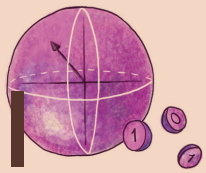
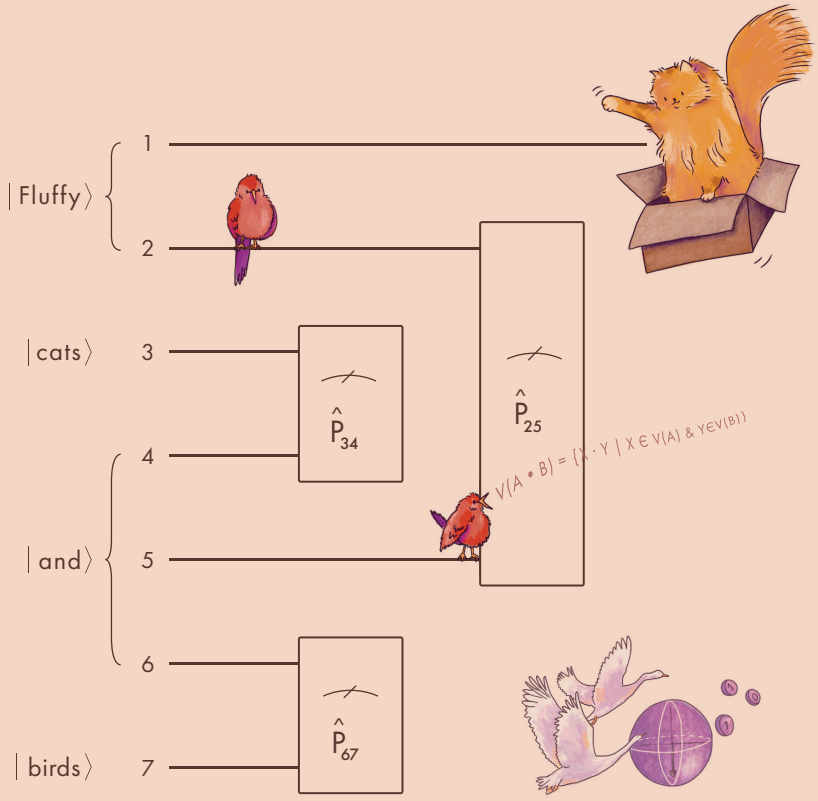
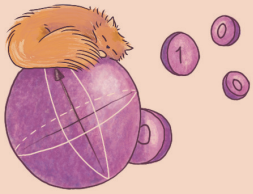
Quantum distributional semantics



QUANTUM ALGORITHMS APPLIED TO NATURAL LANGUAGE PROCESSING





A.D. Correia

Quantum distributional semantics

Quantum algorithms applied to natural language processing

A. D. Correia

PhD thesis, Utrecht University, September 2022
DOI: doi.org/10.33540/631
ISBN/EAN: 978-94-6423-964-5
Printed by: ProefschriftMaken // www.proefschriftmaken.nl
About the cover: The cat inside the box is an allusion to Schrodinger's cat, that is dead and alive at the same time (that is, in quantum superposition of both states). Similarly, the circuit in the cover represents one of the readings that is possible for the phrase, whose representation can also be in quantum superposition with that of the other reading. Illustration and cover design by Annemiek Schellenbach.

Quantum distributional semantics

Quantum algorithms applied to natural language processing

(with a summary in English)

Kwantum distributionele semantiek

Kwantumalgoritmen toegepast op natuurlijke taalverwerking

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. H.R.B.M. Kummeling, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op maandag 19 september 2022 des middags te 12.15 uur

door

Adriana Duarte Correia

geboren op 24 mei 1994 te Loulé, Portugal

Promotoren:
Prof. dr. ir. H. T. C. Stoof
Prof. dr. M. Moortgat

Beoordelingscommissie:
Prof. dr. C.J. van Deemter
Prof. dr. ir. H.A. Dijkstra
Prof. dr. I. Moerdijk
Prof. dr. M. Sadrzadeh
Prof. S.J.L. Smets

Dit werk werd mogelijk gemaakt met financiële steun van Peter Koeze en de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).

CONTENTS

LIST OF PUBLICATIONS

This thesis is based on the following publications:

- ◇ Correia, A. D., Moortgat, M., and Stoof, H. T. C., "Density matrices with metric for derivational ambiguity." *Journal of Applied Logics* **7**(5), 795–822 (2020).
- ◇ Correia, A. D., Stoof, H. T. C., and Moortgat, M., "Putting a spin on language: A quantum interpretation of unary connectives for linguistic applications." In *Proceedings of the 17th International Conference on Quantum Physics and Logic* (QPL '20). EPTCS **340**, 114–140 (2021).
- ◇ Apallius de Vos, I. M., van den Boogerd, G. L., Fennema, M. D., and Correia, A. D., "Comparing in context: Improving cosine similarity measures with a metric tensor." In *Proceedings of the 18th International Conference on Natural Language Processing* (ICON '21), 128-138 (2021).
- ◇ Correia, A. D., Moortgat, M., and Stoof, H. T. C., "Quantum computations for disambiguation and question answering. "*Quantum Information Processing* **21**, 126 (2022).

Other publications by the author:

- ◇ Correia, A. D., Leestmaker, L. L., Broere, J. J., Stoof, H. T. C., "Asymmetric games on networks: Towards an Ising-model representation." *Physica A: Statistical Mechanics and its Applications* **593**, 126972 (2022).
- ◇ Correia, A. D., and Stoof, H. T. C., "Nash equilibria in the response strategy of correlated games." *Scientific Reports* **9**(1), 1-8 (2019).

Other publications the author has contributed to:

- ◇ Huber, M. A., Correia, A. D., Ramgoolam, S., Sadrzadeh, M., "Permutation invariant matrix statistics and computational language tasks." ArXiv preprint 2202.06829.
- ◇ McPheat, L., Wijnholds, G., Mehrnoosh S., Correia, A. D., Toumi, A., "Anaphora and ellipsis in Lambek calculus with a relevant modality: Syntax and semantics." *Journal of Cognitive Science* **22**(2), 1-34 (2021).

PREFACE

"Este livro é como um livro qualquer. Mas eu ficaria contente se fosse lido apenas por pessoas de alma já formada. Aquelas que sabem que a aproximação, do que quer que seja, se faz gradualmente e penosamente - atravessando inclusive o oposto daquilo que se vai aproximar."

– Clarisse Lispector, "A Paixão Segundo G.H."

This thesis is the result of four years of research, done at the Institute of Theoretical Physics and at the Institute of Linguistics, under the supervision of Prof. Dr. Ir. Henk. T. C. Stoof and Prof. Dr. Michael Moortgat, in collaboration with the Center for Complex Systems Studies, at Utrecht University, due to the generous contributions of Peter Keuze to this center.

Inspired by [56], it results from a fascination with physics and linguistics, how their formalisms intertwine, and a desire to explore and bring these connections further.

The bulk of the work developed during this time resides in Chapters 2 to 5, that lead towards published research. To better understand it, a background in three major topics helps carry the weight. First and foremost, computational natural language processing, to understand the language automation resources developed in recent years, and what the current challenges are. Then, formal linguistics, to work from very precise notions of grammar, or *syntax*, and meaning, or *semantics*, that provides a framework to solve some of those challenges. And finally, quantum mechanics, necessary to understand how we will approach the problems, taking advantage of representing words as quantum states. These subjects are introduced in Chapter 1, in what is intended to be a framing background of some of the ideas, notations and concepts that will appear in later chapters. Given the scope of this work, I attempted in this chapter to accommodate the readers of diverse backgrounds, thinking of both experts and non-experts

in each topic. Although a lot of care was taken in this regard, it can still be rather technical at points, especially for a reader unfamiliar with a specific topic. In this case, a number of further resources is given throughout, as well as examples, and I sincerely hope that the attentive reader will still be rewarded with the understanding of the problems and solutions in the following chapters.

With respect to the research itself, the starting motivation was to make use of density matrices as word representations while relying on a more refined notion of grammar than what had been previously used. The overarching problem that we tackle is that of representing several readings of ambiguous phrases simultaneously as quantum superpositons, concluding that it is a problem that is particularly well suited to quantum computation. On this journey, a number of other discoveries are made: in Chapter 2 we learn how Einstein's notation of general relativity can be used to distinguish between ambiguous readings; in Chapter 3 how derivational ambiguities (arising in languages like Dutch) can be treated, quite literaly, with an extra spin; in Chapter 4, how the notion of a metric can help us better compare words that appear in similar contexts; and in Chapter5, how a quantum search algorithm can be deployed to search for the answer to a question, faster than what a classical computer could.

Utrecht,
19 September, 2022

1

INTRODUCTION

1.1 NATURAL LANGUAGE PROCESSING

One of the central obstacles of human-computer interaction is the fact that computers understand very little about the meanings of words. A telling example is the necessity to come up with other symbolic representations to establish this communication: programming languages were designed to this effect. From binary assembly languages to a more modern, "English-like", programming language, such as Python, some translation from a human request to a command line always had to take place before an action was executed by the computer. After these methods were established, a new question was ripe: can we extract the meaning and intent of natural language utterances directly in an automated way, from sentences to statements that our computers can understand? What follows sketches the efforts made to this date with regards to inferring the meanings of words by using large volumes of text data to find the frequency with which expressions appear in that data.

1.1.1 *Vector Space Models*

In the field of Natural Language Processing (NLP), Vector Space Models (VSMs) have turned out to be one its most successful tools. The idea behind a VSM is to represent each word or document as a point in a certain space. In this sense, a document can be regarded as anything from an entire

page of text, to a paragraph, a sentence, or even a "tweet". This model first became relevant for an information retrieval system called SMART [121, 135]. Points that are close in that space should represent similar content, contrary to points that are far apart. In information retrieval, a query such as "fluffy dog" is turned into a point in that space, and the closest point to it is given as the answer to the user. One such answer could be "poodle". The success of the application of the model in this task inspired the application of VSMs to other tasks with encouraging positive outcomes, that achieved supra-human results on the TOEFL dataset [112] and human-like performance on the American entrance college SAT tests [134].

Essential to this success were two important underlying ideas, that explain the name of the approach: that the points that represent the documents can be seen as vectors that belong to a larger vector space, and that the notions of distance between these vectors encode meaningful semantic relationships, namely of similarity [122, 135].

While the use of vectors was common in artificial intelligence and cognitive science previously [121], the bottleneck resided in how to build them. To understand how this could be done, suppose that we characterize elements in a dataset, that we want to represent as vectors, by the values of some features, or attributes, that measure different aspects of that element, like their color, shape or size [146]. The elements represented by such vectors belong to a vector space that has these features as a basis, where for example "poodle" might be described by the vector $(3.4, 4.5, 1.2)$, with the entries representing the color, shape and size of the object, in some numerical scale. Another way in which words can be represented as vectors is if the features correspond to the characters that compose that word. For a basis $\{n_i\}$, with i a letter in the alphabet, and n_i the number of times the letter i appears in it. For the word "poodle", as an example, the vector in this basis is $(\dots, 1, 1, \dots, 1, \dots, 2, 1, \dots)$, where 1 corresponds to the number of times the letters "d", "e", "l" and "p" appear, respectively, and 2 that number but for the letter "o". The ellipsis representing a value of 0 for all the other alphabetic entries of the vector.

In a similar way, in cognitive science, prototype theory also makes use of vectors, assuming some members of a category are more prototypical than others [67]. For example, "labrador" is a more central (prototypical) member of the category "dog", while "borzoi" is less so, being more peripheral. Thus concepts can have various degrees of memberships in categories, described by *graded categorization*. A natural way to implement this idea early on was to have concepts as vectors with some concept features as basis, and categories as sets of vectors [130]. However, these were usually based on numerical scores that were elicited by questioning human subjects, and so were not easy to automated.

Hopefully, this shows that finding the right features and the respective appropriate weight is no trivial task [92]. When moving to more automated ways of extracting representations of language fragments, using for instance machine learning techniques, inputs are usually still represented using feature vectors, and, although they yield more accurate results in downstream tasks, their interpretation also more elusive.

1.1.2 *Term-document and word-context matrices*

To automatically obtain vectors that represent words and phrases, in a way that preserves some semantic information but at the same time optimizes tasks such as information retrieval, the *statistical hypothesis* is used. It claims that statistical patterns of human word usage can be leveraged to infer what people really mean when they search, for example, for "fluffy dog". To test this hypothesis, Furnas *et al.* [46] asked a number of annotators to describe certain target words using other words, such as "green" for "lime", and studied the patterns that emerged, finding significant agreement in the words chosen across annotators. This hypothesis subsequently gave way to the adoption of two other important ones: the bag-of-words hypothesis, and the distributional hypothesis.

If we have a large number of documents that can each be represented as a vector, it can be useful to organize them as a matrix. While the columns correspond to documents, the row vectors of the matrix correspond to the

terms, that is, features that describe the document and that are usually taken to be the words that appear in the document. This matrix is called the *term-document matrix* [135]. If a certain word appears more than once in a document, its vector entry corresponds to that frequency, similarly to what was done when representing a word by its characters. This introduces a frequency component to the vectors that can be automated. One property of this representation is that the ordering in which the words appear in the document does not matter, and as such it is called the *bag-of-words* representation. Note that this representation is not very useful when different meanings are obtained by changing the ordering of the words, such as in "Mary likes John" versus "John likes Mary". However, for information retrieval, the bag-of-words representation proved to be good enough, allowing the frequency of the words in a document to inform its relevance to a particular query. This is at the basis of the success of SMART [123], where word-frequency tables were capable of capturing information about the document's content.

Soon after, Derweester *et al.* [37] noted that if, instead of comparing column vectors, they compared the row vectors of the term-document matrix, they would get a vector representation of the word or term instead, containing information about the distribution of that term across documents. Therefore, comparing these vectors allowed them to compare different words. However, for this application, the entire document might not be of the optimal length. Therefore, in general, we can create a *word-context matrix*, where the context can be a word, phrase, sentence, paragraph, chapter or, ultimately, a document [135]. To make this choice, the *distributional principle* is often invoked. According to Harris [54], words that appear in similar contexts tend to have similar meanings. While his original assertion did not, in principle, require any frequency analysis, it has been used as such in this context. This principle is used to justify the use of VSMs in the task of word similarity, in the following way: if the context of a word is defined by its surrounding words, then similar frequencies of these words would indicate that the words are similar [78]. For example, if we extract the vectors of the target words "happy" and "glad", they should be fairly similar, as these words often appear close to the same context words, and although this is

not a methodology without fault ("happy" and "sad" might also appear in the same contexts), it often provides satisfactory results, namely when distinguishing words that appear in very different contexts, as exemplified in Fig. 1.1.

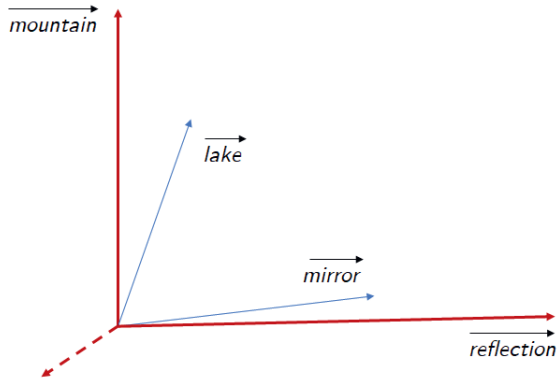


Figure 1.1: Vector representation of two words.

Thus, similar row vectors in the word-context matrix produced similar word meanings, which therefore automated the problem of synonymy in information retrieval [37]. Given that to some extent these vectors contained semantic information, the term "distributional semantics" came to be often interchangeable with VSMs.

There are some interesting historical precursors leading up to Deerwester's insight. In first instance, already the philosopher Wittgenstein had argued that a word can be understood by its context, but he was mostly referring to the physical, real-world context, while in this case we employ other words as contexts [147]. Recently, some debate has been generated regarding whether this type of context will ever be enough for computers to capture the full meanings of words [14], and Wittgenstein's intuition might be better captured in what is nowadays known as multimodal learning, where images of the referents are used to aid text interpretation [63]. Weaver [142] had already suggested in 1955 that the word context for disambiguation

should matter in the context of translation[a]. It would be Firth, however, who would capture this idea in what is now the slogan of the distributional principle: "You shall know a word by the company it keeps." [143]. The main insight is that words don't appear randomly in text; instead, they form patters, and if we find clever enough ways of analysing them, we might just be able to infer their meaning.

1.1.3 *Statistical Natural Language Processing*

The idea that information about the context of written text can be extracted from the distributions of words across large amounts of text gave rise to yet another type of language representation. Statistical language modelling, or more simply just language modelling (LM) involves estimating a probability distribution that captures statistical regularities in natural language [76]. Its roots date back to the beginning of the 20th century, when Markov first used what is now known as Hidden Markov Models to try to model letter sequences in works of Russian literature [80]. It would be Shannon, however, who would in 1951 introduce the first systematic prediction of content in an English text, introducing and using *n-grams* [127]. For the task of predicting the next word, a stochastic probability function P can be defined such that the probability of the next word depends on the preceding $n-1$ words:

$$P(w_n|w_1,...,w_{n-1}). \tag{1.1}$$

a "First, let us think of a way in which the problem of multiple meaning can, in principle at least, be solved. If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. "Fast" may mean "rapid"; or it may mean "motionless"; and there is no way of telling which. But if one lengthens the slit in the opaque mask, until one can see not only the central word in question, but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. The formal truth of this statement becomes clear when one mentions that the middle word of a whole article or a whole book is unambiguous if one has read the whole article or book, providing of course that the article or book is sufficiently well written to communicate at all. The practical question is, what minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?" [142]

Common values of n are 2, 3 or 4 [79]. By looking at the frequency in which these arrangements of words appear anywhere else in the text. This makes it possible to calculate, for example, in "Mary has a fluffy dog", what is the probability that the word "dog" follows the word "fluffy". For n=2, this depends on the distribution $P(\text{dog}|\text{fluffy})$ with which these two words appear next to each other across the data, as well as the normalized frequency $P(\text{fluffy})$ of appearance of "fluffy", resulting in $P(\text{fluffy dog}) = P(\text{dog}|\text{fluffy})P(\text{fluffy})$. Given the sparsity of language data, $n = 3$ is usually used, which is a value also common for the context windows of word-context matrices in VSMs. In fact, windows of around 2 seem to achieve the best results in this type of extraction, suggesting that the meaning of a word can be extracted from its immediate surroundings [112].

