{i=1}^n \alpha_i a_i \in V$ and $K = \{i : \alpha_i \neq 0\}$.

Note that $\neg\alpha_i = 1$ for $i \notin K$ and $\neg\alpha_i = 0$ for $i \in K$. Hence, $\neg v = \sum_{i=1}^n (\neg\alpha_i) a_i = \sum_{i \notin K} a_i$ and therefore

$$f(\neg v) = \sum_{i \notin K} f(a_i) = \sum_{i \notin K} \sum_{j=1}^m \beta_{ji} b_j = \sum_{j=1}^m (\sum_{i \notin K} \beta_{ji}) b_j$$

$$\begin{aligned}\neg f(v) = \neg f(\textstyle\sum_{i=1}^n \alpha_i a_i) &= \neg \textstyle\sum_{i=1}^n \alpha_i (\sum_{j=1}^m \beta_{ji} b_j) = \neg \sum_{j=1}^m (\sum_{i=1}^n \alpha_i \beta_{ji}) b_j \\ &= \neg \textstyle\sum_{j=1}^m (\sum_{i \in K} \alpha_i \beta_{ji}) b_j = \sum_{j=1}^m \neg(\sum_{i \in K} \alpha_i \beta_{ji}) b_j \,.\end{aligned}$$

Hence

$$f(\neg v) = \neg f(v) \text{ if and only if } \sum_{i \notin K} \beta_{ji} = \neg \sum_{i \in K} \alpha_i \beta_{ji}, \text{ for } 1 \leq j \leq m \,. \tag{7}$$

Assume that f commutes with negation. Then $f(\overrightarrow{1}) = f(\neg \overrightarrow{0}) = \neg f(\overrightarrow{0}) = \neg \overrightarrow{0} = \overrightarrow{1}$. To prove the second condition, let $1 \leq k \leq n$ and apply the preceding equivalence to $v = a_k$. In this case, $K = \{k\}$, $\alpha_k = 1$ and $\alpha_i = 0$ for $\neq k$. The assumption implies that

$$\neg\beta_{jk} = \sum_{i \neq k} \beta_{ji}, \text{ for } 1 \leq j \leq m \,.$$

The orthogonality of $f(a_k)$ and $f(a_i)$ for $i \neq k$ follows, because

$$\langle f(a_k) | f(a_i) \rangle = \sum_j \beta_{jk}\beta_{ji} \leq \sum_j \beta_{jk} (\sum_{l \neq k} \beta_{jl}) = \sum_j \beta_{jk} \neg\beta_{jk} = 0 \,.$$

Conversely, assume that the two conditions are satisfied. The first condition implies

$$\overrightarrow{1} = f(\overrightarrow{1}) = f(\sum_{i=1}^n a_i) = \sum_{j=1}^m (\sum_{i=1}^n \beta_{ji}) b_j \,,$$

that is to say $\sum_{i=1}^{n} \beta_{ji} = 1$ for $1 \le j \le m$. Hence for every $1 \le j \le m$ there exists i_j such that $\beta_{ji_j} = 1$. If $k \ne i_j$ then $\beta_{jk} = 0$, because $f(a_{i_j})$ and $f(a_k)$ are orthogonal. Let $v = \sum_{i=1}^{n} \alpha_i a_i \in V$. To derive the righthand equality of (7), distinguish between two cases. If $i_j \notin K$, then $\sum_{i \notin K} \beta_{ji} = 1$ and $\sum_{i \in K} \alpha_i \beta_{ji} = 0$. Hence the righthand equality holds in this case. If $i_j \in K$, then $\sum_{i \notin K} \beta_{ji} = 0$ and $\neg \sum_{i \in K} \alpha_i \beta_{ji} = \neg \alpha_{i_j} = 0$. The last equality holds by (3), because $\alpha_{i_j} \ne 0$ by definition of K. ⊣

2.3. Projectors of I-spaces. Let V be an I-space with basis $\{b_1, \ldots, b_n\}$. A linear map $P : V \to V$ is a *projector* if it is idempotent and self-adjoint, that is to say

$$P(P(v)) = P(v) \text{ and } \langle P(v)|w\rangle = \langle v|P(w)\rangle, \text{ hold for all } v, w \in V\,.$$

The latter condition holds if and only if the matrix of P is equal to its transpose. Examples are the projectors $P_i,\ 1 \le i \le n$, where P_i sends b_i to itself and every other basis vector to the null vector. Clearly, $P_i = |b_i\rangle\langle b_i|$.

Every projector P determines a subspace E_P, the so-called *invariance space* of P, namely

$$E_P = \{w : w = P(w)\} = P(V)\,.$$

Note that E_P is generated by the vectors $P(b_i),\ 1 \le i \le n$, and that the vector $P(\overrightarrow{1}) = P(b_1) + \cdots + P(b_n)$ is its largest element.

Following [15], define the quantum connectives on the set of projectors thus

$$\neg P = P^{\perp},\ P \vee Q = P + Q,\ P \wedge Q = P \circ Q,\ P \to Q = \{u : Q(P(u)) = P(u)\}\,,$$

where the letters P, Q refer to projectors or, equivalently, their invariance spaces.

A vector $v = \alpha_1 b_1 + \cdots + \alpha_n b_n$ defines two projectors in an I-space. One is the dyad $|v\rangle\langle v| = v \otimes v$, for which the the invariance space is the span G_v, like in a Hilbert space. The other one is the projector P_v defined on basis vectors thus

$$P_v(b_i) = \alpha_i b_i, \text{ for } 1 \le i \le n\,.$$

Then for $w = \beta_1 b_1 + \cdots + \beta_n b_n$,

$$P_v(w) = \alpha_1 \beta_1 b_1 + \cdots + \alpha_n \beta_n b_n = v \wedge w\,. \tag{8}$$

P_v is a linear map by definition. It is also a projector. Indeed, the associated square matrix (γ_{ij}) satisfies $\gamma_{ii} = \alpha_i$ and $\gamma_{ij} = 0$ for all $i,\ j \ne i$. Hence P_v is self-adjoint. It is idempotent, because scalar multiplication is idempotent.

Note that v is the largest vector that is invariant under P_v. Moreover, the subspace of vectors less or equal to v coincides with the invariance space of P_v

$$E_v := \{w : w \le v\} = E_{P_v} = \{w : w = P_v(w)\}\,.$$

Call P_v the *order projector* and E_v the *order space* of v. It includes the span G_v of v. Span and order space coincide if $v = \beta b_i$ for some basis vector b_i. Otherwise, the inclusion is strict. Note that $P_i = P_{b_i},\ 1 \le i \le n$. Clearly,

$$E_v \subseteq E_w \Longleftrightarrow v \le w\,.$$

THEOREM 1. *The map $v \mapsto P_v$ is a one-one lattice homomorphism from V into the quantum lattice of projectors defined on V. Its range is the subspace generated by the projectors $P_1, \ldots, P_n$.*

PROOF. We must prove that

$$E_v{}^{\perp} = E_{\neg v},\ E_v \vee E_w = E_{v+w},\ E_v \wedge E_w = E_{v\wedge w},\ E_v \to E_w = E_{v\to w}\,.$$

The first equality follows from (5). The second follows immediately from the definitions. To see the third equality, note that $P_w(P_v(u)) = w \wedge v \wedge u = P_{w\wedge v}(u)$ for all u. Therefore, $P_v \circ P_w = P_{v\wedge w}$. Finally, recall that $E_v \to E_w = \{u : E_w(E_v(u)) = E_v(u)\}$. Then $u \in E_{v\to w}$ if and only if $u \leq v \to w$ if and only if $u \wedge v \leq w$ by (5). By the preceding equality, $E_w(E_v(u)) = E_v(u)$ if and only if $w \wedge v \wedge u = v \wedge u$. The latter equality is equivalent to $u \wedge v \leq w$, which terminates the proof. ⊣

Of particular interest are the intrinsic vectors. The order subspace corresponding to a vector of the form $a_{i_1} + \cdots + a_{i_m}$ is generated by the set of basis vectors $\{a_{i_1}, \ldots, a_{i_m}\}$ and the associated projector $P_{i_1} + \cdots + P_{i_m}$ is also a projector in the Hilbert space. Its value for an arbitrary vector $v \in V$ is the same in V and in the Hilbert space. From now on, a vector is identified with its associated order subspace/projector. In particular, an intrinsic subspace/projector is the order order subspace/projector of an intrinsic vector.