1.1.4 *Machine Learning and Natural Language Processing*

While the distributional and the modelling approaches to automatic extraction of meaning lived somewhat independently, in 2013 Mikolov *et al.* [87] introduced a paper that radically changed the way in which these two methods interacted. I introduce this advancement in the field because it is important to understand for two reasons. The first is that it happened in parallel with the development of ways to compose, in a grammatical way, vectors that were originally obtained using word-context matrices, introduced for instance in Mitchell *et al.* [91], a field of study that this thesis advances, but it remains to be shown how precisely these ideas extend to more dynamic ways of obtaining vector representations, some of which already seem to have some notion of syntax embedded [59]. The second is that, on trying to go beyond this restriction, it motivates the explorations in Chapter 4.

The first machine learning algorithms for Natural Language Processing were trained on a language modelling goal, that is, as a speedup technique to implement statistical language models, where the probabilities output by the model corresponded to the Markov probabilities for a given n. How each word is represented as an input of these learning models varied, but usually started as 1-of-V, where V is the size of the vocabulary, or were otherwise

initiated at random. Instead of remaining fixed, these vectors would be updated in order to optimize those probabilities, using more robust machine learning architectures, like recursive neural networks [88]. It was then shown that if the goal was turned into finding the vector representations that predicted the neighbouring words (CBOW), or that instead predicted a certain word based on the neighbouring words (Skip-gram), then the resulting vectors contained semantic information. In particular, an additive measure of the vectors after being projected onto the principal components provided relationships between vectors such as country/capital [89]. This approach became known as "word2vec" and was ground-breaking because it gave a new, more efficient, way of finding vector representations of words that were not just based on counts, were efficient to train with large amounts of data, and seemed to contain relevant semantic information. What they provided, most of all, was a great set of vectors to initialize other models for different tasks, and no less for the distributional compositional programme, that we will introduce in the next section. From then on, other language models that provide more informative vector representations have been developed, from BERT [**?**] to more recently GPT-2 [111], as more and more tasks rely on their improvement for better results.

1.2 FORMAL GRAMMAR

There is one crucial ingredient that the VSMs lack on their own: how the notions of meaning are related with the notions of grammar. However, we need this to go beyond the "bag-of-words" models, in an explainable and controlled way. The proposal is to do this by composing the representations, or interpretations, of the individual words in accordance with grammatical rules, in order to obtain representations of larger fragments of text for which it is difficult to get frequency statistics. Take, for instance, the expression "fluffy dog". As it is harder to find a vector of context frequencies for this entire expressions than it is for "fluffy" and "dog", from the sheer fact that it appears less often in any text, we might be better off finding a vector representation for each word and an accurate way to represent the interaction between them. To do that, some notions of formal linguistics

are required. What follows starts by contextualizing the discussion around compositionality, paving the way to the introduction of rigorous notions of grammar and meaning, by presenting formal languages for the syntax and the semantics. These concepts form the substrate of the present thesis. For a deeper understanding of the formalisms that we will be using, the reader is referred to Refs. [1, 60, 98, 99].

1.2.1 *Compositionality*

"Every time I fire a linguist, the performance of the speech recognizer goes up." The apocryphal words of one of the principal proponents of the statistical approach to language modelling, F. Jelinek, are often cited as support to the idea that frequency-based semantics can completely dispense with any linguistic work done previously. However, he apparently hadn't completely lost faith that this work would soon become relevant: "We must 'put language back into language modelling"'[118].

To properly put in place a solution to Jelinek's challenge, the call is to include grammar and theoretical notions of meaning into our treatment of language. This turns out to be key in dealing with linguistic cases that cannot be captured statistically in a simple way, as we will see in later chapters with sentence-level ambiguities.

To frame this inclusion, a foundational discussion from the earlier philosophical and linguistics communities is worth revisiting. It contemplates whether *contextuality* or *compositionality* were the most important approaches for language processing and understanding [61]. According to the contextuality proponents, a meaning can only be assigned to a complete sentence, and so its parts can only be understood in the context of that sentence; we can think of this as the *sentence-to-word* meaning direction. On the contrary, compositionality implies that words have meanings of their own, and it is their composition that forms the meanings of larger fragments of text, such as a sentence; the meaning is assigned in a *word-to-sentence* direction. A key distinction is that in the contextuality framework a fragment like "likes

Mary" has no meaning, while one can be assigned in the compositionality framework, given a proper definition.

Informally, the *principle of compositionality*, thus, states that "the meaning of a compound expression is a function of the meanings of its parts and of the way they are syntactically combined" [104]. That is to say, "the meaning of the whole is computed in some predictable way from the meaning of its parts" [60]. Also referred to as Frege's principle in the context of modern computational linguistics, it was only really stated as such by his student Carnap [24, 105].

Until the 1960's, the contextuality view was dominant in linguistics [61]. In logics, and in computer science as a derivative, the principle of compositionality was more widely accepted. The introduction of this way of working in logics in the 1930's allowed the meanings of a certain expression to be defined unambiguously, with the meaning of a complex expression being given by the meanings of its individual parts and a set of compositional rules. For example, if in Python we write `print("dog")`, the output will invariably be `dog`, since the command `print` and the string `"dog"` are well defined, as well as their interaction, or composition. The idea that human languages can be understood too via the composition of its building blocks thus gained strength alongside the development of computer languages, resorting likewise to formal logical statements.

The philosopher and mathematical logician Richard Montague was dissatisfied with the way semantics was treated by Chomsky's proposal in 1965 [25], which introduced a "deep structure" of a complete sentence that would have to be given before any semantics could be computed, making it a non-compositional process. Montague can be credited for making the informal notion of compositionality precise [93], being of the opinion that a single mathematical theory could be used to study the syntax and semantics of both natural languages and formal languages as in logic or computer science. In his view, syntax and semantics are understood as similar algebraic structures (sets of symbols and rules to manipulate them), and so a compositional interpretation can be understood as a structure-preserving mapping, a *homomorphism*, from syntax to semantics.

This makes it an extremely useful method to study the relationships between form and meaning in natural language, in particular when we are trying to model natural language in a way that a computer can understand, since this way of thinking about the meanings of expressions and their manipulation is also the basis of programming languages. Therefore, if we can abstract language down to some rules, independently of the particular inputs, we can map them to (computer) programs. For a sentence such as "John likes Mary", we can first ignore the meanings of these particular words, and formally describe what it has in common with any other sentence that is likewise formed by a subject, an object, and a transitive verb, and afterwards go back to the specific words to process its meaning as a computer program.

Montangue's theory is quite general, as it doesn't commit to a particular choice for the syntactic or semantic algebras, so long as they are related in a structure preserving way. In the sections that follow, we provide some background for the choices that are made in this thesis, first introduced by van Benthem [137] and Moortgat [95]. We first discuss the typelogical grammar that will serve as the syntactic front end for the compositional mapping, and then we turn to the λ-calculus that we use to talk about meanings, and show how these two calculi can be related in a compositional way.

1.2.2 *A calculus for syntax: typelogical grammar*

The typelogical grammar that we will use in this thesis is an extended version of the Syntactic Calculus that was introduced by the mathematician Jim Lambek in 1958 [68]. Lambek's syntactic calculus belongs to the family of categorial grammar formalisms. Categories (or types, as we will use both terms interchangeably), as the name suggests, play a central role in this framework.

Before we present the Lambek calculus that we will will ultimately use as the syntax of natural language, we build up to it by explaining some of the developments that precede it. The first attempt at an algebra of

syntax was given by Ajdukiewicz [5]. In this formalism, an expression is assigned a syntactic *type*. Informally, in grammar we use types to group together expressions that behave in a similar way, in the following sense. In a sentence like "John likes Mary", "Mary" can be substituted by many other expressions, such as "the dog that Bob owns", "a house with a garden", "a fluffy dog", etc. We then say that "Mary" and all these other expressions belong in the same type, and we call it the type, or category, of noun phrases.

Categorial grammars refine this view on types as sets of expressions with a similar behaviour by introducing a distinction between complete and incomplete expressions. For complete expressions, we don't need extra information to come to a grammaticality judgement. These expressions are assigned *basic* types. For incomplete expressions, we need some extra context before we can decide on whether they are syntactically well-formed. We can see this by comparing "the dog that Bob owns" with "the dog that Bob", which still needs a completion. Incomplete expressions like the latter get a *complex* type, that encodes how they will combine syntactically.

To keep things simple, let us assume basic syntactic types np, n and s, corresponding to noun phrases, simple common nouns, and well-formed sentences, respectively. For complex types, a type-constructor, $-$, is introduced such that, together with any type A and any type B, it generates the fractional type $\frac{A}{B}$. A lexicon then assigns a type to each word. Next, the types combine via a cancellation scheme, represented in the following notation:

$$\frac{A}{B}, B \to A. \tag{1.2}$$

It reads as "An expression of type $\frac{A}{B}$ combines with an expression of type B into a formula of type A". The intuition is that these types interact in the same way as arithmetic multiplication and fraction, $\frac{a}{b} \times b = a$.

Introducing the notion of *directionality*, an expression of a complex type can combine with an expression of simpler type expression either from its left or from its right. Bar-Hillel [10] accommodated this in the previous

syntactic account by replacing the type-constructor with two new ones, / and \, representing directional fractions. The type A/B reads "A over B" the type $A\backslash B$ is reads "A under B". There are now two corresponding cancellation rules,

$$B, B\backslash A \to A, \qquad A/B, B \to A, \tag{1.3}$$

indicating concatenation and cancellation to the immediate left, or to the immediate right. These statements are read respectively as "An expression of type B to the left of an expression of type $B\backslash A$ combines into an expression of type A", and "An expression of type B to the right of an expression of type A/B combines into an expression of type A".

Taking into consideration the directionality of language in this way, we can correctly assert the grammaticality of a large portion of grammatical phrases. The sequence of types in "flies Mary", for instance, does not combine to the type s using these rules, as can be immediately verified. A more complex example is that of a sentence formed with a transitive verb. Assigning the types np to "Mary", np to "John" and $(np\backslash s)/np$ to 'likes', we can indeed conclude that "John likes Mary" is a well-formed sentence by using the rules in 1.3, by showing that the types of the constituent words reduce to the type s, schematically as

$$\dfrac{\dfrac{\text{John}}{np} \quad \dfrac{\dfrac{\text{likes}}{np\backslash s/np} \quad \dfrac{\text{Mary}}{np}}{np\backslash s}}{s} \tag{1.4}$$

and that we cannot use these rules to show the grammaticality for any other word reordering.

Then, building on this earlier work, Lambek [68] formulated his Syntactic Calculus, showing that, beyond just the algebraic symbols and mappings between them, a full-fledged deductive system for reasoning about types and their relations could be constructed. At the type level, the syntactic calculus adds a third connective to Bar-Hillel's formalism, an explicit product operation $\bullet$ that is a multiplicative type-constructor, and stands for expres-

sion concatenation. The connectives / and \ are its left and right residuals. The full set of syntactic types, Lambek types, is thus given recursively in the following notation:

$$F ::= P|F \bullet F|F/F|F\backslash F, \tag{1.5}$$

where P is the set of basic type formulas, F is the set of all type formulas, and | stands for choice, that is, a type is either basic *or* of one of the connective-generated complex types. Since this process works recursively, there is an unlimited possibility for different types.

The intended model for these types makes the informal view of types as "sets of expressions with a shared behaviour" explicit. In it, types are interpreted as sets of expressions, where we write $v(A)$ for the set of expressions of type A. Whatever the interpretation may be for basic types, for complex types, we want it to reflect our understanding of the type-forming operations:

$$v(A \bullet B) = \{x \cdot y \,|\, x \in v(A) \;\&\; y \in v(B)\}, \tag{1.6}$$
$$v(C\backslash B) = \{x \,|\, \forall y \in v(B),\, x \cdot y \in v(C)\}, \tag{1.7}$$
$$v(A/C) = \{y \,|\, \forall x \in v(A),\, x \cdot y \in v(C)\}. \tag{1.8}$$

The operation $\cdot$ is interpreted as concatenation of expressions. For fractional types, this formulation describes how their interpretation is that expressions that belong to those types result in an expression of the "numerator" type after it is concatenated with expressions of the "denominator" type.

With this language model in mind, we can now turn to the rules of the syntactic calculus. The calculus is designed to produce statements $A \to B$ expressing valid conclusions $v(A) \subseteq v(B)$.

As for the rules, we first need $A \to A$, since every set is a subset of itself, and then *cut* rule

$$\text{if } A \to B \text{ and } B \to C, \text{ then } A \to C, \tag{1.9}$$

since set inclusion is transitive. Since concatenation is associative, we also need to add such a rule:

$$A \bullet (B \bullet C) \leftrightarrow (A \bullet B) \bullet C. \tag{1.10}$$

Finally, we have that

$$A \to C/B, \tag{1.11}$$

if and only

$$A \bullet B \to C, \tag{1.12}$$

and if and only if

$$B \to A\backslash C. \tag{1.13}$$

With these rules, from the axioms

$$B\backslash A \to B\backslash A, \qquad A/B \to A/B, \tag{1.14}$$

we can now not only derive the following cancellation schemata

$$B \bullet B\backslash A \to A, \qquad A/B \bullet B \to A, \tag{1.15}$$

that is akin to Eq. (1.3) and works in the direction of less complex types, but also rules that increase the complexity of types, such as

$$B \to A/(B\backslash A), \quad B \to (A/B)\backslash A, \tag{1.16}$$

an operation known as *type-lifting*, from which we can obtain, for example, the type of subject pronouns (words like "he" or "she") from the type of noun phrases, as $np \to s/(np\backslash s)$, as well as the type of object pronouns (him or her) from the type of nouns, as $np \to (s/np)\backslash s$. The former indicates that every noun phrase in subject position can be replaced by a subject pronoun, such as in the sentence "He likes Mary". In contrast, a noun phrase in

object position is replaced by a distinct type of pronoun, as in "John likes her". This exemplifies the different behaviours of these two families of pronouns, which in turn justifies the assignment of distinct syntactic types. The correctness of this type assignment can be seen in that, while both pronoun types can be formed from the type of noun phrases, we cannot generate, using these rules, one pronoun type from the other.

In the presence of transitivity, given in Eq. 1.9, the questions of how to find out whether $A \to B$ holds is not immediately obvious. To solve this problem, Lambek reformulated the syntactic calculus as a logic sequent calculus. The reformulation has the nice property that if one can prove a theorem, then it is always possible to prove it without resorting to the cut rule. The sequent calculus, in other words, has a decision procedure, an algorithm that allows one to decide in a finite number of steps whether a derivability statement holds or not.

To see this, suppose that we want to know whether "dog fluffy" is grammatical. That is, we want to decide whether n/n concatenated with n from its right combines to n. For this, we need to check if our rules allow us to combine the types into the following statement, which we intuitively infer is wrong:

$$n \bullet n/n \to n. \tag{1.17}$$

Starting from Eqs. 1.15, we know already that we cannot use either of them directly. We also know that we cannot use that Eqs. 1.12 and 1.13 are equivalent to Eq. 1.11 because the target type is not fractional. But we could still try to use the cut rule, using some other type A,

$$n \to A \quad \text{and} \quad A \to n. \tag{1.18}$$

If we find such an A, we are one step closer to knowing where the combination of expressions is grammatical or not. However, the number of possibilities for A, together with the number of times that we can keep applying the cut rule after we find A, is infinite, and it might not be possible to decide whether the initial assertion was correct or not. Since we have shown that only the cut rule gives a possibility to show that n combines

with n/n to give n, there is no *cut-free* way of showing that this combination is possible. Therefore, by the sequent calculus we can conclude that this combination is ungrammatical, in accordance with our intuition.

Because of this, we can now *prove* whether an expression is grammatical. For instance, to say that a sentence like "John likes Mary" is a well-formed sentence, is to say that we can prove that the ordered list of types np, $(np\backslash s)/np$, np derives type s. The idea of this proof is to go from axioms to conclusions. The following statement,

$$np, (np\backslash s)/np, np \vdash s, \tag{1.19}$$

that we call a *sequent* and where the turnstile stands for "derives", should be the conclusion statement of that proof. The initial statements are the axioms, that introduce the types of the individual words, and in this calculus we prove whether a conclusion follows from the axioms.

This sequent calculus is known as the (non-)associative Lambek calculus **L**, of which the non-multiplicative fragment is used in this thesis, and that we omit here because it is formally introduced in Chapter 2.

1.2.3 *A calculus for semantics: λ-calculus*

Let us now turn to the semantic calculus that forms the target for the compositional mapping, which has a structure entirely parallel to the syntactic calculus. The language consists of the terms of the (typed) λ-calculus and the semantic types provide the typing rules that pick out the well-formed expressions of this language. We then discuss the interpretation process that assigns a semantic value in some intended model to each well-formed expression. For the latter, we consider two options: a Montague-style model-theoretic , truth-conditional semantics, and the distributional, vector-based semantics that is the subject of this thesis. The development of a formal semantics using the typed λ-calculus is well established [23, 40, 60], and here we introduce some of the technicalities that will aid the reader of Chapters 2 and 3.

Parallel to the Lambek types that we had previously, we now introduce semantic types τ. The semantic type system has the same recursive structure that we saw for the syntactic types, but its details change. While for syntax we distinguish the basic types np, n and s, because they behave differently syntactically, for the semantic types we will similarly have to decide what the essential distinctions are at the basic type level. This is inextricably related to the interpretation of the type system, to which we come back below. For incomplete expressions, instead of $\backslash$ and $/$, we now have a single type-forming operation $\rightarrow$ to form complex types:

$$\tau := \delta | \tau \rightarrow \tau, \tag{1.20}$$

with δ the set of basic types. The reason why now we only have one connective is that word order is traditionally not relevant for the semantic type system which considers language from a meaning perspective. We work on bringing directionality to the level of the interpretation in Chapter 2.

We now turn to the language for which the semantic types provide the typing rules, which is the language of the simply typed λ-calculus, introduced as a model of computation by Alonzo Church [26]. Using a variable-binding operator, the λ[b], this calculus forms the basis of typed functional programming languages, and in that sense allows us to define executable commands using the λ abstraction.

The typed statements of λ-calculus give us a language to talk about the semantics. The well-formed expressions in this language are the *terms* Φ_τ. For each type τ, we have a set of constants C_τ and a set of variables V_τ. Using $c, c', \ldots$ for constants, $x, y, z, \ldots$ for terms in general, and $t, u, \ldots$ for terms in general, we establish three term formation rules. The first is an atomic rule, stating that all constants and variables of type τ are well-formed expressions of this type, that is, are in Φ_τ. Then we introduce the *application rule*, stating that if $t \in \Phi_{\tau \rightarrow \tau'}$ and $u \in \Phi_\tau$, then $(t\, u) \in \Phi_{\tau'}$.

b Other variable-binding operators include the $\exists$ operator and the $\forall$ operator, that can be introduced in this calculus to retrieve first-order logic statements.

Finally, there is the *abstraction* rule, stating that if $x \in V_\tau$ and $t \in \Phi_{\tau'}$, then $\lambda x.\, t \in \Phi_{\tau \to \tau'}$.

Intuitively, expressions with an arrow type $\tau \to \tau'$ can be understood as functions. In such a view, the application rule tells us how we can use a function by applying it to an argument of the appropriate type, and the abstraction rule tells us how we can create a function. Then, the interpretation of the term language makes this intuition precise.

The first interpretation that we will put forward is the model-theoretic one, resulting in a Montague-style semantics, where a compositional interpretation establishes a relation between language and "the world". The intended model for this kind of interpretation consists of the set U, the "universe of discourse", encompassing all entities that language can talk about and refer to, and the boolean set $\{\texttt{true}, \texttt{false}\}$. For each type τ, we define a denotation domain D_τ, which is the set that contains the possible meanings of expressions of type τ. In a simple account, this interpretation uses the basic types e and t, such that $D_e = U$ and $D_t = \{\texttt{true}, \texttt{false}\}$. The domains of complex types are generated recursively from the previous as

$$D_{\tau' \to \tau} = D_\tau^{D_{\tau'}}, \tag{1.21}$$

where $D_\tau^{D_{\tau'}}$ stands for all functions from $D_{\tau'}$ to D_τ.

Each term $t \in \Phi_\tau$ is now assigned a semantic value which is a member of the denotation domain of its semantic type given as $[\![t]\!]_I^g \in D_\tau$. Here g is the assignment function that assigns an arbitrary element of D_τ to variables of type τ, such that $[\![x]\!]_I^g = g(x)$, and I is the interpretation function that points each constant term to a specific value in the denotation domain, such that $[\![c]\!]_I^g = I(c)$, which is constant across assignments. For application, we have, for $t \in \Phi_{\tau \to \tau'}$ and $u \in \Phi_\tau$, that

$$[\![(t\, u)]\!]_I^g = [\![t]\!]_I^g \left([\![u]\!]_I^g\right) \in \Phi_{\tau'}. \tag{1.22}$$

For abstraction, for $x \in V_\tau$ and $t \in \Phi_{\tau'}$ we have that $[\![\lambda x.t]\!]_I^g$ is a function from D_τ to $D_{\tau'}$, such that for each object $a \in D_\tau$, it computes the semantic value of $[\![t]\!]$ for an assignment g' that assigns a to x and for the rest is like g.

An alternative interpretation that we can give to the semantic language is a distributional, vector-based one, where the interpretation domains are taken as vector spaces. These vector spaces are intended to be obtained from VSMs. The practical way in which compositionality is implemented when words are represented as vectors is done by assuming the same two basic semantic types, e and t, but by assigning them as domains the vector spaces N and S, respectively, so that $D_e = N$ and $D_t = S$. The vector space N is where nouns are represented, and S is the vector space where the meaning of sentences is represented (Fig. 1.2). As such, maps between elements of these vector spaces are given as linear maps, with these themselves forming a vector space [7], such that the domain of complex semantic types is given as

$$D_{\tau' \to \tau} = D^*_{\tau'} \otimes D_\tau. \tag{1.23}$$

For all other things, this model behaves in an equivalent way to the model-theoretic interpretation.

1.2.4 *From Syntax to Semantics*

We will now show how the languages that we have developed so far for syntax and semantics are connected, and how that can be used to our advantage for natural language processing.

To represent the syntax-semantics mapping, we define the homomorphism $\lceil . \rceil$, that sends syntactic types to semantic types. For the model-theoretic interpretation, $\lceil n \rceil = e \to t$, $\lceil np \rceil = e$ and $\lceil s \rceil = t$. Simple common nouns, such as "dog", with syntactic type n, are interpreted with a function of type $e \to t$ because this is the type of characteristic functions in the set D_e. For instance, in the case of "dog", it specifies which elements of U are dogs, by mapping those elements to the the boolean value `true`. In the distributional interpretation, $\lceil n \rceil = \lceil np \rceil = e$ and $\lceil s \rceil = t$. We see here that the distinction between common nouns and noun phrases is flattened, as we can obtain a distributional vector for a word like "dog".

Complex syntactic types are sent to semantic types in a similar way for both interpretations, as

$$\lceil A\backslash B\rceil = \lceil B/A\rceil = \lceil A\rceil \to \lceil B\rceil \tag{1.24}$$

Next, let us see how $\lceil . \rceil$ acts on derivations, sending a syntactic derivation to its semantics counterpart. Going back to the example of "John likes Mary", we have two constants of type e, John' and Mary', whose interpretation is some element in U, and a constant of type $e \to e \to t$, likes', that takes two individuals and provides a relationship between them, which is either true or false. In this case, the resulting interpretation of the sentence is whether the relationship between John and Mary estblished by the verb holds, namely whether it is true that John likes Mary. The composition of the semantic types is parallel to that of that of the syntactic types, as given in Eq. (1.4):

$$\dfrac{\dfrac{\text{John}'}{e} \quad \dfrac{\dfrac{\text{likes}'}{e \to e \to t} \quad \dfrac{\text{Mary}'}{e}}{e \to t}}{t} \quad . \tag{1.25}$$

The resulting (constant) term in this case is (likes'(Mary)')John'), interpreted as boolean value. In terms of notation, to follow this parallel more broadly we decorate sequents such as those in Eq. (2.1) with constant or variable terms,

$$x : n, z : (n\backslash s)/n, y : n \vdash ((zy)x) : s, \tag{1.26}$$

a notation that we use throughout this thesis.

This guarantees that each expression has a unique derivation and meaning structure assigned. But the insight is greater than this: by understanding this way of reasoning, we can reach a level of abstraction that is akin to those of programming languages, and draw the parallel with natural languages, by thinking of words of complex syntactic types as maps, such as functions, acting on others of simpler types. Thus, in assigning syntactic types to the words, knowing how these map to semantic types and how these are

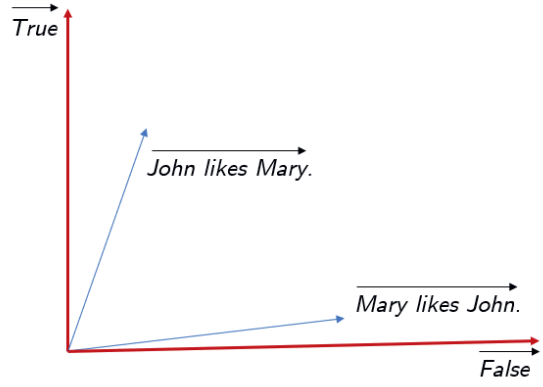


Figure 1.2: Vector interpretation of sentences.

interpreted, should suffice to arrive at the semantics of larger fragments of text. A big advantage of this approach is that it requires only the surface form of the input and of the types of the words, and also provides an independent interpretation of the building blocks of a sentence that can be treated in a compositional way.

1.2.5 *Distributional Compositional Semantics*

Let us now give a couple of examples of how this machinery acts when the domains are vector spaces. Take again "fluffy dog" and "John likes Mary". The constant fluffy′ is interpreted as an element of the space $N^* \otimes N$, and while the constant dog′ is an a vector in N. Given an orthonormal basis $\{\hat{e}_i\}$ of N, the dual vector space can be identified with N (the lifting of this restriction is studied in Chapter 2), these interpretations can be made explicit as

$$[\![\text{dog}']\!]_I = \sum_i d_i \, \hat{e}_i, \tag{1.27}$$

$$[\![\text{fluffy}']\!]_I = \sum_{jk} f_{jk} \, \hat{e}_j \otimes \hat{e}_k. \tag{1.28}$$

To arrive at the interpretation of the larger fragment by composition, we act with the matrix that represents the adjective on the vector that represents the noun, just as we apply a map to an argument, interpreting the application operation, which mirrors the syntactic cancellation rule, as matrix multiplication, in linear algebra equivalent to the inner product, following the methodology originally introduced in [33]:

$$\begin{aligned} [\![\text{fluffy}'[\text{dog}']]\!]_I &= [\![\text{fluffy}']\!]_I . [\![\text{dog}']\!]_I \\ &= \sum_{jk} f_{jk} \, \hat{e}_j \otimes \hat{e}_k \cdot \sum_i d_i \, \hat{e}_i = \sum_{jk} f_{jk} d_k \, \hat{e}_j. \end{aligned} \tag{1.29}$$

For the transitive sentence, the verb is a cube in the $N \otimes S \otimes N$ space, and as it contracts with the subject and object via matrix multiplication, a vector in the sentence space results. Note that because the verb cube might not be symmetric in the N components, it makes a difference whether the same words appear as subjects or objects (see Fig. 1.2). The general idea is that words become tensors of higher rank the more arguments they require.

This approach is especially interesting because of the fact that we can use the vectors that were obtained from the context in a compositional way. One of the limitations, though, is how quickly the dimensions of the tensors scale up, and therefore how applicable these ideas are using the currently available computational tools, when it comes to memory and processors. In these regards, quantum computation promises a number of new developments that are worth exploring.

1.3 QUANTUM MECHANICS

Another take at resolving the apparent incompatibilities between VSMs and probabilistic language models is a probabilistic interpretation of the vector entries of VSMs. As argued in [110], "quantum mechanics is already a clearly extant framework that combines both probabilistic and geometric insights, with coordinates of vectors being related to probability amplitudes", referring to the work of [138], that already argued that the probabilistic interpretation of quantum mechanics could be used for the vectors obtained in the context of information retrieval.

Moving to a quantum framework, words will no longer be represented just as vectors of any vector space, but as states on a Hilbert space. To understand this, it helps to be familiar with some of the notation and concepts of quantum mechanics. There are a number of reasons to extent the interpretation domains of the words in this way. One is that it allows for the introduction of complex valued parameters, which increases the amount of information that can be represented in a state when compared to a real-valued vector. Another is that we can talk about two types of probability distribution that contribute to the states coefficients, quantum and classical probabilities, which again expand the possible interactions between word representations. Lastly, using a quantum mechanics framework allows for the exciting idea that we can model words and phrases such that they can hold several ambiguous meanings simultaneously via *quantum superposition*. All these quantum properties have in recent years started to be implemented by using quantum computers, which will very soon be able to speed up certain computations that are currently costly, but this not without the ingenious new algorithms. The development of such algorithms for the processing of natural language is the aim of this thesis.

If the reader does not have a background in physics, all this might sound like a daunting proposition, so in this section we first give a colourful example of what a quantum system is, and what constitutes a state in such a system, before a compact overview of the mathematical concepts that we will make use of is given, leading to a precise understanding of what a quantum superposition is. Comprehensive references can be found

in Refs. [34, 102]. The understanding of these is at the heart of most of what is ultimately achieved in this thesis, and so understanding what follows, especially in terms of the notation introduced, is the golden ticket to understanding how a quantum mechanics interpretation of sentences can be of any benefit to the merging of linguistics with distributional models of meaning.

1.3.1 *Quantum Systems*

In what follows, a curious fact will be fleshed out: the reason why we can combine certain materials and set them on fire to get fireworks with shiny colors is the same one that allows scientists to figure out what materials compose the stars in the sky. It turns out that both these questions are connected to the quantum properties of the materials. We will explain this as a simple, and simplified, example of what a quantum system is, and how it could be useful for language processing.

The puzzle of how fireworks and stars are connected is one that scientists faced in the second half of the 19th century. By 1814, the physicist Fraunhofer had invented the spectroscope. When directed at a source of natural light, like a candle flame or the sun, it decomposed that light into the specific wavelengths, in nanometers, that added up to the color of the light that could be seen with the naked eye. It was known that the visible light was characterized by a wave, with each color corresponding to a specific wavelength, and it was already known, since Newton, via his prism decomposition of visible light, that the light coming from the sun contained the full spectrum of colors. In 1824, Fraunhofer published what happened when he pointed his spectrometer at the sun [45]. What he expected to see were all colors of the rainbow in a continuum, as in the first spectrum of Fig. 1.4. However, he observed instead something quite extraordinary: some very specific lines (wavelengths) were missing, as can be seen in the spectrum of Fig.1.3.

Something else that he realized was that his spectrometer allowed the colors emitted by the gases of certain heated materials to be measured exactly,

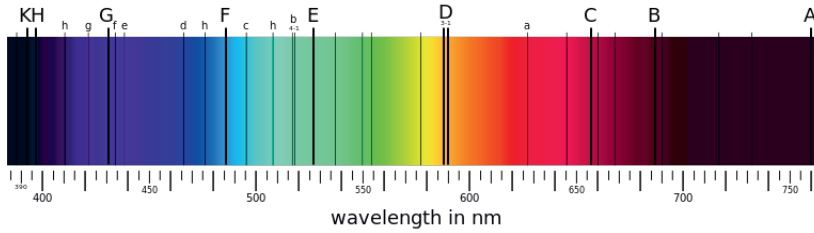


Figure 1.3: Absorption spectrum of the sun.

and that each material had a specific signature in terms of the lines that it produced (Fig. 1.4).

By comparing both the spectrum of the sun and the spectra of different materials, he found that what was missing in the former found a perfect match with what was found in the latter. If each material had an absorption spectrum that corresponded to the negative of the emission spectrum (see Fig. 1.5), then this could make it possible to figure out what materials composed the atmosphere of the sun: a cold gas of a certain material in the sun's atmosphere absorbed light of the exact same wavelengths than the heated gas of that same material, emitted in his lab experiments. So by mapping out the spectral lines of each element and comparing their absorption with the lines missing in the full visible spectrum coming from the sun, it was possible to figure out what the atmosphere of the sun is made of: mainly hydrogen and helium.

In the same way, we can understand that fireworks get their colors for the same reason: if we want a yellow firework, we do it by burning sodium salts, and if we want a green one, we can use copper salts. We can rest assured that these materials will always emit the same colors.

But how to explain such reliability in the colors? The answer came from quantum mechanics. The first finding was that the light that we see is not only a wave, but that it is actually composed by a beam of particles, and that each different color corresponds to a particle that is vibrating with a certain wavelength. These light particles are called *photons*. So this means that if a photon has a certain wavelength, making a beam of more particles exactly like that one will not change the color of the light

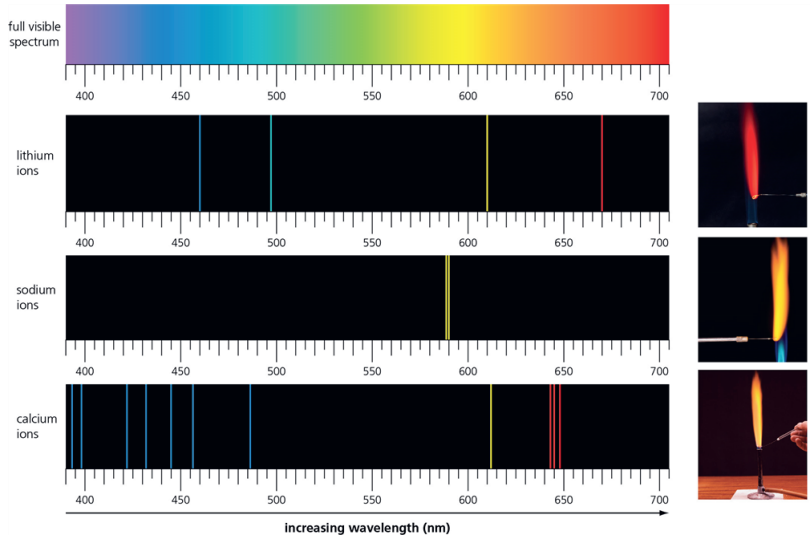


Figure 1.4: Flame spectroscopy of different elements.

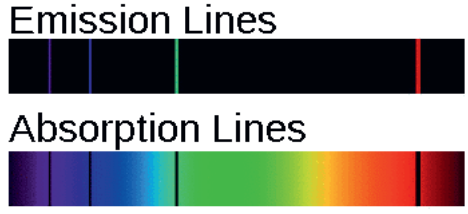


Figure 1.5: Emission vs absorption spectrum.

beam, but only its intensity. The second assertion was that each particle of matter, *atoms*, could only interact with light particles with a very specific wavelength, either in absorbing or emitting them. Since different materials, are made up of different types of atoms, these different materials also interact with light particles of different wavelengths. That is, for example, if photons pass through *atoms* of hydrogen in the atmosphere of the sun, those atoms will only care about the photons that vibrate at very specific wavelengths, retaining them and so not allowing them to travel until they reach us and can be observed. Since these atoms don't interact with the photons of other wavelengths, these remain invisible to them. Similarly with the emission spectrum, if we heat up the atom, only when it reaches a certain temperature can a light particle be released. This is because both the temperature and the photon emission are related with the *energy* available, so that *thermal* energy is absorbed, and then emitted as a photon, of which the wavelength represents a very specific energy. The rest of the energy, that is not emitted as light, is released as heat. And so, there is no emission of photons with intermediate wavelengths, and we are bound to see the colors that we see. Almost a century after Fraunhofer's results, in 1913 Bohr used the recently developed theory of quantum mechanics to show that wavelengths of the emission spectrum of the hydrogen corresponded exactly to the differences in energy between the states of the hydrogen atom [19] (Fig. 1.6). Incidentally, it is worth mentioning that the Physics Institute in Utrecht had an important role to play in the ever more precise measurement of the spectral lines around this time, chiefly under Ornstein, contributing to the validation of the new theory of quantum mechanics [55, 148].