§3. Concept Spaces, Logical Properties. Compositionality of meaning is closely linked to logic in natural language. The quantum logic of tensor products of 2-dimensional spaces has additional properties that allow a smooth transfer from sentence truth to concept truth as we shall see in Section 6.

Given a non-empty set $P = p_1, \ldots, p_d$, let $C(p_i)$ be a 2-dimensional I-space and denote $p_{i\top}, p_{i\perp}$ its two basis vectors. The *concept space defined by* P is the tensor product

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

Each 2-dimensional factor of a concept space stands for a *basic concept.* The choice of the basic concepts is highly variable. For a set of documents, the most relevant words, a *point of view*, make up the basic concepts. Another choice consists of the most frequent words. Features or attributes of a classification system may constitute a set of basic concepts. The key-words of Roget's (or the speaker's mental) thesaurus provide another example of basic concepts.

In a classification system, $p_{i\top}$ stands for 'feature p_i, yes' and $p_{i\perp}$ for 'feature p_i, no'. Hence, a basis vector of $C(P)$ constitutes a choice between a yes and no answer for each basic concept, i.e. there is an $f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\perp}\}$ such that

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

The concept space is the linguistic analogue to the compound system in quantum logic where the building blocks are 2-dimensional complex Hilbert spaces, see [1]. In quantum mechanics, p_i may stand for the spin of particles, the basis vector $p_{i\top}$ for upward spin and the basis vector $p_{i\perp}$ for downward spin.

A concept space, due to the fine-grained structure of its basis vectors, can be viewed both as a space of propositions and as a space of events. Some notations concerning particular vectors will be helpful to establish this duality.

The first concerns vectors which are generalisations of basis vectors. Let $K = \{i_1, \ldots, i_k\}$ be a subset of $\{1, \ldots, d\}$ such that $i_1 < \cdots < i_k$. Call

$g \in \prod_{i=1}^{d} \{p_i, \neg p_i, \overrightarrow{1}\}$ *a partial choice on* K if

$$g(i) = \overrightarrow{1} \text{ if and only if } i \notin K, \text{ for } 1 \le i \le d\,.$$

The *partial choice vector* associated to g is

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

Let $q_{i_j} = g(i_j)$ for $1 \le j \le k$, where g is a partial choice on $\{i_1, \ldots, i_k\}$. Then

$$v_g = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes \overrightarrow{q_{i_1}} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes \overrightarrow{q_{i_k}} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1}$$

Clearly, if g is total, i.e. if $K = \{1, \ldots, d\}$, then v_g is a basis vector of $C(P)$.

Lemma 5. *Every partial choice vector is an intrinsic vector.*

Proof. Let g be the partial choice on $K = \{i_1, \ldots, i_k\}$ satisfying $g(i_j) = q_{i_j} \in \{p_{i_j\top}, p_{i_j\bot}\}$ for $1 \le j \le k$. Define the subset of total choices

$$G = \left\{ f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\} : \ f(i_j) = q_{i_j} \text{ for } 1 \le j \le k \right\}.$$

Then show

$$\overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes \overrightarrow{q_{i_1}} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes \overrightarrow{q_{i_k}} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \sum_{f \in G} b_f\,. \tag{9}$$

by induction on d. The case $d = 1$ is trivial. For the induction step, use the bilinearity of the tensor product. ⊣

Corollary 1. *The full vector of* $C(P)$ *is the tensor product of the full vectors of each factor*

$$\overrightarrow{1} = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1}\,.$$

Proof. Recall that $\overrightarrow{1} = \sum_{f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\}} b_f$. Hence, the equality to show is the particular case of (9). ⊣

The simplest among the partial choice vectors are the *elementary vectors*

(10)

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

The subspace/projector corresponding to an elementary vector will also be called elementary.

Elementary vectors may be viewed as elementary propositions or elementary events. The next two lemmas show that arbitrary events, i.e. unions of conjunctions of elementary events coincide with the intrinsic vectors of $C(P)$.

Lemma 6. *The elementary vectors generate the Boolean algebra of intrinsic vectors.*

Proof. Every intrinsic vector is a sum of some basis vectors. Remains to show that

- *elementary vectors are closed under negation*

and that

- *every basis vector is a conjunction of elementary vectors.*

The first assertion follows from

$$\neg\overrightarrow{p_i} = \overrightarrow{\neg p_i} \text{ and } \neg(\overrightarrow{\neg p_i}) = \overrightarrow{p_i} \tag{11}$$

It suffices to prove the lefthand equality. The righthand equality follows by (6).

For every total choice $f \in \prod_{i=1}^{d}\{p_{i\top}, p_{i\bot}\}$, define the two scalars

$$\alpha_{if} = \begin{cases} 1 & \text{if } f(i) = p_{i\top} \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \beta_{if} = \begin{cases} 1 & \text{if } f(i) = p_{i\bot} \\ 0 & \text{else} \end{cases} .$$

Clearly, $\alpha_{if} = \neg\beta_{if}$. Note that $\alpha_{if} = 1$ if $f \in \{g : g(i) = p_{i\top}\}$ and $\alpha_{if} = 0$ if $f \notin \{g : g(i) = p_{i\top}\}$. Therefore, by (9), $\overrightarrow{p_i} = \sum_f \alpha_{if} b_f$ and $\overrightarrow{\neg p_i} = \sum_f \beta_{if} b_f$. The property now follows from the definition of vector negation.

The second assertion follows from the more general lemma below. ⊣

Lemma 7. *Every partial choice vector is a conjunction of elementary vectors.*

Proof. Let $q_{i_j} \in \{p_{i_j\top}, p_{i_j\bot}\}$ be given for $1 \le j \le k$. We must show that

$$\overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes q_{i_1} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes q_{i_k} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \overrightarrow{q_{i_1}} \wedge \cdots \wedge \overrightarrow{q_{i_k}} . \tag{12}$$

For every total choice f and every $j \le k$, define the scalar

$$\chi_{i_j f} = \alpha_{i_j f} \text{ if } q_{i_j} = p_{i_j\top} \text{ and } \chi_{i_j f} = \beta_{i_j f} \text{ if } q_{i_j} = p_{i_j\bot} .$$

From the definitions of $\alpha_{i_j f}$ and $\beta_{i_j f}$ follows that

$$\chi_{i_j f} = 1 \text{ if } f(i_j) = q_{i_j} \text{ and } \chi_{i_j f} = 0 \text{ otherwise.}$$

Hence, the product of these scalars satisfies

$$\chi_{i_1 f} \ldots \chi_{i_k f} = 1 \text{ if } f(i_j) = q_{i_j} \text{ for } 1 \le j \le k, \text{ and } \chi_{i_1 f} \ldots \chi_{i_k f} = 0 \text{ else} .$$

Now compute

$$\overrightarrow{q_{i_1}} \wedge \cdots \wedge \overrightarrow{q_{i_k}} = (\sum_f \chi_{i_1 f} b_f) \wedge \cdots \wedge (\sum_f \chi_{i_k f} b_f) = \sum_f (\chi_{i_1 f} \ldots \chi_{i_k f}) b_f$$

using distributivity and the fact that $b_f \wedge b_f = b_f$ and $b_f \wedge b_{f'} = \overrightarrow{0}$ for $f \neq f'$. Finally, for $G = \{f : f(i_j) = q_{i_j} \text{ for } 1 \le j \le k\}$

$$\sum_f (\chi_{i_1 f} \ldots \chi_{i_k f}) b_f = \sum_{f \in G} b_f .$$

Equality (12) follows by (9). ⊣

The following corollary reformulates Lemma (6) in terms of projectors.