The theory of quantum mechanics explains this by asserting the following: each quantum system has a number of allowed *states* in which it can be observed, and all of these states together define the system. As such, there is a number of states that they can transition to-and-fro. In the context of atoms, the states are associated with specific amounts of energy, which are then different for atoms of varying elements.

That means that each atom only deals with photons of specific energies, or not at all. Each time the atom goes into a *lower* energy state, it lets go of

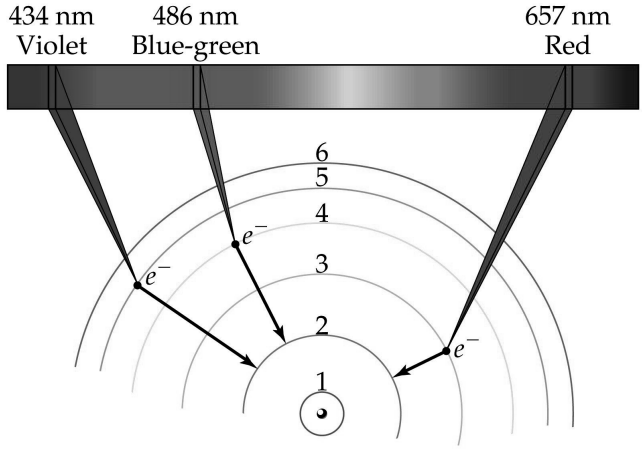


Figure 1.6: Some hydrogen transition lines in the visible spectrum. The energy transition is due to a change in the state of its electron.

the excess energy by emitting a photon of a wavelength that corresponds exactly to that energy difference. On the contrary, in order to go to a higher energy state, the atom has to absorb a photon that contains the exact energy necessary to compensate for the energy difference. Now we can explain why these emissions always happen at specific wavelengths: there are no in-between states for the atoms to be in, and two photons with half of the necessary energy will typically *not* do the job.

But what is truly remarkable in a quantum system is that, while *which* states it can be in are perfectly well established, *whether* it will be in a particular state is a probabilistic issue. This means that if a photon with the exact amount of energy for an atomic state transition comes along, this transition will only happen with a certain probability, since the atom can be in a linear superposition of the states before and after the absorption of the photon. In what follows we will make this assertion more precise, by moving to a more mathematical treatment of a quantum system.

1.3.2 *Quantum formalism*

Suppose that we have a quantum system Ψ. This can be described using n independent states, with n the dimension of the Hilbert space (a complex-valued vector space with an inner product) where it is being described. When a measurement is performed, the system is observed in exactly one and only one of these states. To describe this mathematically, we set these states to be the orthonormal basis vectors of the Hilbert space.

Before the measurement, the system can be in a *quantum superposition* of states. While this term is often thrown around as sort of mystical property, let me take this chance to break the spell, and hopefully convince you that this is simpler than it sounds: purely from a mathematical definition point of view, a quantum superposition is nothing but a state that can be described as a linear combination of basis states. It is, therefore, *any* vector in a Hilbert space other than a basis state.

The story starts to get a little bit more interesting when we consider what the coefficients of that linear combination are. These turn out to be related with the probabilities that the system is observed in the corresponding basis states. Again, this makes it sound spookier than it is. In fact, the best that quantum mechanics gives is a probability for obtaining the result of a measurement, these probabilities can actually be calculated very precisely, in the way that we will introduce shortly, and have been extensively verified time and time again in experiments with unprecedented accuracy.

Another special feature of quantum mechanics is that, after a measurement, the system is described exactly by the basis state in which it was observed, and no longer by the previous superposition, which is destroyed by the measurement process. Again, mathematically, this is just the same as saying that the measurement corresponds to a *projection* to a basis state.

In the case of the atom, the emission of a photon of a certain wavelength corresponds to an observation, and it allows us to know exactly the lower energy state in which the atom is now in. However, in the moments right before or right after, the atom is described by a quantum superposition of possible states (see Fig. 1.7).

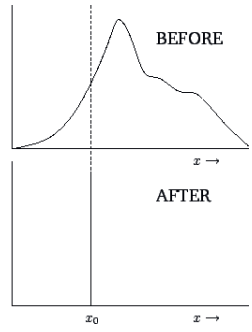


Figure 1.7: Collapse of the wave function.

Let us now make the previous remarks more precise, and introduce some notation for that. The ith basis state of a Hilbert space A is represented by $|i\rangle_A$, which is a vector in the vector space A and is called a *ket*. Dual to the ket is the *bra*, represented by ${}_A\langle i|$ and given by the conjugate transpose of the ket, which lives in the dual vector space A^*[c]:

$$|\Psi\rangle_A = \sum_i C_j \, |i\rangle_A \,, \tag{1.30}$$

$${}_A\langle\Psi| = \sum_i C_i^* \, {}_A\langle i|. \tag{1.31}$$

Take as an example the Hilbert space that has two orthonormal basis states $|0\rangle$ and $|1\rangle$. Then a quantum system Ψ that can be in one of these two states can be represented as a superposition state in this space as

$$|\Psi\rangle = C_0 \, |0\rangle + C_1 \, |1\rangle \,. \tag{1.32}$$

This state reads as "The quantum system Ψ is, with certainty, in state $|\Psi\rangle$."

The coefficients of the linear combination are related with the probability p_i that the system is measured in each possible state by the modulus squared of the corresponding coefficient, $p_i = |C_i|^2$. This means, for example, that

c It should be clear that the bra belongs to A^*. However, to make the notation less heavy, we write simply A in the subscript.

the system Ψ will be observed in state $|0\rangle$ with probability $|C_0|^2$. Thus, in order for the probabilities to add up to 1, the sum of all $|C_i|^2$ has to be 1. This restricts the values of the coefficients. In order for the probability of measuring system Ψ in either state to be equal, the state that it has to be in is

$$|\Psi\rangle_{eq} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}e^{i\phi}|1\rangle, \tag{1.33}$$

where we see the complex-valued exponent introducing a *relative phase* between the states.

Using that $\sum_j |j\rangle_A\langle j| = \mathbb{1}$, the general state of the quantum system can be rewritten with respect to this basis as

$$|\Psi\rangle_A = \sum_{ij} C_i |j\rangle_A\langle j|i\rangle_A. \tag{1.34}$$

Now we move to the formalism of density matrices, where we can understand what we talk about when we distinguish between classical and quantum probabilities. The density matrix is an object in a tensor product space defined as the tensor product between the possible states of the system and the respective duals, weighted by the corresponding probability p'_j of being in that state:

$$\rho = \sum_j p'_j |\Psi_j\rangle_A\langle\Psi_j|. \tag{1.35}$$

Since this is the tensor product between two states, this is a *pure* density matrix when p'_j is 1 for a particular value of j, and 0 otherwise. Our previous system Ψ can be described using a pure density matrix, using Eq. (1.32) as

$$\rho_\Psi^{pure} = |\Psi\rangle\langle\Psi| = |C_0|^2|0\rangle\langle 0| + C_0C_1^*|0\rangle\langle 1| + C_0^*C_1|1\rangle\langle 0| + |C_1|^2|1\rangle\langle 1|. \tag{1.36}$$

This product state can be presented by a matrix as well, which has the coefficients as elements:

$$\rho_{\Psi}^{pure} = \begin{pmatrix} |C_0|^2 & C_0 C_1^* \\ C_0^* C_1 & |C_1|^2 \end{pmatrix}. \tag{1.37}$$

So we see that there are two types of probability: p', as in Eq. 1.35, is a *classical* probability. In the case of Eq. 1.37, we *know* with certainty what is the state in which system Γ is, which is $|\Gamma\rangle$, and so $p' = 1$. Then the elements in the diagonal of the density matrix are the *quantum* probabilities, which are, as we have seen, the probabilities that the system is *observed*, or measured, in a certain basis state.

With respect to the basis vectors, the density matrix can be expressed as

$$\rho = \sum_{ii'} X_{ii'} \, |i\rangle_A \otimes {}_A\langle i'| \equiv \sum_{ii'} X_{ii'} \, |i\rangle \, {}_A\langle i'| \in A \otimes A^*. \tag{1.38}$$

It is also possible that at least p'_j are different from zero, in which case we talk about a *mixed* state, which makes for an entirely diagonal density matrix if the states are basis states. In this case, the values of the probabilities p'_j correspond to classical probabilities. For instance, the system Ψ could be at another moment described as

$$\rho_{\Gamma}^{mix} = p'_0 \, |0\rangle \langle 0| + p'_1 \, |1\rangle \langle 1| . \tag{1.39}$$

This state can be read as "The quantum system Ψ is either in state $|0\rangle$, with probability p'_0, or in state $|1\rangle$, with probability p'_1." Compare this with the pure state reading of Eq. 1.32. In this difference lies on of the reasons why quantum systems show such wonderful behaviour: if we think that the interactions between quantum systems involve quantum state multiplications, as we will see below, which can be expressed as matrix multiplication, we see that, if these matrices are diagonal (classical), the diagonal values of the interacting system are readily computable; but since this is not necessarily the case, the off-diagonal terms of each system contribute to the diagonal terms of the system after the interaction, and, as

we have seen, these are the values that govern the actual observations. This has the important consequence too that the order of the multiplications can matter, which suits well with the idea the order in which the word interpretations are composed should be meaningful.

1.3.3 *Quantum States as Distributional Semantics*

Let us come back to the agenda of this thesis, which is to use this machinery to describe sentence-level ambiguities as quantum superpositions, so we can take advantage of the quantum properties of the system.

One way to represent words as quantum objects starts by defining the domains of the semantic types using Hilbert spaces and words as states therein. If for instance n is interpreted in the Hilbert space $\mathcal{H}^N$, then

$$[\![\text{dog}']\!]_{I'} = \sum_i d_i \, |i\rangle \,, \tag{1.40}$$

$$[\![\text{fluffy}']\!]_{I'} = \sum_{jk} f_{jk} \, |jk\rangle \,, \tag{1.41}$$

resulting in the previous noun contraction. Implementing this contraction in a quantum computer is a topic of exploration of Chapter 5.

But furthermore, the ability to include classical probabilities as well is naturally embedded in the formalism. The previous formalism only regards states that are pure, and that is not without its limitations. Moving to mixed states, which can only be done in the formalism of density matrices, has been proposed as a way to represent ambiguities in word senses in one single representation of that word [9, 85, 109, 119], something that cannot be done with simple VSMs. We want to bring this success to a full-fledged representation of words and their composition using density matrices that also represent sentence-level ambiguities, such as the syntactic and structural ambiguities that we look at in Chapters 2 and 3, with examples given below. In this setting, the domains of basic semantic types are the tensor-product vector space where density matrices are described (see also

Ref. [126]). Some technicalities of the handling of density matrices can be found in Appendix B. In this setting, the domains of the semantic types are linear maps between these spaces, and our previous examples now become

$$[\![\text{dog}']\!]_{I''} = \sum_{ii'} D_{ii'} \left|i\right\rangle \left\langle i'\right| , \tag{1.42}$$

$$[\![\text{fluffy}']\!]_{I''} = \sum_{jj'kk'} F_{jj',kk'} \left|jk\right\rangle \left\langle j'k'\right| . \tag{1.43}$$

The interpretation of the composition, that related the interpretations of the individual words in the image of the syntactic composition, is given by a contraction between tensors, given as the partial trace over the space where that contraction takes place, in this particular case the space where nouns are interpreted, $\mathbb{N} = \mathcal{H}^{N*} \otimes \mathcal{H}^{N}$:

$$\begin{aligned} [\![\text{fluffy}'[\text{dog}']]\!]_{I''} &= \text{Tr}_{\mathcal{N}} \left([\![\text{fluffy}']\!]_{I''} . [\![\text{dog}']\!]_{I''} \right) \\ &= \sum_{ii',kk'} F_{jj',kk'} D_{k'k} \left|j\right\rangle \left\langle j'\right| . \end{aligned} \tag{1.44}$$

This will be basis for the rest of the chapters in this thesis. In Chapter 2 we introduce a modification of this formalism to include directionality in the semantic type domains to study a certain number of sentence-level ambiguities and how to keep them in quantum superposition. The following chapters are, each in their own way, its spin-offs: in Chapter 3 we extend the formalism to include other types of ambiguities present in Dutch; in Chapter 4 we numerically implement an aspect of the formalism essential to the directionality of the interpretations; and in Chapter 5 we go through the ropes of quantum computation to implement the quantum state contractions that allow us to keep the several readings of ambiguous sentences in quantum superposition.

1.4 OUTLOOK

In **Chapter 2** we look at the phenomenon of syntactic ambiguities. Sentences like *Babies are what the mother eats*, *Crowds rush to see Pope trample man to death* or *Killer sentenced to die for second time in 10 years* have appeared as real headlines of newspaper or magazine articles [22], and are instances of such linguistic phenomena. While from the reader's side these headlines can be undoubtedly amusing, more seriously intended communication can be harmed them, and so approaching them from a syntactic perspective offers a level of control that simple "bag-of-words" cannot. Examples include the interpretation of the law [124], or the understanding of incomplete medical notes, also known as 'doctor scribbles' [43]. What is at hand in these cases is that different syntactic contractions can take place, and under the view that the semantics and syntax are connected, we can study the different meanings that arise from different syntactic contractions. The focus will be a phrase of the form of *old women and men*. The scope of "old" in the noun phrase should become clear from context if it is followed by the segment *can't get pregnant*.

In **Chapter 3** we extend the syntax of the previous chapter, the Lambek calculus, with modalities that allow us to control the use of the structural commutativity rule, essential to the derivation of structural ambiguities in verb-final structures that exist is Dutch relative clauses. We aim to give a quantum state semantic interpretation of these modalities, by introducing an extra vector space, akin to a spin space, that keeps track of the use of the commutativity structural rule, once more giving us control over which reading is intended at the level of the interpretation. We do this as an extension of the more general framework of directional density matrices introduced in the previous chapter.

In **Chapter 4**, we explore the idea already proposed in the first chapter that the contractions between the vector representations of words should not be done using the Euclidean inner product, as done in the [33] and related work, but that within certain contexts the ways in which vectors relate should be refined. We use a machine learning algorithm to fine-tune the inner products between word representations that belong to similar

contexts, achieving an improvement on the correlation between the resulting cosine similarities and human-reported similarity judgements.

Finally, in **Chapter 5**, introduce the way to the contractions on a quantum computer, to describe the different ambiguity readings using quantum superposition. We first show how this works for the syntactic ambiguities of the first chapter, and then use the fact that we know how to input words as quantum states to develop a quantum search algorithm, Grover's algorithm, to find the correct answer to a natural language question, directly from the representation of this question generated by using a contraction of quantum states that is compositional according to what we have introduced here.

2

DENSITY MATRICES WITH METRIC FOR DERIVATIONAL AMBIGUITY

ABSTRACT Recent work on vector-based compositional natural language semantics has proposed the use of density matrices to model lexical ambiguity and (graded) entailment (e.g. Piedeleu et al 2015, Bankova et al 2019, Sadrzadeh et al 2018). Ambiguous word meanings, in this work, are represented as mixed states, and the compositional interpretation of phrases out of their constituent parts takes the form of a strongly monoidal functor sending the derivational morphisms of a pregroup syntax to linear maps in FdHilb. Our aims in this paper are threefold. Firstly, we replace the pregroup front end by a Lambek categorial grammar with directional implications expressing a word's selectional requirements. By the Curry-Howard correspondence, the derivations of the grammar's type logic are associated with terms of the (ordered) linear lambda calculus; these terms can be read as programs for compositional meaning assembly with density matrices as the target semantic spaces. Secondly, we extend on the existing literature and introduce a symmetric, nondegenerate bilinear form called a "metric" that defines a canonical isomorphism between a vector space and its dual, allowing us to keep a distinction between left and right implication. Thirdly, we use this metric to define density matrix spaces in a directional form, modeling the ubiquitous derivational ambiguity of natural language syntax, and show how this allows an integrated treatment of lexical and derivational forms of ambiguity controlled at the level of the interpretation.

2.1 INTRODUCTION

Semantic representations of language using vector spaces are an increasingly popular approach to automate natural language processing, with early comprehensive accounts given in [27, 91]. This idea has found several implementations, both theoretically and computationally. On the theoretical side, the principle of compositionality [62] states that the meaning of a complex expression can be computed from the meaning of its simpler building blocks and the rules used to assemble them. On the computational side, the distributional hypothesis [54] asserts that a meaning of a word is adequately represented by looking at what words most often appear next to it. Joining these two approaches, a distributional compositional categorical (DisCoCat) model of meaning has been proposed [33], mapping the pregroup algebra of syntax to vectors spaces with tensor operations, by functorialy relating the properties of the categories that describe those structures, allowing one to interpret compositionality in a grammar-driven manner using data-extracted representations of words that are in principle agnostic to grammar. This method has been shown to give good results when used to compare meanings of complex expressions and with human judgements [51]. Developments in the computation of these vectors that use machine learning algorithms [89] provide representations of words that start deviating from the count-based models. However, each model still provides a singular vector embedding for each word, which allows the DisCoCat model to be applied with some positive results [144].