Corollary 2. *Elementary projectors are closed under negation. The projectors corresponding to basis vectors are conjunctions of elementary projectors. The elementary projectors generate the Boolean algebra of intrinsic projectors.*

This result can be improved

Theorem 2. *For every set $P = \{p_1, \ldots, p_d\}$, the map $p \mapsto \overrightarrow{p}$ extends to an isomorphism from the free Boolean algebra $B(P)$ generated by P onto the sublattice of intrinsic subspaces of $C(P)$.*

Proof. The intrinsic vectors of $C(P)$ have coordinates 0 or 1 and therefore form a Boolean algebra. Hence, the map $p \mapsto \overrightarrow{p}$ extends to a unique Boolean homomorphism from $B(P)$ into the lattice of intrinsic subspaces. According to a classical theorem, see for example [6], the free Boolean algebra $B(P)$ is isomorphic to the lattice of subsets of $\prod_{i=1}^{d}\{0,1\}$ generated by the following subsets

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

⊣

Hence, the lattice of projectors of a tensor product of 2-dimensional spaces includes classical propositional calculus[1] namely the projectors corresponding to the intrinsic vectors/subspaces/projectors. In particular, intrinsic vectors identify with propositions. From now on, we use *concept* or *concept vector/projector* instead of 'intrinsic vector/projector of a concept space'.

By Theorem 2, one can use *induction on the complexity of propositions* in definitions and proofs concerning intrinsic subspaces. Propositional complexity creates a somewhat unusual hierarchy on subspaces: The subspace of $\overrightarrow{p}_i$ has complexity 0 but dimension 2^{d-1}, whereas the one-dimensional subspaces of the basis vectors have complexity $d-1$. And this holds also in a Hilbert space, because intrinsic subspaces are defined by the same basis vectors in an I-space and a Hilbert space.

§4. Concept Spaces, Classification Systems. Classification systems relate sets of individuals to features. The usual definitions, when transferred to a concept space give rise to a representation of each individual by a concept vector, namely the smallest concept satisfied by the individual. This fact is essential when translating sentences to concepts.

4.1. Basic definitions. A classification system $(B, P, \models)$ consists of a set of *individuals* B, a set of *features* $P = \{p_1, \ldots, p_d\}$ and a binary *classification relation* $\models \subseteq B \times P$.

Extend the classification relation from features to concept vectors in $C(P) = C(P_1) \otimes \ldots \otimes C(P_d)$ by induction on the complexity of propositions

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

Extend the classification relation from individuals to non-empty sets of individuals

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

We shall see in Section 6 that pregroup grammars define a 'natural' classification system inherent to pregroup semantics.

A *class* is any subset of B defined by some concept vector v

$$\mathcal{C}(v) = \{x \in B \ : \ x \models v\}\,.$$

[1] the free Boolean algebra identifies with propositional calculus modulo equiderivability.

Clearly, $\mathcal{C}(v)$ is the largest subset of B satisfying v. The next lemma is proved by induction on the complexity of intrinsic projectors.

LEMMA 8. *The map $v \mapsto \mathcal{C}(v)$ is a homomorphism from the lattice of intrinsic projectors of $C(P)$ into the Boolean algebra of subsets of individuals.*

Every individual and every subset of individuals determines a basis vector in C(P). Indeed, for $x \in B$ and $1 \leq i \leq d$, let $q(x)_i \in C(p_i)$ be given by

$$q(x)_i = p_{i\top} \text{ if } x \models p_i$$
$$q(x)_i = p_{i\perp} \text{ if } x \not\models p_i\,.$$

Define

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

For $Y \subseteq B$ define

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

LEMMA 9. *The vector v_x is the smallest concept satisfied by x. The same holds if x is replaced by a set Y. $\mathcal{C}(v_Y)$ is the smallest of all classes including Y. In particular, $x \in \mathcal{C}(v_x)$ and $Y \subseteq \mathcal{C}(v_Y)$.*

PROOF. It follows from the results in the previous section that v_x and v_Y are indeed concept vectors. Use induction on the propositional complexity of c to show that

$$x \models c \text{ if and only if } v_x \leq c$$

The equivalence for a subset Y

$$Y \models c \text{ if and only if } v_Y \leq c \tag{13}$$

is an immediate consequence of the equivalence for individuals. For the last affirmation, it suffices to show that

$$Y \models c \Leftrightarrow Y \subseteq \mathcal{C}(c) \Leftrightarrow \mathcal{C}(v_Y) \subseteq \mathcal{C}(c),$$

which follows from (9). ⊣

One consequence of the lemma above is that satisfaction in a classification system coincides with the conditional logic for intrinsic 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 individual x by the chosen set of key-words and the same holds if x is replaced by a set Y. Semantic vector models ignore words designating individuals and work rather with words referring to sets. For example, the set Y may be the extension of a compound noun phrase. In Section 6, we show how v_Y is computed from the vectors standing for the words.

Information retrieval systems use the inner product of the Hilbert space to describe similarity of words. If words are represented by intrinsic vectors, the inner product in the I-space takes only the two values 0 and 1. It would be tempting to think that the inner product captures satisfaction. This is not the case for arbitrary sets. Satisfaction of a concept by an individual can be defined by the inner product, but not satisfaction by a set with more than one element.

LEMMA 10. *For every non-empty subset Y and concept vector c*

$$\begin{aligned} \exists x(x \in Y \wedge x \models c) &\Leftrightarrow \langle v_Y | c\rangle = 1 \\ Y \models c &\Rightarrow \langle v_Y | c\rangle = 1, \end{aligned}$$

but the converse of the latter implication does not hold in general.

PROOF. First, we show the equivalence in the particular case where $Y = \{x\}$, namely that

$$x \models c \Leftrightarrow \langle v_x | c\rangle = 1 \tag{14}$$

by induction on the Boolean structure of the concept vector c. Indeed, if c has complexity 0 then $c = \overrightarrow{p_i}$ for some p_i. Assume first that x satisfies $\overrightarrow{p_i}$. Then the i-th factor of the basis vector v_x is $p_{i\top}$. Therefore, $\langle v_x|\overrightarrow{p_i}\rangle = \langle v_x|\sum_{f(i)=p_{i\top}} b_f\rangle = \sum_{f(i)=p_{i\top}}\langle v_x|b_f\rangle \geq \langle v_x|v_x\rangle = 1$. Next assume that x does not satisfy $\overrightarrow{p_i}$. Then $v_{xi} = p_{i\bot}$ and therefore $\langle v_x|b_f\rangle = 0$ for all f satisfying $f(i) = p_{i\top}$. The equality $\langle v_x|\overrightarrow{p_i}\rangle = 0$ follows. For the induction step, use the linear map $v \mapsto \langle v_x|v\rangle$ from $C(P)$ to I, which commutes with disjunction, conjunction and negation by Lemmas 3 and 4.

The asserted properties and a counterexample for the converse implication follow from (14) and the equality $\langle v_Y|c\rangle = \sum_{x\in Y}\langle v_x|c\rangle$. ⊣

In the presence of a probability distribution on the event space generated by the p_i, i.e. the sublattice of concepts in $C(P)$, the value of the distribution for an event c may be interpreted as the probability that an individual in B satisfies c. A formal development is beyond the scope of this paper.

4.2. Polysemous Concept Spaces. Two different meanings of a polysemous word designate two disjoint subsets of individuals. The individuals designated by one meaning have no features in common with the individuals designated by another meaning. Otherwise said, polysemous words require a concept space per meaning. Each meaning of the word defines a set Y_i of individuals for each of them. The word is represented by the sum of its two meanings $v_{Y_1} + v_{Y_2} \in C(P_1) \oplus C(P_2)$.

A *polysemous concept space* is a finite sum

$$C(P_1) \oplus \cdots \oplus C(P_n),$$

where the P_i's are non-empty, pairwise disjoint sets and $n > 1$.

Given classification systems $(B_i, P_i, \models_i)$ on the summands $C(P_i)$ with pairwise disjoint sets B_i, define satisfaction for elements x and subsets Y of $B_1 \cup \cdots \cup B_n$ in the polysemous space $C(P_1) \oplus \cdots \oplus C(P_n)$ by

$$\begin{gathered} x \models v \text{ if and only if } x \models_i v_i \text{ and } x \in B_i \\ Y \models v \text{ if and only if } v_{Y\cap B_i} \models_i v_i, \text{ for } 1 \leq i \leq n, \end{gathered}$$

where the vectors $v_i \in C(P_i)$ are determined by the equality $v = \sum_i v_i$.

The last lemma in this section gives a caveat: the negation operator of a polysemous concept space does not coincide with the sum of the negation operators of the summands.

LEMMA 11. *Assume P_1 and P_2 are disjoint sets. The maps $v \mapsto v + \overrightarrow{0}$ from $C(P_1)$ into $C(P_1) \oplus C(P_2)$ and $w \mapsto \overrightarrow{0} + w$ from $C(P_2)$ into $C(P_1) \oplus C(P_2)$ are* *linear order preserving embeddings that commute with the logical connectives* $\wedge$ *and* $\vee$*, but not with* $\neg$.

Proof. By Lemmas 3 and 4 $\dashv$

§5. Vector Semantics by Pregroup Grammars. In the following, the U, S, C are constants. Think of U as the *universe*, where the basis vectors of U are the *individuals* and sums of basis vectors identify with subsets of individuals. In particular, the null-vector $\overrightarrow{0}$ stands for the empty set and the full vector $\overrightarrow{1}$ stands for the set of all individuals. Linguistically, basis vectors correspond to singulars and sums of several basis vectors to plurals. S stands for the *sentence space*, its basis vectors are called *truth-values*. We assume, that the logic is two-valued, i.e. S has exactly two basis vectors, namely $\top$ for '*true*' and $\bot$ for '*false*'. The null-vector $\overrightarrow{0} \in S$ stands for 'no truth values', the full vector $\overrightarrow{1} \in S$ for 'all truth-values'. Hence $\overrightarrow{1} = \top + \bot$. Finally, C stands for some polysemous concept space $C(P_1) \oplus \cdots \oplus C(P_k)$. Thinking of C as a semantic vector models, words that correspond to basis vectors are unambiguous and their meaning is irreducible to other meanings.

5.1. Technicalities. Like every other categorial grammar, a pregroup grammar has a lexicon, i.e. a finite list of *entries*, namely triples $w : T :: m$, where w is a word, T a type and m a *meaning* expression. The latter depends functionally on the pair $w : T$. It can be interpreted in any symmetric compact-*2*-category. A *type* is a string of simple types, where a *simple type* has one of the forms $x, y, \dots$ or $x^\ell, y^\ell, \dots$ or $x^r, y^r, \dots$. The types $x, y, \dots$ are called *basic types* and form a partially ordered set. They stand for grammatical notions. The types $x^\ell, y^\ell, \dots$ are called *left adjoints* and the $x^r, y^r, \dots$ *right adjoints*.

This description differs in two aspects from the original one in [8]. There, only pregroup dictionaries are considered where the entries are pairs $w : T$. The meaning expressions have been added explicitly, because a functional reading of pregroup types is not as obvious as that of higher order types. Moreover, compact bilinear logic also requires iterated left and right adjoints $x^{\ell\ell}$, $x^{\ell\ell\ell}, \dots$, x^{rr}, $x^{rrr}, \dots$. The restriction to dictionaries without iterated adjoint types, however, does not change the set of recognised languages nor the structure of the derivations, nor the semantic interpretation, [11].

Consider the following entries

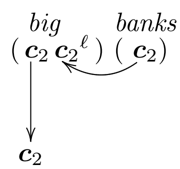
$$no\text{: } \boldsymbol{s}\boldsymbol{s}^\ell\boldsymbol{n}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{no}}} S \otimes S^* \otimes U \otimes U^*$$

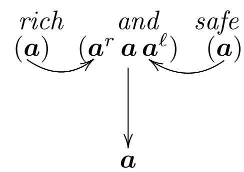
$$\begin{array}{llll} some: \boldsymbol{n}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{some}}} U \otimes U^* & & are: \boldsymbol{n}_2{}^r\boldsymbol{s}\boldsymbol{a}^\ell :: I \xrightarrow{\overline{\texttt{are}}} U^* \otimes S \otimes C^* \\ big: \boldsymbol{c}_2\boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{big}}} U \otimes U^* & & rich: \boldsymbol{a} :: I \xrightarrow{\overline{\texttt{rich}}} C \\ banks: \boldsymbol{c}_2 :: I \xrightarrow{\overline{\texttt{bank}}} U & & and: \boldsymbol{a}^r\boldsymbol{a}\boldsymbol{a}^\ell :: I \xrightarrow{\overline{\texttt{and}_\texttt{c}}} C^* \otimes C \otimes C^* \end{array}$$

The basic types $\boldsymbol{c}_2, \boldsymbol{n}_2, \boldsymbol{s}, \boldsymbol{a}$ stand for 'plural count noun', 'plural noun phrase', 'sentence', 'predicate', in that order. The only inequality in this set is $\boldsymbol{c}_2 < \boldsymbol{n}_2$. The basic types $\boldsymbol{c}_2, \boldsymbol{n}_2$ are interpreted by the I-space U, $\boldsymbol{s}$ by S and $\boldsymbol{a}$ by C. The left and right adjoint of a basic type are interpreted by the dual V^* of the space V of the basic type.

As usual, a vector V is identified with a linear map from I to V. For example, $\overline{\texttt{rich}} : I \to C$ is identified with $\texttt{rich} \in C$. The vectors $\overline{\texttt{big}}$, $\overline{\texttt{are}}$, $\overline{\texttt{and}_\texttt{C}}$ are the matrices of the linear maps $\texttt{big} : U \to U$, $\texttt{are} : U \otimes C \to S$, $\texttt{and}_C : C \otimes C \to C$, in that order. Due to the identification of linear maps $m : V \to W$ and vectors in $\overline{m} : I \to V^* \otimes W$, we refer to the meaning expressions in the lexicon either as vectors or as linear maps, whichever is more convenient. All meaning expressions are intrinsic vectors. Other properties of meanings are given in due course.

Syntactical analysis does not involve meaning vectors. By definition, a string of words $w_1 \dots w_n$ is *recognised* by a pregroup grammar if there are entries $w_1 : T_1 :: m_1, \dots,\ w_n : T_n :: m_n$ such that the concatenated type $T_1 \dots T_n$ reduces to a basic type. Reductions are the pregroup analogues for parsing tree. They identify with oriented planar graphs, for example

$r_1 =$ 

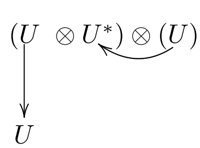
$r_2 =$ 

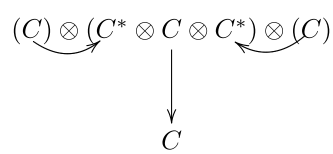
Reductions involve vertical links and underlinks only. The latter correspond to evaluation in higher order logic and to co-units of adjunction in category theory. The links are compatible with the partial order of basic types, i.e. $x \to y$ implies $x \le y$. By convention, the tail of the underlink is always at a basic type, a 'filler', its head at an adjoint type, a 'hole'. If the head is to the left of the tail, the adjoint must be a left adjoint. The same holds if 'left' is replaced by 'right'.