The principal limitation of these embeddings, designated *static* embeddings, is that it provides the same word representation independently of context. This hides polysemy, or even subtler gradations in meaning. Using the DisCoCat framework, this issue has been tackled using density matrices to describe lexical ambiguity [108**?**], and using the same framework also sentence entailment [119] and graded hyponymy [9], since the use of matrices allows the inclusion of correlations between context words. From the computational side, the most recent computational language models [38, 107] present contextual embeddings of words as an intrinsic feature. In this paper we aim at reconciling the compositional distributional model

and these developments by presenting density matrices as the fundamental representations of words, thus leveraging previous results, and by introducing a refined notion of tensor contraction that can be applied even if we do not assume that we are working with static embeddings coming from the data, thus additionally presenting the possibility of eliminating the distinction between context and target words, because all words can be equally represented with respect to one another. To achieve this, we build the components of the density matrices as covariant or contravariant by introducing a metric that relates them, extending to the interpretation space the notion of directionality of word application, as a direct image of the directional Lambek calculus. After that, we attach permutation operations that act on either type of components to describe derivational ambiguity in a way that keeps multiple readings represented in formally independent vector spaces, thus opening up the possibility of integration between lexical and syntactic ambiguity.

Section 2.2 introduces our syntactic engine, the Lambek calculus **(N)L**$_{/,\backslash}$, together with the Curry-Howard correspondence that associates syntactic derivations with programs of the ordered lambda calculus $\lambda_{/,\backslash}$. Section 2.3 motivates the use of a more refined notion of inner product and introduces the concept of a tensor and tensor contraction as a basis independent application of a dual vector to a vector, and introduces a metric as the mechanism to go from vectors extracted from the data to the dual vectors necessary to perform tensor contraction. Section 2.4 gives some background on density matrices, and on ways of capturing the directionality of our syntactic type logic in these semantic spaces using the previously described metric. Section 2.5 then turns to the compositional interpretation of the $\lambda_{/,\backslash}$ programs associated with **(N)L**$_{/,\backslash}$ derivations. Section 2.6 shows how the directional density matrix framework can be used to capture simple forms of derivational ambiguity.

2.2 FROM PROOFS TO PROGRAMS

With his [68, 69] papers, Jim Lambek initiated the 'parsing as deduction' method in computational linguistics: words are assigned formulas of a type logic designed to reason about grammatical composition; the judgement whether a phrase is well-formed is the outcome of a process of deduction in that type logic. Lambek's original work was on a calculus of *syntactic* types, which he presented in two versions. With $\mathbf{L}_{/,\backslash}$ we refer to the simply typed (implicational) fragment of Lambek's [68] associative syntactic calculus, which assigns types to *strings*; $\mathbf{NL}_{/,\backslash}$ is the non-associative version of [69], where types are assigned to *phrases* (bracketed strings).[a]

Van Benthem [15] added semantics to the equation with his work on **LP**, a commutative version of the Lambek calculus, which in retrospect turns out to be a precursor of (multiplicative intuitionistic) linear logic. **LP** is a calculus of *semantic* types. Under the Curry-Howard 'proofs-as-programs' approach, derivations in **LP** are in 1-to-1 correspondence with terms of the (linear) lambda calculus; these terms can be seen as *programs* for compositional meaning assembly. To establish the connection between syntax and semantics, the Lambek-Van Benthem framework relies on a homomorphism sending types and proofs of the syntactic calculus to their semantic counterparts.

In this paper, rather than defining semantic interpretation on a commutative type logic such as **LP**, we want to keep the distinction between the left and right implications $\backslash, /$ of the syntactic calculus in the vector-based semantics we aim for. To achieve this, our programs for meaning composition use the language of Wansing's [141] *directional* lambda calculus $\lambda_{/,\backslash}$. Wansing's overall aim is to study how the derivations of a family of substructural logics can be encoded by typed lambda terms. Formulas, in the substructural setting, are seen as information pieces, and the proofs manipulating these formulas as information processing mechanisms, subject

a Neither of these calculi by itself is satisfactory for modelling natural language syntax. To handle the well-documented problems of over/undergeneration of **(N)L**$_{/,\backslash}$ in a principled way, the logics can be extended with modalities that allow for controlled forms of reordering and/or restructuring. We address these extensions in [36].

Terms: $t, u ::= x \mid \lambda^r x.t \mid \lambda^l x.t \mid t \triangleleft u \mid u \triangleright t$

Typing rules:

$$\frac{}{x : A \vdash x : A}\ Ax$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda^r x.t : B/A}\ I/ \qquad \frac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda^l x.t : A\backslash B}\ I\backslash$$

$$\frac{\Gamma \vdash t : B/A \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash t \triangleleft u : B}\ E/ \qquad \frac{\Gamma \vdash u : A \quad \Delta \vdash t : A\backslash B}{\Gamma, \Delta \vdash u \triangleright t : B}\ E\backslash$$

Figure 2.1: Proofs as programs for **(N)L**$_{/,\backslash}$.

to certain conditions that reflect the presence or absence of structural rules. The terms of $\lambda_{/,\backslash}$ faithfully encode proofs of **(N)L**$_{/,\backslash}$; information pieces, in these logics, cannot be copied or deleted (absence of Contraction and Weakening), and information processing is sensitive to the sequential order in which the information pieces are presented (absence of Permutation).

We present the rules of **(N)L**$_{/,\backslash}$ with the associated terms of $\lambda_{/,\backslash}$ in Fig 3.1. The presentation is in the sequent-style natural deduction format. The formula language has atomic types (say *s, np, n* for sentences, noun phrases, common nouns) for complete expressions and implicational types $A\backslash B$, B/A for incomplete expressions, selecting an A argument to the left (resp. right) to form a B.

Ignoring the term labeling for a moment, judgments are of the form $\Gamma \vdash A$, where the antecedent Γ is a non-empty list (for **L**) or bracketed list (**NL**) of formulas, and the succedent a single formula A. For each of the type-forming operations, there is an Introduction rule, and an Elimination rule.

Turning to the Curry-Howard encoding of **NL**$_{/,\backslash}$ proofs, we introduce a language of directional lambda terms, with variables as atomic expressions,

left and right λ abstraction, and left and right application. The inference rules now become *typing* rules for these terms, with judgments of the form

$$x_1 : A_1, \ldots, x_n : A_n \vdash t : B. \tag{2.1}$$

The antecedent is a typing environment providing type declarations for the variables x_i; a proof constructs a program t of type B out of these variables. In the absence of Contraction, Weakening and Permutation structural rules, the program t contains $x_1, \ldots, x_n$ as free variables exactly once, and in that order. Intuitively, one can see a term-labelled proof as an algorithm to compute a meaning t of type B with parameters x_i of type A_i. In parsing a particular phrase, one substitutes the meaning of the constants (i.e. words) that make it up for the parameters of this algorithm.

2.3 DIRECTIONALITY IN INTERPRETATION

In order to introduce the directionality of the syntactic calculus in the semantic calculus, we expand on the existing literature that uses **FdVect** as the interpretation category by calling attention to the implied inner product. We introduce a more abstract notion of tensor, tensor contraction and the need to introduce explicitly the existence of a metric, coming from the literature of general relativity, following the treatment in [139].[b] Formally, a metric is a function that assigns a distance between two elements of a set, but if applied to the elements of a set that is closed under addition and scalar multiplication, that is, the elements of a vector space, it becomes an inner product. Since we will be looking at vector spaces, we use the terms metric and inner product interchangeably.

To motivate the need for a more careful treatment regarding the inner product, lets look at a very simple yet illustrative example. Suppose that a certain language model provides word embeddings that correspond to two-dimensional, real valued vectors. In this model, the words "vase" and "wall" have the vector representations $\vec{v}$ and $\vec{w}$, respectively

b An alternative introductory treatment of tensor calculus can be found in [41].

$$\vec{v} = (0,1) \quad \text{and} \quad \vec{w} = (1,0). \tag{2.2}$$

This representation could mean that they are context words in a count-based model, since they form the standard (orthogonal) basis of $\mathbb{R}^2$, or that they have this particular representation in a particular context-dependent language model. To compute cosine similarity, the notion of Euclidean inner product is used, where the components corresponding to a certain index are multiplied:

$$\vec{v} \cdot \vec{w} = 0 \cdot 1 + 1 \cdot 0 = 0, \tag{2.3}$$

which we can use to calculate the cosine of the angle θ between these vectors,

$$\cos(\theta) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|} = \frac{0 \cdot 1 + 1 \cdot 0 = 0}{1 \cdot 1} = 0. \tag{2.4}$$

Thus, if the representations of these words are orthogonal, then using this measure to evaluate similarity we conclude that these words are not related. However, there is a degree of variation in the vectors that are assigned to the distributional semantics of each word. *Static embeddings* are unique vector representations given by a global analysis of a word over a corpus. The unique vector assigned to the semantics of a word depends on the model used to analyze the data, so different models do not necessarily put out the same vector representations. Alternative to this are *dynamic embeddings*, which assign different vector representations to the same word depending on context, within the same model.

Therefore, there are at least three ways in which the result of eq.4.1 and subsequent interpretation can be challenged:

1. **Static Embeddings.** If the representations come from a count-based model, choosing other words as context words changes the vector representation and therefore these words are not orthogonal to one another anymore; in fact this can happen with any static embedding

representation when the basis of the representation changes. Examples of models that give static embeddings are Word2Vec [89] and GloVe [106].

2. **Dynamic Embeddings.** When the vector representations comes from a context-dependent embedding, changing the context in which the words are evaluated influences their representation, which might not be orthogonal anymore. Dynamic embeddings can be obtained with i.e. ELMo[107] BERT[38] and GPT-2[111].
3. **Expectation of meaning.** Human judgements, which are the outcomes of experiments where subjects are explicitly asked to rate the similarity of words, predict that some words should have a degree of relationship. Therefore, the conclusion with respect to similarity we derive from orthogonal representations of certain words might not be valid if there is a disagreement with their human assessment. These judgements are condensed in datasets such as the MEN dataset [21].

While points 1 and 2 can be related, caution is necessary in establishing that link. On a preliminary inspection, comparing the cosine similarity of context-free embeddings of nouns extracted from pre-trained BERT [38] with the normalized human judgements from the MEN dataset [21], we find that the similarity between two words given by the language model is systematically overrated when compared to its human counterpart. One possible explanation is that the language model is comparing all words against one another, so it is an important part of similarity that the two words belong to the the same part of speech, namely nouns, while humans assume that as a condition for similarity evaluation. Further, though we can ask the language model to rate the similarity of words in specific contexts, that has not explicitly been done with human subjects. A more detailed comparison between context-depend representations and human judgement constitutes further research.

One way to reconcile the variability of representations and the notion of similarity is to expand the notion of inner product to be invariant under the change of representations. Suppose now that by points 1 or 2 the representations of "vase" and "wall" change, respectively, to

$$\vec{v}' = (1,1),\ \vec{w}' = (-1,2). \tag{2.5}$$

These vectors also form a basis of $\mathbb{R}^2$, but not an orthogonal one. If we use the same measure to compute similarity, taking normalization into account, the Euclidean inner product gives $\vec{v}' \cdot \vec{w}' = (-1) \cdot 1 + 1 \cdot 2 = 1$ and cosine similarity gives

$$cos(\theta') = \frac{\vec{v}' \cdot \vec{w}'}{\|\vec{v}'\| \cdot \|\vec{w}'\|} = \frac{1}{\sqrt{2} \cdot \sqrt{5}} = \frac{1}{\sqrt{10}}. \tag{2.6}$$

If now we have a conflict between which representations are the correct ones, we can look at the human evaluations of similarity. Suppose that it corresponds too to $\frac{1}{\sqrt{10}}$.

We argue in this paper that, by introducing a different notion of inner product, we can fine-tune a relationship between the components of the vectors with the goal to preserve a particular value, for example a human similarity judgement. In this framework, the different representations of words in dynamic embeddings are brought about by a change of basis, similarly to what happens when the context words change in static embeddings, in which case the value of the inner product should be preserved. This can be achieved by describing the inner product as a tensor contraction between a vector and a dual vector, with the latter computed using a metric.

Let V be a finite dimensional vector space and let V^* denote its dual vector space, constituted by the linear maps from V to the field $\mathbb{R}$. A tensor T of type (k, l) over V is a multilinear map

$$T : \underbrace{V^* \times \cdots \times V^*}_{k} \times \underbrace{V \times \cdots \times V}_{l} \to \mathbb{R}. \tag{2.7}$$

Once applied on k dual vectors and l vectors, a tensor outputs an element of the field, in this case a real number. By this token, a tensor of type $(0,1)$ is a dual vector, which is the map from the vector space to the field, and a tensor of type $(1,0)$, being technically the dual of a dual vector, is naturally isomorphic to a vector. Given a basis $E = \{\hat{e}_i\}$ in V and its dual basis ${}_dE =$

$\{\hat{e}^j\}$ in V^*, with $\hat{e}^j(\hat{e}_i) = \delta_i^j$, the tensor product between the basis vectors and dual basis vectors forms a basis $B = \{\hat{e}_{i_1} \otimes \cdots \otimes \hat{e}_{i_k} \otimes \hat{e}^{j_1} \otimes \cdots \otimes \hat{e}^{j_l}\}$ of a tensor of type (k, l), allowing the tensor to be expressed with respect to this basis as

$$T = \sum_{i_1,\ldots,i_k,j_1,\ldots,j_l} T^{i_1\ldots i_k}{}_{j_1\ldots j_l} \hat{e}_{i_1} \otimes \cdots \otimes \hat{e}_{i_k} \otimes \hat{e}^{j_1} \otimes \cdots \otimes \hat{e}^{j_l}. \tag{2.8}$$

The basis expansion coefficients $T^{i_1\ldots i_k}{}_{j_1\ldots j_l}$ are called the *components* of the tensor.

We can perform two important operations on tensors: apply the tensor product between them, $T' \otimes T$, and contract components of the tensor, CT. The first operation happens in the obvious way, while the second corresponds to applying one of the basis dual vectors to a basis vector, resulting in an identification and summing of the corresponding components:

$$(CT)^{i_1\ldots i_{k-1}}{}_{j_1\ldots j_{l-1}} = \sum_\sigma T^{i_1\ldots\sigma\ldots i_{k-1}}{}_{j_1\ldots\sigma\ldots j_{l-1}}. \tag{2.9}$$

The outcome is a tensor of type $(k-1, l-1)$. Note that this procedure is basis independent, because of the relationship between the basis and dual basis. For a tensor of type $(1, 1)$, which represents a linear operator from V to V, tensor contraction corresponds precisely to taking the trace of that operator. To simplify the notation, we will use primed indices instead of numbered ones when the tensors have a low rank. We define a special $(0, 2)$ tensor called a *metric* d:

$$d = \sum_{j,j'} d_{jj'} \hat{e}^j \otimes \hat{e}^{j'}. \tag{2.10}$$

This tensor is symmetric and non-degenerate. The contraction of this tensor with two vectors v and w gives the value of the inner product:

$$d(v, w) = \sum_{j,j'} v^j d_{jj'} w^{j'}. \tag{2.11}$$

Because of symmetry, $d(v,w) = d(w,v)$, and because of non-degeneracy, the metric is invertible, with its inverse d^{-1} expressed as

$$d^{-1} = \sum_{i,i'} d^{ii'} \hat{e}_i \otimes \hat{e}_{i'}. \tag{2.12}$$

Given that the elements extracted from the data are elements of V, the contractions that need to be performed, for example for the application of the compositionality principle in vector spaces, must involve a passage from vectors to dual vectors as seen in the DisCoCat model, before contraction takes place. The metric can be used to define a canonical map between V and V^* via the partial map that is obtained when only one vector is used as an argument of the metric, giving rise to the dual vector ${}_d v : v \mapsto d(-,v)$, with the slash indicating the empty argument slot:

$$d(v,w) \equiv d(v,-)(w) \equiv {}_d v(w). \tag{2.13}$$

This formulation is basis independent, since it results from tensor contraction. Once a basis is defined, the resulting dual vector can be expressed as

$$v^d = \sum_{i,j,j'} d_{jj'} v^i \hat{e}^j \otimes \hat{e}^{j'}(\hat{e}_i) = \sum_{j,j'} d_{jj'} v^{j'} \hat{e}^j = \sum_{j'} v_{j'} \hat{e}^{j'}, \tag{2.14}$$

where we rewrite $v_{j'} = \sum_j d_{jj'} v^{j'}$.

We call the components of vectors, with indices "up", the *contravariant* components, and those of dual vectors, with indices "down", the *covariant* components. Thus, consistent with our notation, the metric can be used to "lower" or "raise" indices, applying contraction between the metric and the tensor and relabeling the components:

$$
\begin{aligned}
d(T) &= \sum_{i_1,\ldots,i_k,j_1,\ldots,j_{l+2}} d_{j_{l+1},j_{l+2}} T^{i_1,\ldots,i_k}{}_{j_1,\ldots,j_l} \hat{e}^{j_{l+1}} \otimes \hat{e}^{j_{l+2}}(\hat{e}_{i_1}) \otimes \ldots \otimes \hat{e}^{j_l} \\
&= \sum_{i_1,\ldots,i_k,j_1,\ldots,j_{l+1}} d_{j_{l+1},i_1} T^{i_1,\ldots,i_k}{}_{j_1,\ldots,j_l} \hat{e}^{j_{l+1}} \otimes \hat{e}_{i_2} \otimes \ldots \otimes \hat{e}^{j_l} \\
&= \sum_{i_2,\ldots,i_k,j_1,\ldots,j_{l+1}} T_{j_{l+1}}{}^{i_2,\ldots,i_k}{}_{j_1,\ldots,j_l} \hat{e}^{j_{l+1}} \otimes \hat{e}_{i_2} \otimes \ldots \otimes \hat{e}^{j_l}. \qquad (2.15)
\end{aligned}
$$

The effect of the metric on a tensor can be captured by seeing how we rewrite the components of some example tensors:

- $\sum_{j'} d_{jj'} T^{j'}{}_{j''} = T_{jj''}$;
- $\sum_{i'} T^{i}{}_{i'} d^{i'i''} = T^{ii''}$;
- $\sum_{j',j'''} d_{jj'} d_{j''j'''} T^{j'j'''} = T_{jj''}$.