The representation of reductions by graphs is based on the categorical characterisation of compact bilinear logic as the free compact *2*-category generated by the basic types, [13].

A *semantic interpretation* is a functor from the free compact *2*-category generated by the lexical entries into a symmetric compact *2*-category, for example the category of I-spaces or the category of real Hilbert spaces.

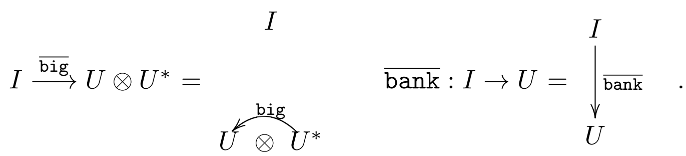
Hence, a reduction defines a linear map, which is represented by the graph obtained by replacing the simple types by the corresponding spaces. For example,

$r_1 =$ 

$r_2 =$  .

An underlink with tail at V and head at V^* denotes a linear map from $V^* \otimes V$ to I, which, due to the isomorphisms $V^* \otimes V \simeq V \otimes V^* \simeq V \otimes V^*$, is identified with inner product $\langle . | . \rangle_V$ of V. A vertical link stands for an identity. For example the reduction r_1 is $id_U \otimes \langle . | . \rangle_U : U \otimes U \otimes U \to U \otimes I \simeq U$. This identification of underlinks with the inner product holds in all categories of semi-modules.

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

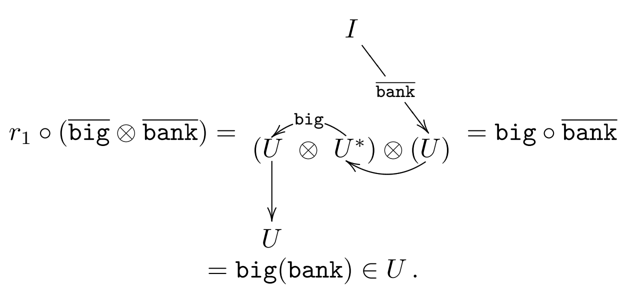


Overlinks correspond to abstraction in higher order logic and units of adjunction in category theory.

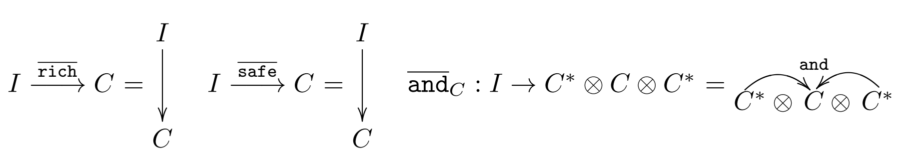
For every string of words $w_1 \ldots w_n$ and all lexical entries $w_1 : T_1 :: m_1, \ldots, w_n : T_n :: m_n$, the graph $m_1 \otimes \ldots \otimes m_n$ has only vertical links and overlinks. Not all strings of words, however, have a meaning, but grammatical strings do. Indeed, if the string $w_1 \ldots w_n$ is recognised by the grammar then there are lexical entries $w_1 : T_1 :: m_1, \ldots, w_n : T_n :: m_n$ and a reduction r of $T_1 \ldots T_n$ to a basic type. Define the *meaning vector* of the grammatical string $w_1 \ldots w_n$ as

$$m(w_1, \ldots, w_n) = r \circ (m_1 \otimes \ldots \otimes m_n)\,.$$

The categorical nature of the graphs r and $m_1 \otimes \ldots \otimes m_n$ implies that they both designate linear maps. Hence they can be composed. The result is computed graphically. Connect the graphs at there joint interface and follow the paths from top to bottom picking up the labels along the way. For example,

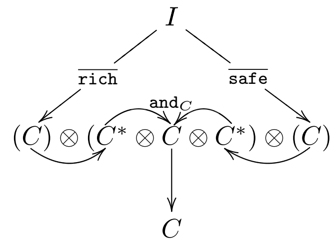


The graphs representing the word vectors in the string *rich and safe* are



Again, the meaning vector of the string *rich and safe* of type $\boldsymbol{a}\,\boldsymbol{a}^r\boldsymbol{a}\boldsymbol{a}^\ell\boldsymbol{a}$ is computed by composing the tensor product of word vectors with the reduction

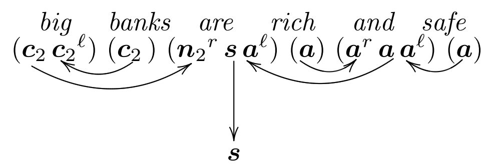
$$r_2 \circ (\overline{\texttt{rich}} \otimes \overline{\texttt{and}}_C \otimes \overline{\texttt{safe}}) =$$



$$= \texttt{and}_C \circ (\overline{\texttt{rich}} \otimes \overline{\texttt{safe}}) = \texttt{and}_C(\texttt{rich}, \texttt{safe}) \in C\,.$$

Finally, the reduction of the sentence *big banks are rich and safe* is

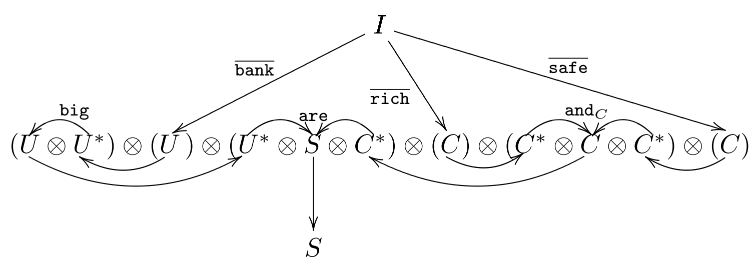
$$r =$$



.

Composing the corresponding linear map with the tensor product of the word vectors we obtain the meaning vector of the sentence

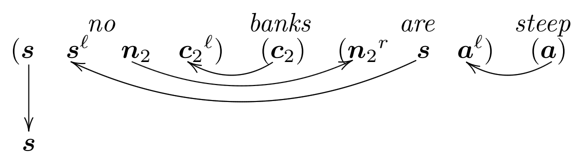
$$r \circ (\overline{\texttt{big}} \otimes \overline{\texttt{bank}} \otimes \overline{\texttt{are}} \otimes \overline{\texttt{rich}} \otimes \overline{\texttt{and}} \otimes \overline{\texttt{safe}}) =$$



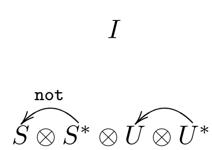
$$= \texttt{are}(\texttt{big} \circ \overline{\texttt{bank}}) \otimes (\texttt{and}_C \circ (\overline{\texttt{rich}} \otimes \overline{\texttt{safe}})))$$
$$= \texttt{are}(\texttt{big}(\texttt{bank}), \texttt{and}_C(\texttt{rich}, \texttt{safe})) \in S\,.$$

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

- find the reduction of the sentence

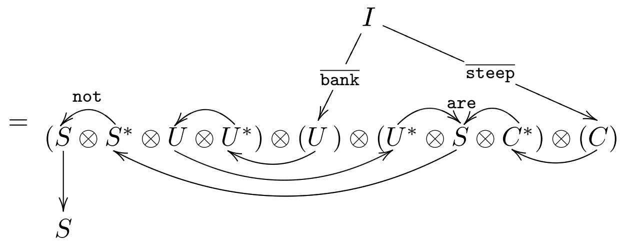


- draw the graph of the meaning vector $\overline{\texttt{no}} = \overline{\texttt{not} \otimes \texttt{id}_\texttt{U}} : T \to S \otimes S^* \otimes U \otimes U^*$



- compose the tensor product of the word vectors with the reduction and 'walk' the graph

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



$$= \texttt{not}(\texttt{are}(\texttt{bank}, \texttt{steep}))\,.$$

To sum up: All grammatical strings are interpreted by variable free expressions formed from linear maps or vectors introduced by words in the lexicon.

5.2. 'Noise' and Logic. The probabilistic approach to natural language ignores 'noise' words. They contribute, however, significantly to the meaning of a string of words, because they have a logical content. Below follow some examples how a pregroup grammar captures this logical content.

The 'noise' word *and* is polymorph. It has, among others, the types

$$\begin{array}{l} \textit{and}\colon \boldsymbol{a}^r \boldsymbol{a} \boldsymbol{a}^\ell :: \overline{\texttt{and}_\texttt{C}} : I \to C^* \otimes C \otimes C^* \\ \textit{and}\colon \boldsymbol{s}^r \boldsymbol{s} \boldsymbol{s}^\ell :: \overline{\texttt{and}}_S : I \to S^* \otimes S \otimes S^*\,. \end{array}$$

The logical content of *and* depends on the context in which it occurs and so does the type that the parsing algorithm chooses for it. The logical content varies with the type and so do the properties characterising the corresponding meaning vector.

The linear map $\texttt{and}_C : C \otimes C \to C$ is the unique linear map satisfying for all basis vectors c, c'

$$\texttt{and}_C(c \otimes c') = c \wedge c'\,.$$

Similarly, the *sentence conjunction* $\texttt{and}_S : S \otimes S \to S$, is the unique linear map with the following values for the basis vectors of $S \otimes S$

$$\texttt{and}_S(\top \otimes \top) = \top,\ \texttt{and}_S(\bot \otimes \top) = \texttt{and}_S(\top \otimes \bot) = \texttt{and}_S(\bot \otimes \bot) = \bot\,.$$

Linearity implies $\texttt{and}_S(\overrightarrow{0}) = \overrightarrow{0}, \texttt{and}_S(\overrightarrow{1}) = \bot + \top = \overrightarrow{1}$. Note that sentence conjunction differs from concept conjunction in S, because

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

Similarly, *sentence negation* $\texttt{not} : S \to S$ is defined explicitly by its values on the basis vectors

$$\texttt{not}(\top) = \bot,\ \texttt{not}(\bot) = \top\,.$$

Sentence negation coincides with concept negation on the basis vectors, because

$$\texttt{not}(\top) = \bot = \neg\top,\ \texttt{not}(\bot) = \top = \neg\bot\,,$$

but not on arbitrary vectors. Indeed, $\texttt{not}(\overrightarrow{1}) = \overrightarrow{1}$, whereas $\neg\overrightarrow{1} = \overrightarrow{0}$.

Determiners also constitute 'noise'. The logical content of the meaning vectors $\overline{\texttt{all}}$, $\overline{\texttt{some}}$ and $\overline{\texttt{no}}$ is captured by the following properties of the corresponding linear maps

$$\begin{array}{lll} \texttt{all}(x) & = x & \text{for all } x \in B \\ \texttt{some}(x) & = x \text{ or } = \overrightarrow{0} & \text{for all } x \in B \\ \texttt{no} & = \texttt{not} \otimes \texttt{id}\,. & \end{array}$$

The property concerning $\texttt{some} : U \to U$ holds for many maps, namely for all partial identities of U. By *partial identity* we mean any linear map $f : U \otimes W \to U$ satisfying for all $w \in W$

$$f(x,w) = x \text{ or } f(x,w) = \overrightarrow{0}, \text{ for all } x \in B\,.$$

Every partial identity $f : U \to U$ is a projector and the conjunction $f \wedge f' = f \circ f'$ of partial identities $f : U \to U$ and $f' : U \to U$ is again a partial identity satisfying

$$(f \wedge f')(x) = f(x) \wedge f'(x), \text{ for all } x \in B\,.$$

From now on, we assume that the linear map associated to an adjective in attributive position, say $\texttt{big} : U \to U$, is a partial identity.

Identify a set of individuals $Y \subseteq B$ with the sum of its basis vectors $\sum_{x \in Y} x$. In set-theoretical notation, the properties characterising $\texttt{all}$ and $\texttt{some}$ are

$$\texttt{all}(Y) = Y, \ \texttt{some}(Y) \subseteq Y, \text{ for all } Y \subseteq B\,.$$

Any partial identity $f : U \to U$ satisfies

(15) $$f(Y) = \{x \in B \,:\, f(x) = x\} \cap Y, \text{ for every } Y \subseteq B\,.$$

§6. Retrieving Information from Sentences. The examples discussed here are the sentences *All banks are steep*, *No banks are safe*, *Some banks are steep*, *Big banks are rich and safe.* To keep this paper within reasonable limits, only the case of unary predicates is developed in detail. The generalisation to the polysemous case also is omitted.

6.1. Sentence Truth. As a first step towards transferring the compositional logical semantics of sentences to the semantic vector model, take a new look at the predicates of first order logic. Verbs handle both individuals and sets, e.g. *Bill left* versus *Joe and Bill left.* Hence verbs behave like 'two-sorted predicates', i.e. predicates that accept elements and sets as arguments and return elements and sets as values.

In I-spaces, a *two-sorted predicate* $p : V \to S$ is a linear map that sends basis vectors of V to basis vectors of S. For example, the meaning $m : V \to S$ of a verb is a *two-sorted* predicate.[2] Obviously, a two-sorted predicate is defined both for an individual, i.e. a basis vector, and a subset of individuals, i.e. a sum of basis vectors. On basis vectors it behaves like a one-sorted predicate, i.e.

(16) $$p(x) = \top \text{ or } p(x) = \bot, \text{ for every basis vector } x \text{ of } V\,.$$

Two-sorted predicates are obviously closed under composition with the sentence connectives.

[2]This assumption is part of the particular properties to be mentioned in due course.

The value of a two-sorted predicate for a set may well be a set itself. The following lemma, which is a particular case of the Fundamental Lemma in [14], tells us when.

LEMMA 12. [Fundamental Property]
Let $p : V \to S$ be a predicate and Y a subset of basis vectors of V. Then the following holds

$$
\begin{array}{ll}
 & p(Y) = \overrightarrow{0} \Leftrightarrow Y = \emptyset \\
(17) & p(Y) = \top \;\; \Leftrightarrow \forall_x (x \in Y \Rightarrow p(x) = \top) \text{ and } Y \neq \emptyset \\
 & p(Y) = \bot \;\; \Leftrightarrow \forall_x (x \in Y \Rightarrow p(x) = \bot) \text{ and } Y \neq \emptyset \\
 & p(Y) = \overrightarrow{1} \Leftrightarrow \exists_{x \in Y} \exists_{y \in Y} \, (p(x) = \top \text{ and } p(y) = \bot)\,.
\end{array}
$$

PROOF. Write $Y = \sum_{x \in Y} x$. Then, by linearity, $p(Y) = \sum_{x \in Y} p(x)$. For the last equivalence, use $\overrightarrow{1} = \top + \bot$. ⊣

The two-sorted predicates include the one-sorted predicates $P \subseteq A$ of first order logic defined for elements only. Indeed, each one-sorted predicate P defines a two-sorted predicate p by

$$x \in P \text{ if and only if } p(x) = \top \text{ for all basis vectors } x\,.$$

There is another aspect distinguishing two-sorted predicates. The argument space V of a two-sorted predicate is not necessarily an iterated tensor product of the universe U. The notion of 'arity', however, involves arguments in U only. More precisely, a two-sorted predicate $p : V \to S$ is *n-ary* if V decomposes into n-times the factor U and a factor W that is a tensor product of the spaces I, C, S only. For example, $x \mapsto \texttt{are}(x, c) : U \to S$ is a unary two-sorted predicate for every basis vector $c \in C$.