Most importantly, a proper tensor is only defined in the form of eq.2.8, so whenever we have a tensor that has components "up" and "down" in different orders, for example in $T_j{}^i$, this is in fact a tensor of type $(1,1)$ of which the actual value of the components is

$$\sum_{i',j'} d^{ii'} d_{jj'} T^{j'}{}_{i'}. \qquad (2.16)$$

Returning to our toy example with the words "vase" and "wall", we can look at the change in vector representations as a change of basis $\hat{e}_i = \sum_{i'} \Lambda_i{}^{i'} \hat{e}'_{i'}$:

$$\vec{v} = \sum_i v^i \hat{e}_i = \sum_{ii'} v^i \Lambda_i{}^{i'} \hat{e}'_{i'} = \sum_{i'} v'^{i'} \hat{e}'_{i'}, \qquad (2.17)$$

corresponding to a change in the vector components $v'^{i'} = v^i \Lambda_i{}^{i'}$. The components of the metric change as

$$d'_{j''j'''} = \Lambda_{j''}{}^{j} d_{jj'} \Lambda^{j'}{}_{j'''}. \qquad (2.18)$$

With this change, we can show that inner product remains invariant under a basis change:

$$w'^{i'} v'_{i'} = w'^{i'} d_{i'j'} v'^{j'} = w'^{i'} \Lambda_{i'}{}^{i} d'_{ij} \Lambda^{j}{}_{j'} v'^{j'} = w^i d'_{ij} v^j = w^i v_i. \tag{2.19}$$

In this way, finding the right metric allows us to preserve a value that is constant in the face of context dependent representations. Assuming a metric that has the following matrix representation in the standard basis,

$$d = \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix}, \tag{2.20}$$

its application to the vector elements in eqs.2.2 gives a value of the inner product calculated in the new representation:

$$v'_{i'} w'^{i'} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1. \tag{2.21}$$

Since norms of the vector have to be calculated using the same notion of inner product,

$$\|\vec{v}\| = \sqrt{v^i g_{ij} v^j}, \tag{2.22}$$

we find exactly the cosine similarity calculated in eq.2.6. Note that this formalism allows us to deal with non-orthogonal basis, but does not require it: in fact, there is an implicit metric already when we compute the Euclidean inner product in eq.2.2, given by $d_{orth} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ in the standard basis, which is the one assumed when talking about an orthonormal basis.

Since these new tools allow us to preserve a quantity in the face of a change of representation, we can start reversing the question on similarity: given a certain human judgement on similarity, or another constant of interest, what is the metric that preserves it across different representations[c]? Once

c In case the quantity we wish to preserve is other than that of the Euclidean inner product in either representation, there is an option to expand the vector representation of our words by adding vector components that act as parameters, to ensure that the quantity is indeed

the vector spaces are endowed with specific metrics, the new inner product definitions permeate all higher-rank tensor contractions that are performed between higher and lower rank tensors, namely the ones that will be used in the interpretation of the Lambek rules,[d] and can further be extended to density matrices.

2.3.1 *Metric in Dirac Notation*

We want to lift our description to the realm of density matrices. We now show how the concept of a metric can also be introduced in that description, such that the previously described advantages carry over.

Dirac notation is the usual notation for vectors in the quantum mechanics literature. To make the bridge with the previous concepts from tensor calculus, we introduce it simply as a different way to represent the basis and dual basis of a vector space. Let us rename their elements as *kets* $|i\rangle \equiv \hat{e}_i$ and as *bras* $\langle j| \equiv \hat{e}^j$. The fact that the bases are dual to one another is expressed by the orthogonality condition $\langle j|i\rangle = \delta_{ij}$, which, if the vector basis elements are orthogonal to each other, is equivalent to applying the Euclidean metric to $|i\rangle$ and $|j\rangle$. Using Dirac notation, a vector and dual vector are represented as $v \equiv |v\rangle = \sum_i v^i |i\rangle$ and $v^d \equiv \langle u| = \sum_j v_j \langle j|$[e] If the

conserved. This would be similar to the role played by the time dimension in Einstein's relativity theory.

d Using this formalism, we can replace the unit and counit maps ϵ and η maps of the compact closed category **FdVect** by

$$\eta^l : \mathbb{R} \to V \otimes V^* :: 1 \mapsto \mathbb{1} \otimes d(\mathbb{1}, -)$$

$$\eta^r : \mathbb{R} \to V^* \otimes V :: 1 \mapsto d(-, \mathbb{1}) \otimes \mathbb{1}$$

$$\epsilon^l : V^* \otimes V \to \mathbb{R} :: d(-, v) \otimes u \mapsto d(u, v)$$

$$\epsilon^r : V \otimes V^* \to \mathbb{R} :: v \otimes d(u, -) \mapsto d(u, v).$$

e For orthonormal basis over the field of complex numbers, the covariant components are simply given by the complex conjugate of the contravariant ones, $v_i = \bar{v}^i$.

basis elements are not orthogonal, this mapping has to be done through a more involved metric. To express this, in this paper we introduce a modified Dirac notation over the field of real numbers, inspired by the one used in [47] for the treatment of quantum states related by a specific group structure[f]. The previous basis elements of V are written now as $|_i\rangle \equiv \hat{e}_i$ and the corresponding dual basis as $\langle^j| \equiv \hat{e}^j$, such that $\langle^j|_i\rangle = \delta_i^j$. In this basis, the metric is expanded as $d = \sum_{j,j'} d_{j'j} \langle^j| \otimes \left\langle^{j'}\right|$ while the inverse metric is expressed as $d^{-1} = \sum_{ii'} d^{i'i} |_i\rangle \otimes |_{i'}\rangle$. The elements of the metric and inverse metric are related by $\sum_i d_{j'i} d^{ii'} = \delta_{j'}^{i'}$. Applying the metric to a basis element of V, we get

$$\langle_i| \equiv d(-, |_i\rangle) = \sum_{jj'} d_{j'j} \left\langle^j\right| \otimes \left\langle^{j'}\middle|_i\right\rangle = \sum_j d_{ij} \left\langle^j\right| . \tag{2.23}$$

Acting with this on $|_{i'}\rangle$ to extract the value of the inner product, the following formulations are equivalent:

$$d(|_{i'}\rangle, |_i\rangle) = d(-, |_i\rangle)\, |_{i'}\rangle = \sum_j d_{ij} \left\langle^j\middle|_{i'}\right\rangle = \langle_i|_{i'}\rangle = d_{ii'}. \tag{2.24}$$

When the inverse metric is applied to $\langle^j|$ it gives

$$\left|^j\right\rangle \equiv d\left(-, \left\langle^j\right|\right) = \left\langle^j\right| \sum_{ii'} d^{i'i} |_i\rangle \otimes |_{i'}\rangle = \sum_{i'} d^{i'j} |_{i'}\rangle , \tag{2.25}$$

with a subsequent application on $\left\langle^{j'}\right|$ giving

$$d^{-1}\left(\left\langle^{j'}\right|, \left\langle^j\right|\right) = \left\langle^{j'}\right| d\left(-, \left\langle^j\right|\right) = \left\langle^{j'}\right| \sum_{i'} d^{i'j} |_{i'}\rangle = \left\langle^{j'}\middle|^j\right\rangle = d^{j'j}. \tag{2.26}$$

f This treatment can be extended to the field of complex numbers by considering that the metric has conjugate symmetry, $d_{ij} = \bar{d}_{ji}$ [120].

Consistently, we can calculate the value of the new bras and kets defined in eqs.2.23 and 2.25 applied to one another, showing that they too form a basis/dual basis pair:

$$\left\langle {}_{i}\middle|{}^{j}\right\rangle = \sum_{j'} d_{ij'} \left\langle {}^{j'}\middle| \sum_{i'} d^{i'j} \, |_{i'}\rangle = \sum_{i'j'} d_{ij'} d^{i'j} \left\langle {}^{j'}\middle|{}_{i'}\right\rangle = \sum_{j'} d_{ij'} d^{j'j} = \delta_i^j. \tag{2.27}$$

If the basis elements are orthogonal, the components of the metric and inverse metric coincide with the orthogonality condition.

2.4 DENSITY MATRICES: CAPTURING DIRECTIONALITY

The semantic spaces we envisage for the interpretation of the syntactic calculus are density matrices. A *density matrix* or density operator is used in quantum mechanics to describe systems for which the state is not completely known. For lexical semantics, it can be used to describe the meaning of a word by placing distributional information on its components. As standardly presented[g], density matrices that are defined on a tensor product space indicate no preference with respect to contraction from the left or from the right. Because we want to keep the distinction between left and right implications in the semantics, we set up the interpretation of composite spaces in such a way that they indicate which parts will and will not contract with other density matrices.

The *basic* building blocks of the interpretation are density matrix spaces $\tilde{V} \equiv V^* \otimes V$. For this composite space, we choose the basis formed by $|_i\rangle$ tensored with $\langle_{i'}|$, $\tilde{E} = \{|_i\rangle \langle_{i'}|\} = \{\tilde{E}_J\}$. Carrying over the notion of duality to the density matrix space, we define the dual density matrix space $\tilde{V}^* \equiv V^* \otimes V$. The dual basis in this space is the map that takes each basis element of $\tilde{V}$ and returns the appropriate orthogonality conditions. It is formed by $\langle^j|$ tensored with $\left|^{j'}\right\rangle$, ${}_d\tilde{E} = \left\{\left|^{j'}\right\rangle \langle^j|\right\} = \{\tilde{E}^J\}$, and is applied on the basis vectors of $\tilde{V}$ via the trace operation

g A background for the non-physics reader can be found in [102].

$$\tilde{E}^J\left(\tilde{E}_I\right) = \mathrm{Tr}\left(\left|_i\right\rangle\left\langle_{i'}\right|^{j'}\right\rangle\left\langle^j\right|\right) = \sum_l \left\langle^l\middle|_i\right\rangle\left\langle_{i'}\middle|^{j'}\right\rangle\left\langle^j\middle|_l\right\rangle$$
$$= \sum_{jj'}\left\langle^j\middle|_i\right\rangle\left\langle_{j'}\middle|^{i'}\right\rangle\delta^j_i\delta^{j'}_{i'} \equiv \delta^J_I. \tag{2.28}$$

Because density operators are hermitian, their matrices do not change under conjugate transposition, which extends to elements of the basis of the density matrix space. In this way, we can extend our notion of metric to the space of density matrices, where a new metric D emerges from d, expanded in the basis of V^* as

$$D = \sum_{J,J'} D_{JJ'}\tilde{E}^J \otimes \tilde{E}^{J'} \tag{2.29}$$
$$= \sum_{jj',j''j'''} d_{j''j'}d_{j'''j}\left|^{j'}\right\rangle\left\langle^j\right| \otimes \left|^{j'''}\right\rangle\left\langle^{j''}\right|. \tag{2.30}$$

We can see how both definitions are equivalent by their action on a density matrix tensor $T \equiv \sum_I T^I\tilde{E}_I \equiv \sum_{ii'} T^{ii'}\left|_i\right\rangle\left\langle_{i'}\right|$. Staying at the level of $\tilde{V}$ and $\tilde{V}^*$, we use eq.2.29 to obtain

$$D(-,T) = \sum_{I,J,J'} D_{JJ'}T^I\tilde{E}^J \otimes \tilde{E}^{J'}\left(\tilde{E}_I\right) = \sum_{I,J,J'} D_{JJ'}T^I\tilde{E}^J\delta^{J'}_I$$
$$= \sum_{J,J'} D_{JJ'}T^{J'}\tilde{E}^J \equiv \sum_J T_J\tilde{E}^J = \sum_{jj'} T_{j'j}\left|^{j'}\right\rangle\left\langle^j\right|, \tag{2.31}$$

where we redefine $T_J \equiv D_{JJ'}T^{J'}$, thus establishing covariance and contravariance of the tensor components defined over the density matrix space. Looking in its turn at the level of V and V^*, using eq.2.30, we see that both definitions are equivalent:

$$
\begin{aligned}
D(-,T) &= \sum_{ii',jj',j''j'''} T^{ii'} d_{j''j'} d_{j'''j} \left|j'\right\rangle \left\langle j\right| \otimes \mathrm{Tr}\left(\left|j'''\right\rangle \left\langle j''\right|_i \right\rangle \langle i'|\right) \\
&= \sum_{ii',jj',j''j'''} T^{ii'} d_{j''j'} d_{j'''j} \delta_i^{j''} \delta_{i'}^{j'''} \left|j'\right\rangle \left\langle j\right| \\
&= \sum_{ii'jj'} T^{ii'} d_{ij'} d_{i'j} \left|j'\right\rangle \left\langle j\right| \equiv \sum_{jj'} T_{jj'} \left|j'\right\rangle \left\langle j\right|, \qquad (2.32)
\end{aligned}
$$

where we rewrite $T_{jj'} \equiv T^{ii'} d_{ij'} d_{i'j}$[h].

From these basic building blocks, *composite* spaces are formed via the binary operation $\otimes$ (tensor product) and a unary operation $()^*$ (dual functor) that sends the elements of a density matrix basis to its dual basis, using the metric defined above. In the notation, we use $\tilde{A}$ for density matrix spaces (basic or compound), and ρ, or subscripted $\rho_x, \rho_y, \rho_z, \ldots \in \tilde{A}$ for elements of such spaces. The $()^*$ operation is involutive; it interacts with the tensor product as $(\tilde{A} \otimes \tilde{B})^* = \tilde{B}^* \otimes \tilde{A}^*$ and acts as identity on matrix multiplication.

Below in (†) is the general form of a density matrix defined on a single space in the standard basis, and (‡) in the dual basis:

$$
(\dagger) \quad \rho_x^{\tilde{A}} = \sum_{ii'} X^{ii'} \, |_i\rangle_{\tilde{A}} \langle_{i'}|, \qquad (\ddagger) \quad \rho_x^{\tilde{A}^*} = \sum_{jj'} X_{j'j} \left|j'\right\rangle_{\tilde{A}^*} \left\langle j\right|.
$$

Over the density matrix spaces, we can see these matrices as *tensors* as we defined them previously, with $X^I \equiv X^{ii'}$ the *contravariant* components and with $X_{J'} \equiv X_{j'j}$ the *covariant* components.

h Here we can compare our formalism to that of the compact closed category of completely positive maps **CPM(FdVect)** developed in [126]. The categorical treatment applies here at a higher level, however, as introducing the metric defines explicitly the canonical isomorphisms $V \cong V^*$ and $\tilde{V} \cong \tilde{V}^*$, which trickles down to knowing exactly how the symmetry of the tensor product acts on the components of a tensor: $\sigma_{V,V^*} : V^* \otimes V \to V \otimes V^* :: \sum_{ij} T_i{}^j \hat{e}^i \otimes \hat{e}_j \mapsto \sum_{ii',jj'} d^{ii'} d_{jj'} T^{j'}{}_{i'} \hat{e}_i \otimes \hat{e}^j$.

A density matrix of a composite space can be an element of the tensor product space between the standard space and the dual space either from the left or from the right:

$$\rho_y^{\tilde{A}\otimes\tilde{B}^*} = \sum_{ii',jj'} Y^{ii'}_{j'j} \left|{}_i^{\,j'}\right\rangle_{\tilde{A}\otimes\tilde{B}^*} \left\langle{}^{\,j}_{i'}\right|; \tag{2.33}$$

$$\rho_w^{\tilde{B}^*\otimes\tilde{A}} = \sum_{ii',jj'} W_{j'j}^{\,ii'} \left|{}^{j'}_{\,i}\right\rangle_{\tilde{B}^*\otimes\tilde{A}} \left\langle{}^{\,j}_{i'}\right|. \tag{2.34}$$

Although both tensors are of the form $(1,1)$, the last one is a tensor with components $Y^I_{\ J'}$, which relate with a true tensor form by $D^{II'}Y^{\ J}_{I'}D_{JJ'}$. Recursively, density matrices that live in higher-rank tensor product spaces can be constructed, taking a tensor product with the dual basis either from the left or from the right. Multiplication between two density matrices of a standard and a dual space follows the rules of tensor contraction:

$$\rho_y^{\tilde{A}^*}\cdot\rho_x^{\tilde{A}} = \sum_{jj'} Y_{j'j}\left|{}^{j'}\right\rangle_{\tilde{A}^*}\left\langle{}^{j}\right| \cdot \sum_{ii'} X^{ii'}\,|_i\rangle_{\tilde{A}}\langle_{i'}| = \sum_{i',jj'} Y_{j'j}X^{ji'}\left|{}^{j'}\right\rangle_{\tilde{A}}\langle_{i'}|. \tag{2.35}$$

$$\rho_x^{\tilde{A}}\cdot\rho_y^{\tilde{A}^*} = \sum_{ii'} X^{ii'}\,|_i\rangle_{\tilde{A}}\langle_{i'}| \cdot \sum_{jj'} Y_{j'j}\left|{}^{j'}\right\rangle_{\tilde{A}^*}\left\langle{}^{j}\right| = \sum_{i,jj'} X^{ij'}Y_{j'j}\,|_i\rangle_{\tilde{A}}\left\langle{}^{j}\right|, \tag{2.36}$$

respecting the directionality of composition. To achieve full contraction, the trace in the appropriate space is applied, corresponding to a partial trace if the tensors involve more spaces:

$$\mathrm{Tr}_{\tilde{A}}\left(\sum_{i',jj'} Y_{j'j}X^{ji'}\left|{}^{j'}\right\rangle_{\tilde{A}}\langle_{i'}|\right) = \sum_{l,i',jj'} Y_{j'j}X^{ji'}\,{}_{\tilde{A}}\left\langle{}_{l}\right|{}^{j'}\rangle_{\tilde{A}^*}\,{}_{\tilde{A}}\left\langle{}^{\,i'}\right|{}^{l}\rangle_{\tilde{A}^*} = \sum_{jj'} Y_{j'j}X^{jj'}, \tag{2.37}$$

$$\mathrm{Tr}_{\tilde{A}}\left(\sum_{i,jj'} X^{ij'}Y_{j'j}\left|i\right\rangle_{\tilde{A}}\left\langle j\right|\right) = \sum_{l,j',ij} X^{ij'}Y_{j'j}\,{}_{\tilde{A}^*}\!\left\langle l\middle|i\right\rangle_{\tilde{A}\;\tilde{A}^*}\!\left\langle j\middle|l\right\rangle_{\tilde{A}} = \sum_{jj'} X^{jj'}Y_{j'j}. \tag{2.38}$$

We see that the cyclic property of the trace is preserved.