6.2. Coherence Postulates. Words may occur in the lexicon with several meaning vectors, which are syntactically different versions of the same concept. They include in particular adjectives which are interpreted by an endomorphism of U in attributive position and a vector of C in predicative position. The same notation is used for both, the context permitting. For example, `big` refers in the expression $\texttt{are}(x, \texttt{big})$ to a vector of C and in the equality $\texttt{big}(x) = x$ to a map $\texttt{big} : U \to U$.

Intuition tells us that *big banks* are the same as *banks that are big*. Moreover, asserting that *banks are rich and safe* is the same as asserting that *banks are rich and banks are safe*.

For any adjective a, let `a` denote both the attributive interpretation $\texttt{a} : U \to U$ and the predicative interpretation $\texttt{a} : I \to C$. We postulate

Coherence Postulates

$$
\begin{array}{ll}
(18) & \texttt{a}(x) = x \quad \Leftrightarrow \quad \texttt{are}(x, \texttt{a}) = \top \\
 & \texttt{are}(x, c \wedge c') = \texttt{and}_S(\texttt{are}(x, c), \texttt{are}(x, c'))
\end{array}
$$

The equality of the vectors interpreting *big banks* and *banks that are big* is a particular case of the following lemma.

LEMMA 13. *For every adjective a and every subset of individuals $Y \subseteq B$ the following holds*

$$\texttt{a}(Y) = \texttt{who}(Y, \texttt{are}(Y, \texttt{a})).$$

PROOF. The relative pronoun *that* is interpreted by the partial identity $\mathtt{who} : U \times S \to U$ satisfying $\mathtt{who}(x, y) = x$ if $y = \top$ and $\mathtt{who}(x, y) = \overrightarrow{0}$ if $y = \bot$, see [14]. It satisfies

$$\mathtt{who}(Y, p(Y)) = \{x \in Y \,:\, p(x) = \top\}\ .$$

for every predicate $p : U \to S$ and every set of individuals Y. In particular for $p(x) = \mathtt{are}(x, \mathtt{a})$, we have $\mathtt{who}(Y, \mathtt{are}(Y, \mathtt{a})) = \{x \in Y \,:\, \mathtt{are}(x, \mathtt{a}) = \top\} = \{x \in Y \,:\, \mathtt{a}(x) = x\} = a(Y)$, by (18). ⊣

6.3. Concept truth. The second step towards uniting sentence semantics and concept semantics consists in the definition of a classification system based on the lexicon. Intuitively, the notions of truth must coincide in both versions. For example, we expect

$$\begin{aligned} (x, y) &\models \mathtt{love} \text{ if and only if } \mathtt{love}(x, y) = \top \\ x &\models \mathtt{smell} \text{ if and only if } \mathtt{smell}(x) = \top \\ x &\models \mathtt{steep} \text{ if and only if } \mathtt{are}(x, \mathtt{steep}) = \top\,. \end{aligned}$$

Assume that each key-word p_i corresponds to a predicate $x \mapsto \mathtt{p}_i(x)$ defined by a word in the lexicon.

On one hand, define a *lexical classification system* $(B, P, \models)$ by requiring for $p_i \in P$

$$x \models p_i \text{ if and only if } \mathtt{p}_i(x) = \top \tag{19}$$

and extend it to all concept vectors in $C(P)$ as described in Section 4.

On the other hand, view the map from key-words to lexical predicates as a map from basic concepts to predicates

$$\overrightarrow{p_i} \mapsto (x \mapsto \mathtt{p}_i(x)) \tag{20}$$

and extend it to arbitrary concepts using induction on Boolean complexity

$$\begin{aligned} \neg c \quad &\mapsto (x \mapsto \mathtt{not}(c(x))) \\ c \wedge c' &\mapsto (x \mapsto \mathtt{and}_S(c(x), c'(x)))\,. \end{aligned} \tag{21}$$

THEOREM 3. *Concept truth in the lexical classification system coincides with sentence truth, i.e. for every concept vector $c \in C$ and every non-empty subset Y of B the following equivalences hold*

$$Y \models c \Leftrightarrow c(Y) = \top \Leftrightarrow v_Y \to c = 1\,. \tag{22}$$

PROOF. It suffices to show the first equivalence, the second equivalence the follows from (13). First show that the following two equivalences hold for every basis vector $x \in B$ and every concept c

$$x \models c \Leftrightarrow c(x) = \top,\ x \not\models c \Leftrightarrow c(x) = \bot \tag{23}$$

using induction on the propositional complexity of concept vectors. The property holds for basic concepts. Indeed, (19) is equivalent to

$$x \not\models p_i \Leftrightarrow \mathtt{p}_i(x) = \bot\,,$$

because a predicate returns either $\top$ or $\bot$ for an individual.

The property now follows from (23) by the Fundamental Property. ⊣

Roughly speaking, $c(Y)$ is the *sentence vector* and $v_Y \to c$ the *concept vector* for the same string of words. Both Y and c may be compound expressions involving several words in the string. Typically, the determiners `all`, `some`, `no` or the relative pronoun `who` may occur in Y, the propositional connectives in c.

A final comment concerns the inner product of the concept space. It is tempting to replace the evaluation of the predicate (linear map) $c : U \to S$ at an argument Y by the inner product of the concepts c and v_Y of $C(P)$. This is not possible by (10) in Lemma 10.

6.4. Examples.

Example 1. *All banks are steep.*

Compute its meaning according to the procedure described in Section 5, replacing the label `big` by `all`. The resulting vector is $\texttt{are}(\texttt{all}(\texttt{bank}), \texttt{steep})$.

Lemma 14. *Let* $Steep = \{x : \texttt{are}(x, \texttt{steep}) = \top\}$ *and* $\texttt{bank} = \sum_{x \in Bank} x$. *The sentence* All banks are steep *has the three equivalent interpretations*

$$\begin{gathered}\forall x(x \in Bank \Rightarrow x \in Steep),\\ \texttt{are}(\texttt{all}(\texttt{bank}), \texttt{steep}) = \top,\\ v_{Bank} \to \texttt{steep} = \overrightarrow{1}.\end{gathered}$$

Proof. By the Fundamental Property (17)

$$\texttt{are}(\texttt{bank}, \texttt{steep}) = \top \text{ if and only if } \forall x(x \in Bank \Rightarrow x \in Steep),$$

The first equivalence follows, because `all` is the identity. The second equivalence is a special case of (22). ⊣

Example 2. *No banks are steep .*

The meaning vector $\texttt{not}(\texttt{are}(\texttt{bank}, \texttt{steep}))$ has been computed in Section 5

Lemma 15. *The sentence* No banks are steep *has the three equivalent interpretations*

$$\begin{gathered}\forall x(x \in Bank \Rightarrow x \notin Steep)\\ \texttt{not}(\texttt{are}(\texttt{bank}, \texttt{steep})) = \top\\ v_{Bank} \to \neg\texttt{steep} = \overrightarrow{1}.\end{gathered} \tag{24}$$

Proof. Note that the vector $\bot$ is the only vector v of S for which $\texttt{not}(v) = \top$. Hence $\texttt{not}(\texttt{are}(\texttt{bank}, \texttt{steep})) = \top$ is equivalent to $\texttt{are}(\texttt{bank}, \texttt{steep}) = \bot$. The latter is equivalent to $\forall x(x \in Bank \Rightarrow \texttt{are}(x, \texttt{steep}) = \bot)$ by the Fundamental Property, to $\forall x(x \in Bank \Rightarrow x \not\models \texttt{steep})$ by (23) and finally to $\forall x(x \in Bank \Rightarrow x \models \neg\texttt{steep})$, by definition of satisfaction of concepts in Section 4. ⊣