In §2.6 we will be dealing with derivational ambiguity, and for that the concepts of *subsystem* and *permutation operation* introduced here will be useful. A subsystem can be thought of as a copy of a space, described using the same basis, but formally treated as a different space. In practice, this means that different subsystems do not interact with one another. In the quantum setting, they represent independent identical quantum systems. For example, when we want to describe the spin states of two electrons, despite the fact that each spin state is defined on the same basis, it is necessary to distinguish which electron is in which state and so each is attributed to their own subsystem. Starting from a space $\tilde{A}$, two different subsystems are referred to as $\tilde{A}_1$ and $\tilde{A}_2$. If different words are described in the same space, subsystems can be used to formally assign them to different spaces. The permutation operation extends naturally from the one in standard quantum mechanics. We define two permutation operators: $P^{\tilde{A}_1\tilde{A}_2}$ permutes the elements of the basis of the respective spaces, while $P_{\tilde{A}_1\tilde{A}_2}$ permutes the elements of the dual basis. If only one set of basis elements is inside the scope of the permutation operators, then either the subsystem assignment changes,

$$P^{\tilde{A}_1\tilde{A}_2}\,|i\rangle_{\tilde{A}_1}\langle i'|P^{\tilde{A}_1\tilde{A}_2} = |i\rangle_{\tilde{A}_2}\langle i'|; \qquad P_{\tilde{A}_1\tilde{A}_2}\left|i'\right\rangle_{\tilde{A}_1{}^*}\!\left\langle i\right|P_{\tilde{A}_1\tilde{A}_2} = \left|i'\right\rangle_{\tilde{A}_2{}^*}\!\left\langle i\right|; \tag{2.39}$$

or the respective space of tracing changes,

$$\mathrm{Tr}_{\tilde{A}_1}\left(P_{\tilde{A}_1\tilde{A}_2}\,|i'\rangle_{\tilde{A}_2{}^*}\langle i|P_{\tilde{A}_1\tilde{A}_2}\right) = \mathrm{Tr}_{\tilde{A}_2}\left(|i'\rangle_{\tilde{A}_2{}^*}\langle i|\right). \tag{2.40}$$

Note that permutations take precedence over traces. If two words are assigned to different subsystems, the permutations act to swap their space assignment[i]:

$$P^{\tilde{A}_1\tilde{A}_2} \left|i\right\rangle_{\tilde{A}_1} \left\langle i'\right| \otimes \left|j\right\rangle_{\tilde{A}_2} \left\langle j'\right| P^{\tilde{A}_1\tilde{A}_2} = \left|i\right\rangle_{\tilde{A}_2} \left\langle i'\right| \otimes \left|j\right\rangle_{\tilde{A}_1} \left\langle j'\right|, \tag{2.41}$$

$$P_{\tilde{A}_1\tilde{A}_2} \left|i'\right\rangle_{\tilde{A}_1{}^*} \left\langle i\right| \otimes \left|j'\right\rangle_{\tilde{A}_2{}^*} \left\langle j\right| P_{\tilde{A}_1\tilde{A}_2} = \left|i'\right\rangle_{\tilde{A}_2{}^*} \left\langle i\right| \otimes \left|j'\right\rangle_{\tilde{A}_1{}^*} \left\langle j\right|. \tag{2.42}$$

If no word has that subsystem assignment then the permutation has no effect.

2.5 INTERPRETING LAMBEK CALCULUS DERIVATIONS

Let us turn now to the syntax-semantics interface, which takes the form of a homomorphism sending the types and derivations of the syntactic front end **(N)L**$_{/,\backslash}$ to their semantic counterparts. Consider first the action of the interpretation homomorphism on *types*. We write $\lceil . \rceil$ for the map that sends syntactic types to the interpreting semantic spaces. For primitive types we set

$$\lceil s \rceil = \tilde{S},\ \lceil np \rceil = \lceil n \rceil = \tilde{N}, \tag{2.43}$$

with S the vector space for sentence meanings and N the space for nominal expressions (common nouns, full noun phrases). For compound types we have

$$\lceil A/B \rceil = \lceil A \rceil \otimes \lceil B \rceil^*, \text{ and } \lceil A\backslash B \rceil = \lceil A \rceil^* \otimes \lceil B \rceil. \tag{2.44}$$

i We define this as a shorthand application of the permutation operations as defined in eq.2.39, such that eq.2.41 can be calculated w.r.t. that definition as

$$P^{\tilde{A}_1\tilde{A}_2} \left|i\right\rangle_{\tilde{A}_1} \left(\left\langle i'\right|_{\tilde{A}_1} P^{\tilde{A}_1\tilde{A}_2}\right) \otimes \left(P^{\tilde{A}_1\tilde{A}_2} \left|j\right\rangle_{\tilde{A}_2}\right) \left\langle j'\right|_{\tilde{A}_2} P^{\tilde{A}_1\tilde{A}_2}$$

$$= P^{\tilde{A}_1\tilde{A}_2} \left|i\right\rangle_{\tilde{A}_1\ \tilde{A}_2} \left\langle i'\right| \otimes \left|j\right\rangle_{\tilde{A}_1\ \tilde{A}_2} \left\langle j'\right| P^{\tilde{A}_1\tilde{A}_2} = \left|i\right\rangle_{\tilde{A}_2} \left\langle i'\right| \otimes \left|j\right\rangle_{\tilde{A}_1} \left\langle j'\right|,$$

and similarly for eq.2.42.

Given semantic spaces for the syntactic types, we can turn to the interpretation of the syntactic *derivations*, as coded by their $\lambda_{/,\backslash}$ proof terms. We write $[\![\cdot]\!]_g$ for the map that associates each term t of type A with a semantic value, i.e. an element of $\lceil A \rceil$, the semantic space where meanings of type A live. The map $[\![.]\!]$ is defined relative to a assignment function g that provides a semantic value for the basic building blocks, viz. the variables that label the axiom leaves of a proof. As we saw above, a proof term is a generic meaning recipe that abstracts from particular lexical meanings. Specific lexical items, as we will see in §2.6, have the status of *constants*. These constants are mapped to their distributional meaning by an interpretation function I. The distributional meaning corresponds to the embeddings assigned by a particular model to the lexicon. Below we show that this calculus is sound with respect to the semantics of section 2.4.

AXIOM

$$\left[\!\left[x^A\right]\!\right]_g = g(x^A) = \rho_x^{\lceil A \rceil} = \sum_{ii'} X^{ii'} \, |_i\rangle_{\lceil A \rceil} \langle_{i'}| . \tag{2.45}$$

ELIMINATION Recall the inference rules of Fig 3.1.

$E_/$: Premises $t^{B/A}$, u^A; conclusion $(t \triangleleft u)^B$:

$$\left[\!\left[(t \triangleleft u)^B\right]\!\right]_g \equiv \mathrm{Tr}_{\lceil A \rceil}\left(\left[\!\left[t^{B/A}\right]\!\right]_g \cdot \left[\!\left[u^A\right]\!\right]_g\right) \tag{2.46}$$

$$= \mathrm{Tr}_{\lceil A \rceil}\left(\sum_{ii',jj'} T^{ii'}_{j'j} \left|\begin{matrix} j' \\ i \end{matrix}\right\rangle_{\lceil B \rceil \otimes \lceil A \rceil^*} \left\langle \begin{matrix} j \\ i' \end{matrix}\right| \cdot \sum_{kk'} U^{kk'} \, |_k\rangle_{\lceil A \rceil} \langle_{k'}|\right) \tag{2.47}$$

$$= \sum_{ii',jj'} \sum_{kk'} T^{ii'}_{j'j} \cdot U^{kk'} \, \delta^j_k \delta^{j'}_{k'} \, |_i\rangle_{\lceil B \rceil} \langle_{i'}| = \sum_{ii',jj'} T^{ii'}_{j'j} \cdot U^{jj'} \, |_i\rangle_{\lceil B \rceil} \langle_{i'}| . \tag{2.48}$$

$E_\backslash$: Premises u^A, $t^{A\backslash B}$; conclusion $(u \triangleright t)^B$:

$$\left[\!\left[(u \triangleright t)^B\right]\!\right]_g \equiv \mathrm{Tr}_{\lceil A \rceil}\left(\left[\!\left[u^A\right]\!\right]_g \cdot \left[\!\left[t^{A\backslash B}\right]\!\right]_g\right) \tag{2.49}$$

$$= \mathrm{Tr}_{\lceil A \rceil}\left(\sum_{kk'} U^{kk'} \left|_k\right\rangle_{\lceil A \rceil}\left\langle{}'_{k}\right| \cdot \sum_{ii',jj'} T_{jj}^{\ ii'} \left|{}^{j'}_{i}\right\rangle_{\lceil A \rceil^* \otimes \lceil B \rceil}\left\langle{}^{j}_{i'}\right|\right) = \tag{2.50}$$

$$= \sum_{kk'}\sum_{ii',jj'} U^{kk'} \cdot T_{j'j}^{\ \ ii'}\, \delta^j_k \delta^{j'}_{k'} \left|_i\right\rangle_{\lceil B \rceil}\left\langle_{i'}\right| = \sum_{ii',jj'} U^{jj'} \cdot T_{j'j}^{\ \ ii'} \left|_i\right\rangle_{\lceil B \rceil}\left\langle_{i'}\right|. \tag{2.51}$$

INTRODUCTION $I_/$: Premise t^B, with x^A as its rightmost parameter; conclusion $(\lambda^r x.t)^{B/A}$:

$$\left[\!\left[(\lambda^r x.t)^{B/A}\right]\!\right]_g \equiv \sum_{kk'}\left(\left[\!\left[t^B\right]\!\right]_{g^x_{kk'}} \otimes \left|_{k'}\right\rangle_{\lceil A \rceil^*}\left\langle_{k}\right|\right) \tag{2.52}$$

$I_\backslash$: Premise t^B, with x^A as its leftmost parameter; conclusion $(\lambda^l x.t)^{A\backslash B}$:

$$\left[\!\left[\left(\lambda^l x.t\right)^{A\backslash B}\right]\!\right]_g \equiv \sum_{kk'}\left(\left|_{k'}\right\rangle_{\lceil A \rceil^*}\left\langle_{k}\right| \otimes \left[\!\left[t^B\right]\!\right]_{g^x_{kk'}}\right) \tag{2.53}$$

Here $g^x_{kk'}$ is the assignment exactly like g except possibly for the parametric variable x which takes the value of the basis element $\left|_k\right\rangle_{\lceil A \rceil}\left\langle_{k'}\right|$. More generally, the interpretation of the introduction rules lives in a compound density matrix space representing a linear map from $\tilde{A}$ to $\tilde{B}$. The semantic value of that map, applied to any object $m \in \tilde{A}$, is given by $[\![t^B]\!]_{g'}$, where g' is the assignment exactly like g except possibly for the bound variable x^A, which is assigned the value m. Note that now, given the introduction of the metric, the interpretations of A/B and $B\backslash A$ are related by it: if the components of the first are $T_J^{\ I}$, then those of the second are given by those in eq.2.16 adapted for density matrices. This is what introduces directionality in our interpretation: using the metric, we can extract a

certain representation for a function word and distinguish by the values of the components whether it will contract from the left or from the right.

2.6 DERIVATIONAL AMBIGUITY

The density matrix construction has been successfully used to address lexical ambiguity [108], as well as lexical and sentence entailment [9, 119], where different measures of entropy are used to perform the disambiguation. Here we arrive at disambiguation in a different way, by storing in the diagonal elements of a higher order density matrix the different interpretations that result from the different contractions that the proof-as-programs prescribes. This is possible due to the the set-up that is formed by a multi-partite density matrices space, so that, by making use of permutation operations, it happens automatically that the two meanings are expressed independently. This is useful because it can be integrated with a lexical interpretation in density matrices optimized to other tasks, such as lexical ambiguity or entailment. It is also appropriate to treat the existence of these ambiguities in the context of incrementality, since it keeps the meanings separated in their interaction with posterior fragments.

We give a simple example of how the trace machinery can be used on an ambiguous fragment, providing a passage from one reading to the other at the interpretation level, and how the descriptions are kept separated. For this application, the coefficients in the interpretation of the words contain distributional information harvested from data, either from a count-base model or a more sophisticated language model. The final coefficient of each outcomes is the vector-based representation of that reading.

We illustrate the construction with the phrase "tall person from Spain". The lexicon below has the syntactic type assignments and the corresponding semantic spaces.

	syn type A	$\lceil A \rceil$
tall	n/n	$N^* \otimes N \otimes (N^* \otimes N)^*$
person	n	$N^* \otimes N$
from	$(n\backslash n)/np$	$(N^* \otimes N)^* \otimes N^* \otimes N \otimes (N^* \otimes N)^*$
Spain	np	$N^* \otimes N$

Given this lexicon, "tall person from Spain" has two derivations, corresponding to the bracketings "(tall person) from Spain" ($x/tall, y/person, w/from, z/Spain$):

$$\dfrac{\dfrac{\dfrac{}{x:n/n \vdash x:n/n}\,ax \quad \dfrac{}{y:n \vdash y:n}\,ax}{(x:n/n, y:n) \vdash (x \triangleleft y):n}\,/_{E_2} \quad \dfrac{\dfrac{}{w:(n\backslash n)/np \vdash w:(n\backslash n)/np}\,ax \quad \dfrac{}{z:np \vdash z:np}\,ax}{(w:(n\backslash n)/np, z:n) \vdash (w \triangleleft z):n\backslash n}\,/_{E_1}}{[(x:n/n, y:n), (w:(n\backslash n)/np, z:n)] \vdash ((x \triangleleft y) \triangleright (w \triangleleft z)):n}\,\backslash_{E_3}$$

versus "tall (person from Spain)":

$$\dfrac{\dfrac{}{x:n/n \vdash x:n/n}\,ax \quad \dfrac{\dfrac{}{y:n \vdash y:n}\,ax \quad \dfrac{\dfrac{}{w:(n\backslash n)/np \vdash w:(n\backslash n)/np}\,ax \quad \dfrac{}{z:np \vdash z:np}\,ax}{(w:(n\backslash n)/np, z:n) \vdash (w \triangleleft z):n\backslash n}\,/_{E_1}}{[y:n, (w:(n\backslash n)/np, z:n)] \vdash (y \triangleright (w \triangleleft z)):n}\,\backslash_{E_2}}{(x:n/n, [y:n, (w:(n\backslash n)/np, z:n)]) \vdash (x \triangleleft (y \triangleright (w \triangleleft z))):n}\,/_{E_3}$$

In the first reading, the adjective "tall" is evaluated with respect to all people, before it is specified that this person happens to be from Spain, whereas in the second reading the adjective "tall" is evaluated only in the restricted universe of people from Spain.

Taking "from Spain" as a unit for simplicity, let us start with the following primitive interpretations:

- $[\![tall^{n/n}]\!]_I = \sum_{ii',jj'} \mathbf{T}_{ii'}^{j'j} \left|\begin{smallmatrix} i \\ j' \end{smallmatrix}\right\rangle_{N\otimes N^*} \left\langle\begin{smallmatrix} i' \\ j \end{smallmatrix}\right|$,
- $[\![person^{n}]\!]_I = \sum_{kk'} \mathbf{P}_{kk'} \left|k\right\rangle_{N} \left\langle k'\right|$,
- $[\![from\_Spain^{n\backslash n}]\!]_I = \sum_{ll',mm'} \mathbf{F}^{l'l}_{\ mm'} \left|\begin{smallmatrix} m \\ l' \end{smallmatrix}\right\rangle_{N^*\otimes N} \left\langle\begin{smallmatrix} m' \\ l \end{smallmatrix}\right|$.

Interpreting each step of the derivation in the way described in the previous section will give two different outcomes. The first one is

$$
\begin{aligned}
&[\![tall\_person\_from\_Spain^n]\!]^1_I = \\
&= \mathrm{Tr}_N \left(\mathrm{Tr}_N \left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left| \begin{matrix} j' \\ i \end{matrix} \right\rangle_{N \otimes N^*} \left\langle \begin{matrix} j \\ i' \end{matrix} \right| \cdot \sum_{kk'} \mathbf{P}^{kk'} \, |k\rangle_N \langle k'| \right) \right. \\
&\quad \left. \cdot \sum_{ll',mm'} \mathbf{F}_{l'l}^{mm'} \left| \begin{matrix} l' \\ m \end{matrix} \right\rangle_{N^* \otimes N} \left\langle \begin{matrix} l \\ m' \end{matrix} \right| \right) \\
&= \sum_{ii',jj',mm'} \mathbf{T}^{ii'}_{j'j} \, \mathbf{P}^{jj'} \, \mathbf{F}_{i'i}^{mm'} \, |m\rangle_N \langle m'|, \qquad (2.54)
\end{aligned}
$$

while the second one is

$$
\begin{aligned}
&[\![tall\_person\_from\_Spain^n]\!]^2_I = \\
&= \mathrm{Tr}_N \left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left| \begin{matrix} j' \\ i \end{matrix} \right\rangle_{N \otimes N^*} \left\langle \begin{matrix} j \\ i' \end{matrix} \right| \cdot \mathrm{Tr}_N \left(\sum_{kk'} \mathbf{P}_{kk'} \, |k\rangle_N \langle k'| \right. \right. \\
&\quad \left. \left. \cdot \sum_{ll',mm'} \mathbf{F}_{l'l}^{mm'} \left| \begin{matrix} l' \\ m \end{matrix} \right\rangle_{N^* \otimes N} \left\langle \begin{matrix} l \\ m' \end{matrix} \right| \right) \right) \\
&= \sum_{ii',jj',ll'} \mathbf{T}^{ii'}_{j'j} \, \mathbf{P}^{ll'} \, \mathbf{F}_{l'l}^{jj'} \, |i\rangle_N \langle i'|. \qquad (2.55)
\end{aligned}
$$

The respective graphical representations of these contractions can be found in fig.2.2.