Example 3. *Big banks are rich and safe.*

Lemma 16. *Let* $Big = \{x \in B : \texttt{big}(x) = x\}$. *Then the sentence* Big banks are rich and safe *has the three equivalent interpretations*

$$\begin{gathered}\forall x(x \in Big \cap Bank \Rightarrow x \in Rich \;\; and \;\; x \in Safe)\\ \texttt{are}(\texttt{big}(\texttt{bank}), \texttt{rich} \wedge \texttt{safe}) = \top\\ v_{Big} \wedge v_{Bank} \to \texttt{rich} \wedge \texttt{safe} = \overrightarrow{1}.\end{gathered}$$

PROOF. The equality $\mathtt{big}(\mathtt{bank}) = \sum_{x \in Big \cap Bank} x$ is a particular case of (15). By (18), $\mathtt{are}(x, \mathtt{rich} \wedge \mathtt{safe}) = \mathtt{and}(\mathtt{are}(x, \mathtt{rich}), \mathtt{are}(x, \mathtt{safe}))$. ⊣

EXAMPLE 4. *Some banks are steep.*

The meaning vector is $\mathtt{are}(\mathtt{some}(\mathtt{bank}), \mathtt{steep})$.

LEMMA 17. *The sentence* Some banks are steep *has the two equivalent interpretations*

$$\begin{aligned} \mathtt{are}(\mathtt{some}(\mathtt{bank}), \mathtt{steep}) &= \top \\ v_{\mathtt{some}(Bank)} \to \mathtt{steep} &= \overrightarrow{1}\,. \end{aligned} \tag{25}$$

Moreover, $\mathtt{are}(\mathtt{some}(\mathtt{bank}), \mathtt{steep}) = \top \Rightarrow \exists_x (x \in Bank \text{ and } x \in Steep)$

PROOF. The equivalence (25) is a particular case of (22).

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

The results above explain why disambiguation via frequency counts of word occurrences in context windows works and how it could lead to the wrong conclusion.

The sentences in Sections 5 and 6 are translated to a concept of the form $v_Y \to c$. Here, Y is the set of (n-tuples of) individuals to which the sentence refers. The tacit assumption of information retrieval is that sentences are true. Under this assumption, a sentence defining the concept $v_Y \to c$ is true if and only if $v_Y \leq c$. The latter inequality implies that $v_Y \in C(P_i)$ whenever $c \in C(P_i)$, $c \neq \overrightarrow{0}$. This determines the meaning of the word in the sentence, because every individual in Y satisfies c . For example, if $\mathtt{rich}$ belongs to the same concept space $C(P_1)$ as financial institutions and $\mathtt{steep}$ to another concept space $C(P_2)$ constructed from geographical features. The examples 1, 3 and even 4 disambiguate the polysemous word *bank* place the involved concepts into $C(P_2)$. The negated sentence in Example 2, however, does not disambiguate, because the embeddings of $C(P_i)$ into $C(P_1) \oplus C(P_2)$ do not commute with concept negation, by Lemma 11. For example, if $\neg_\oplus$ is the negation of $C(P_1) \oplus C(P_2)$ and $\neg_2$ the concept negation in $C(P_2)$ then $\neg_\oplus \mathtt{steep} = \neg_2 \mathtt{steep} + \overrightarrow{1}_{C(P_1)}$.

§7. Conclusion.

New in the preceding approach is to place two distinct notions of truth, one for concepts and one for sentences inside one and the same category of vector-spaces. The formal description of both gives rise to a method transferring compositionality of functional logical semantics to vector semantics.

An obvious shortcoming is that the translation of grammatical strings into concepts depends not only on syntactical analysis, e.g. the type of words, but also on semantical properties of words. The examples go, however, beyond the compound noun-phrases in [17] or the sentences in [4]. Indeed, 'noise' like determiners, quantifiers, relative pronouns, negation, sentential connectives etc. are no longer ignored, because their logical content is essential for composing the concept representing a string.

Placing a word into a grammatical string is analogous to making a measurement on an observable. The outcome is indeterminate, in the sense that the meaning of a word may change with the surrounding string (measurement). Indetermination is captured by polysemous concept spaces in analogy to quantum logic protocols, [1]. An interesting question is whether the analogy can be pushed further, for example through representing similar words including opposites by vectors in a one-dimensional subspace of a complex Hilbert space. The intrinsic projectors of a tensor product of two-dimensional complex Hilbert space would still capture the logical aspects as described above.

REFERENCES

[1] Samson Abramsky and Bob Coecke, *A categorical semantics of quantum protocols*, ***Proceedings of the 19th annual ieee symposium on logic in computer science***, 2004, pp. 415–425.

[2] Johan van Benthem, *Guards, bounds and generalized semantics*, ***Journal of Language, Logic and Information***, vol. 14 (2005), pp. 263–279.

[3] Johan van Benthem and Kees Doets, Handbook of Philosophical Logic, ch. Higher-Order Logic, pp. 275–329, Reidel Publishing Company, Dordrecht, 1983, pp. 275–329.

[4] Stephen Clark, B. Coecke, E. Grefenstette, S. Pulman, and M. Sadrzadeh, *Concrete compositional sentence spaces*, ***Compositionality and distributional semantic models, esslli 2010***, 2010.

[5] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh, *A compositional distributional model of meaning*, ***Proceedings of conference on quantum interactions*** (W. Lawless P. Bruza and J. van Rijsbergen, editors), University of Oxford, College Publications, 2008.

[6] Paul Halmos, ***Lectures on boolean algebras***, Springer, 1974.

[7] Joachim Lambek, *The mathematics of sentence structure*, ***American Mathematical Monthly***, vol. 65 (1958), pp. 154–170.

[8] ———, *Type grammar revisited*, ***Logical aspects of computational linguistics*** (Heidelberg) (Alain et al. Lecomte, editor), LNAI, vol. 1582, Springer, 1999, pp. 1–27.

[9] ———, ***From word to sentence***, Polimetrica, Milano, Italia, 2008.

[10] Anne Preller, *Category theoretical semantics for pregroup grammars*, ***Logical aspects of computational linguistics*** (Philippe Blache and Edward Stabler, editors), Lecture Notes in Artificial Intelligence, vol. 3492, 2005, pp. 254–270.

[11] ———, *Linear processing with pregroup grammars*, ***Studia Logica***, vol. 87 (2007), no. 2:3, pp. 171–197.

[12] ———, *Toward discourse representation via pregroup grammars*, ***Journal of Language, Logic and Information***, vol. 16 (2007), pp. 173–194.

[13] Anne Preller and Joachim Lambek, *Free compact 2-categories*, ***Mathematical Structures for Computer Sciences***, vol. 17 (2007), no. 1, pp. 1–32.

[14] Anne Preller and Mehrnoosh Sadrzadeh, *Semantic vector models and functional models for pregroup grammars*, ***Journal of Language, Logic and Information***, (2011).

[15] C.J. van Rijsbergen, ***The geometry of information retrieval***, Cambridge University Press, 2004.

[16] Paul Smolensky, *Connectionism, constituency and language of thought*, ***Minds, brains, and computers*** (Robert Cummins and Denise Dellarosa Cummins, editors), 1988, pp. 284–308.

[17] Dominic Widdows, *Semantic vector products: Some initial investigations*, ***Second aaai symposium on quantum interaction, oxford***, AAAI, 2008, pp. 1–12.

LIRMM, 161 RUE ADA, 34194 MONTPELLIER, FRANCE
E-mail: preller@lirmm.fr
URL: http://www.lirm.fr/~preller