Though the coefficients might be different for each derivation, it is not clear how both interpretations are carried separately if they are part of a larger fragment, since their description takes place on the same space. Also, this recipe gives a fixed ordering and range for each trace. To be able to describe each final meaning separately, we use here the concept of *subsystem*. Because different subsystems act formally as different syntactic types and in each derivation the words that interact are different, it follows that each word should be assigned to a different subsystem:

- $[\![tall^{n/n}]\!]_{I_1} = [\![tall^{n/n}]\!]_{I_2} = \sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left| \begin{matrix} j' \\ i \end{matrix} \right\rangle_{N^1 \otimes N^{2*}} \left\langle \begin{matrix} j \\ i' \end{matrix} \right|$,

- $[\![person^n]\!]_{I_1} = \sum_{kk'} \mathbf{P}^{kk'} \, |k\rangle_{N^2} \langle k'|$,
  $[\![person^n]\!]_{I_2} = \sum_{kk'} \mathbf{P}^{kk'} \, |k\rangle_{N^3} \langle k'|$,
- $[\![from\_Spain^{n\backslash n}]\!]_{I_1} = \sum_{ll',mm'} \mathbf{F}_{l'l}^{\;\;mm'} \left|\begin{smallmatrix} l' \\ m \end{smallmatrix}\right\rangle_{N^{1*}\otimes N^3} \left\langle\begin{smallmatrix} l \\ m' \end{smallmatrix}\right|$,
  $[\![from\_Spain^{n\backslash n}]\!]_{I_2} = \sum_{ll',mm'} \mathbf{F}_{l'l}^{\;\;mm'} \left|\begin{smallmatrix} l' \\ m \end{smallmatrix}\right\rangle_{N^{3*}\otimes N^2} \left\langle\begin{smallmatrix} l \\ m' \end{smallmatrix}\right|$.

Notice that the value of the coefficients given by the interpretation functions I_1 and I_2 that describe the words does not change from the ones given in I, only possibly the subsystem assignment does.

Rewriting the derivation of the interpretations in terms of subsystems, the ordering of the traces does not matter anymore since the contraction is restricted to its own subsystem. For the first reading we obtain

$$\begin{aligned}
&[\![tall_person_from_Spain^n]\!]^1_{I_1} = \\
&= \mathrm{Tr}_{N^1}\left(\mathrm{Tr}_{N^2}\left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{\;\;j'j} \left|\begin{smallmatrix} j' \\ i \end{smallmatrix}\right\rangle_{N^1\otimes N^{2*}} \left\langle\begin{smallmatrix} j \\ i' \end{smallmatrix}\right| \cdot \sum_{kk'} \mathbf{P}^{kk'} \, |k\rangle_{N^2} \langle k'| \right.\right. \\
&\quad \left.\left. \cdot \sum_{ll',mm'} \mathbf{F}_{l'l}^{\;\;mm'} \left|\begin{smallmatrix} l' \\ m \end{smallmatrix}\right\rangle_{N^{1*}\otimes N^3} \left\langle\begin{smallmatrix} l \\ m' \end{smallmatrix}\right| \right)\right) \\
&= \sum_{ii',jj',mm'} \mathbf{T}^{ii'}_{\;\;j'j} \, \mathbf{P}^{jj'} \, \mathbf{F}_{i'i}^{\;\;mm'} \, |m\rangle_{N^3} \langle m'| \qquad (2.56)
\end{aligned}$$

and for the second

$$\begin{aligned}
&[\![tall_person_from_Spain^n]\!]^2_{I_2} = \\
&= \mathrm{Tr}_{N^2}\left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{\;\;j'j} \left|\begin{smallmatrix} j' \\ i \end{smallmatrix}\right\rangle_{N^1\otimes N^{2*}} \left\langle\begin{smallmatrix} j \\ i' \end{smallmatrix}\right| \cdot \mathrm{Tr}_{N^3}\left(\sum_{kk'} \mathbf{P}^{kk'} \, |k\rangle_{N^3} \langle k'| \right.\right. \\
&\quad \left.\left. \cdot \sum_{mm',ll'} \mathbf{F}_{l'l}^{\;\;mm'} \left|\begin{smallmatrix} l' \\ m \end{smallmatrix}\right\rangle_{N^{3*}\otimes N^2} \left\langle\begin{smallmatrix} l \\ m' \end{smallmatrix}\right| \right)\right)
\end{aligned}$$

$$
\begin{aligned}
&= \mathrm{Tr}_{N^3}\left(\mathrm{Tr}_{N^2}\left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left|\begin{smallmatrix} j' \\ i \end{smallmatrix}\right\rangle_{N^1\otimes N^{2*}} \left\langle\begin{smallmatrix} j \\ i' \end{smallmatrix}\right| \cdot \sum_{kk'} \mathbf{P}^{kk'} \left|k\right\rangle_{N^3} \left\langle k'\right| \right.\right. \\
&\quad \left.\left. \cdot \sum_{ll',mm'} \mathbf{F}_{l'l}^{mm'} \left|\begin{smallmatrix} l' \\ m \end{smallmatrix}\right\rangle_{N^{3*}\otimes N^2} \left\langle\begin{smallmatrix} l \\ m' \end{smallmatrix}\right| \right)\right) \\
&= \sum_{ii',jj',ll'} \mathbf{T}^{ii'}_{j'j}\, \mathbf{P}^{ll'}\, \mathbf{F}_{l'l}^{jj'} \left|i\right\rangle_{N^1} \left\langle i'\right|. \qquad (2.57)
\end{aligned}
$$

The interpretation of each derivation belongs now to different subsystems, which keeps the information about the original word to which the free "noun" space is attached. We can see this by comparing the upper and lower links in fig. 2.3.

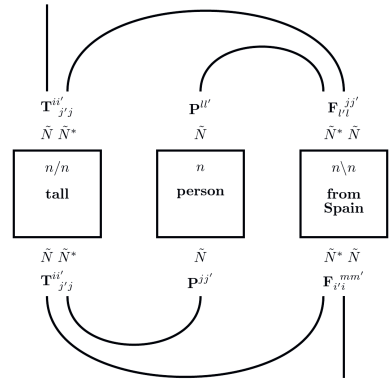


Figure 2.2: Representation of contractions corresponding to the first reading (lower links) and to the second reading (upper links), without subsystems. The final value is a coefficient in the $\tilde{N}$ space as in eq.2.54 and in eq.2.55, respectively.

However, it is not very convenient to attribute each word to a different subsystem depending on the interpretation it will be part of, since that is information that comes from the derivation itself and not from the representations of words. To tackle this problem, one uses permutation operations over the subsystems. Since these have precedence over the trace, when the traces are taken the contractions change accordingly. This changes

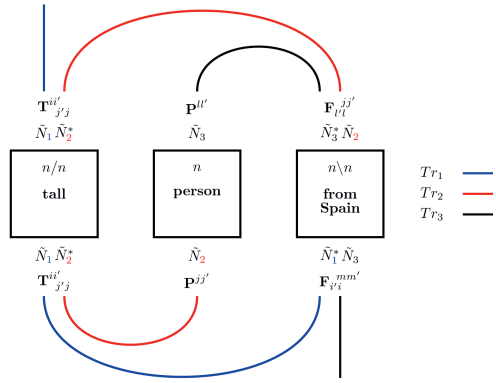


Figure 2.3: Representation of contractions corresponding to the first reading (lower links) and to the second reading (upper links), with subsystems. The final value is a coefficient in the $\tilde{N}$ space as in eq.2.56 and in eq.2.57, respectively.

the subsystem assignment at specific points so it is possible to go from one interpretation to the other, without giving different interpretations to each word initially. Thus, there is a way to go directly from the first interpretation to the second:

$$
\begin{aligned}
&[\![tall\_person\_from\_Spain^n]\!]^2_{I_1} = \\
&= \mathrm{Tr}_{N^1}\left(P_{13}\,\mathrm{Tr}_{N^2}\left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left|\begin{matrix} j' \\ i \end{matrix}\right\rangle_{N^1\otimes N^{2*}} \left\langle\begin{matrix} j \\ i' \end{matrix}\right| \cdot P_{13} P^{23} \sum_{kk'} \mathbf{P}^{kk'} \,|k\rangle_{N^2}\langle k'| \right.\right. \\
&\qquad \left.\left. \cdot \sum_{ll',mm'} \mathbf{F}^{mm'}_{l'l} \left|\begin{matrix} l' \\ m \end{matrix}\right\rangle_{N^{1*}\otimes N^3} \left\langle\begin{matrix} l \\ m' \end{matrix}\right| P^{23} P_{13} \right) P_{13} \right) \\
&= \mathrm{Tr}_{N^3}\left(\mathrm{Tr}_{N^2}\left(\sum_{ii',jj'} \mathbf{T}^{ii'}_{j'j} \left|\begin{matrix} j' \\ i \end{matrix}\right\rangle_{N^1\otimes N^{2*}} \left\langle\begin{matrix} j \\ i' \end{matrix}\right| \cdot \sum_{kk'} \mathbf{P}^{kk'} \,|k\rangle_{N^3}\langle k'| \right.\right. \\
&\qquad \left.\left. \cdot \sum_{ll',mm'} \mathbf{F}^{mm'}_{l'l} \left|\begin{matrix} l' \\ m \end{matrix}\right\rangle_{N^{3*}\otimes N^2} \left\langle\begin{matrix} l \\ m' \end{matrix}\right| \right) \right) .
\end{aligned}
\tag{2.58}
$$

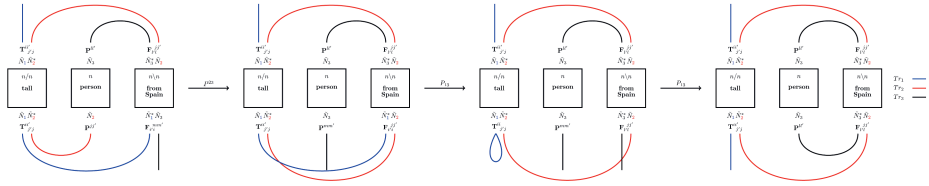


Figure 2.4

The reasoning behind is as follows: the permutation P^{23} swaps the space assignment between that of "person" and the free space in "from_Spain", according to eq.2.42; after that a permutation P_{13} is used as in eq. 2.39 to change the argument space of "from_Spain" from N^{1^*} to N^{3^*}, and then the same permutation is applied again to change the space of tracing, following eq.2.40. In this way, all the coefficients will have the correct contractions and in a different space from the first reading. In fig. 2.4 we can see the action of the permutations by visualizing how both the spaces and the traces change as we go from the lower to the upper links.

Although the metric is not used explicitly in the application of the permutation operators, it is necessary to generate the correct tensors where the permutation operator is applied in the first place, by going from the vector representation that comes directly from the data to one that allows contraction. As an example, the adjective "tall" would have a vector representation from the data as an element of $\tilde{V} \otimes \tilde{V}$, of the form $\mathbf{T}^{ii',kk'}$. We need the metric $d_{kj'}d_{k'j}$ to change its form to $T^{ii'}_{j'j}$. By defining the interpretation space of adjectives as $\tilde{N} \otimes \tilde{N}^*$, we assume this passage has already been made when we assign an interpretation to a word in this space. As an alternative to this derivation, we mention that it is possible to apply a P^{23} permutation followed by a P^{13} permutation that results in the correct contraction of the indices, but fails to deliver the results of the two derivations in different subspaces, as represented in Fig. 2.5. It is however noteworthy that, in order to start with a unique assignment for each word, this alternative derivation can, in any case, only be achieved by distinguishing between subsystems, as well as the covariant and contravariant indices.

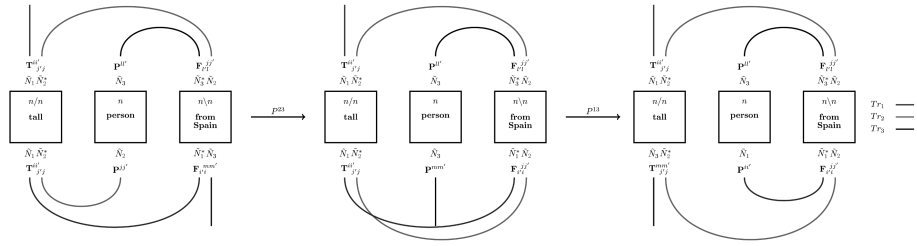


Figure 2.5

2.7 CONCLUSION AND FUTURE WORK

In this paper we provided a density matrix model for a simple fragment of the Lambek Calculus, differently from what is done in [18] who uses density matrices to interpret dependency parse trees. The syntax-semantics interface takes the form of a compositional map assigning semantic values to the $\lambda_{/,\backslash}$ terms coding syntactic derivations. We proposed the use of a metric as a way to reconcile the various vector representations of the same word that come from different treatments, assuming that there is a quantity that is being preserved, such as human judgements. If we know the metric, we can confidently assign only one embedding to each word as its semantic value. A metric can relate these various representations so that we can assign only one vector as its semantic value. The density matrix model enables the integration of lexical and derivational forms of ambiguity. Additionally, it allows for the transfer of methods and techniques from quantum mechanics and general relativity to computational semantics. One example of such transfer is the permutation operator. In quantum mechanics, this operator permits a description of indistinguishable particles. In the linguistic application, it allows one to go from an interpretation that comes from one derivation to another, without the need to to go through the latter, but keeping this second meaning in a different subsystem. Another example is the introduction of covariant and contravariant components, associated with a metric, that allow the permutation operations to be properly applied. In future work, we want to explore the preservation of human judgements found in the literature via a metric that represents the variability of vector

representations of words, either static or dynamic. We also want to extend our simple fragment with modalities for structural control (cf [95]), in order to deal with cases of derivational ambiguity that are licensed by these control modalities. Finally, we want to consider derivational ambiguity in the light of an *incremental* left-to-right interpretation process, so as to account for the evolution of interpretations over time. In enriching the treatment with a metric, we want to explore the consequences of having this new parameter in treating context dependent embeddings.

2.8 ACKNOWLEDGEMENTS

A.D.C. thanks discussions with Sanjaye Ramgoolam and Martha Lewis that contributed to the duality concepts included in the journal version of this paper. This work is supported by the Complex Systems Fund, with special thanks to Peter Koeze.

2.A PROOF TRANSFORMATION: β REDUCTION

The β-reduction is one of the rewrite rules of the λ-calculus. It asserts that applying a term with a lambda-bound variable to a certain argument is equivalent to substituting that argument directly in the original term, before introducing the lambda. In proof-theoretic terms, if an introduction rule is used followed by an elimination rule, the derivation is not minimal. To elucidate this point, below is the skeleton of a derivation where a term of type A is proved twice, by axiom and by an unknown proof:

$$\dfrac{\dfrac{\vdots}{\Delta \vdash n : A} \quad \dfrac{\dfrac{\dfrac{}{x : A \vdash x : A}\, axiom}{\vdots}}{\dfrac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda^l x.m : A \backslash B}\, \backslash_I}}{(\Gamma, \Delta) \vdash (\lambda^l x.m) n : B}\, \backslash_E .$$

The β reduction consists of substituting the unknown proof of the term of type A in place of the axiom, reducing the need for the double proof of that term, and consequently the size of the proof:

$$\dfrac{\dfrac{\vdots}{\Delta \vdash n : A}}{\dfrac{\vdots}{\Delta, \Gamma \vdash m[x/n] : B}} .$$

Through this reduction, a map from one conclusion to the other can be obtained, which has to be an equality regarding their interpretations:

$$[\![(\lambda x.m) n]\!]_g = [\![m[x/n]]\!]_g, \forall g.$$

This equality will be used to check that the density matrix construction interpretation is consistent with the λ-calculus. Below a concrete symbolic derivation before the reduction is shown:

$$\dfrac{\dfrac{\dfrac{}{w:B\vdash w:B}\,ax \quad \dfrac{}{z:B\backslash(A/B)\vdash z:B\backslash(A/B)}\,ax}{w:B,z:B\backslash(A/B)\vdash z(w):A/B}\backslash_{E_2} \quad \dfrac{\dfrac{\dfrac{}{x:A/B\vdash x:A/B}\,ax \quad \dfrac{}{y:B\vdash y:B}\,ax}{x:A/B,y:B\vdash x(y):A}/_{E_1}}{y:B\vdash \lambda^l x.x(y):(A/B)\backslash A}\backslash_{I_1}}{(w:B,z:B\backslash(A/B),u:B)\vdash (z(w))\lambda^l x.x(y):A}\backslash_{E_3}.$$

The interpretation of the several steps of the proof is then given, following the numbering in the proof:

$$[\![E_{/_1}(x^{A/B},y^B)]\!]_g = X^{ii}_{jj'}\, Y^{j'j}\,|_i\rangle_{\ulcorner A\urcorner}\langle_{i'}|,$$

$$[\![I_{\backslash_1}(x(y)^A,x^{A/B})]\!]_g = \left|{}^{i'}_{j}\right\rangle_{\ulcorner B\urcorner\otimes\ulcorner A\urcorner^*}\left\langle{}^{i}_{j'}\right| \otimes Y^{j'j}\,|_i\rangle_{\ulcorner A\urcorner}\langle_{i'}|,$$

$$[\![E_{\backslash_2}(w^B,z^{B\backslash(A/B)}]\!]_g = W^{ll'}\, Z^{m'm}_{l'l,nn'}\left|{}^{n'}_{m}\right\rangle_{\ulcorner A\urcorner\otimes\ulcorner B\urcorner^*}\langle{}^{n}_{m'}|,$$

$$[\![E_{\backslash_3}(z(w)^{A/B},(\lambda x.x(y))^{(A/B)\backslash B})]\!]_g = W^{ll'}\, Z^{i'i}_{l'l,jj'}\, Y^{j'j}\,|_i\rangle_{\ulcorner A\urcorner}\langle_{i'}|.$$

A similar treatment is done for the derivation after the reduction:

$$\dfrac{\dfrac{\dfrac{}{w:B\vdash w:B}\,ax \quad \dfrac{}{z:B\backslash(A/B)\vdash z:B\backslash(A/B)}\,ax}{w:B,z:B\backslash(A/B)\vdash z(w):A/B}\backslash_{E_2} \quad \dfrac{}{y:B\vdash y:B}\,ax}{w:B,z:B\backslash(A/B),u:B\vdash z(w)(y):A}\backslash_{E_4}.$$

The value for $[\![E_{\backslash_2}(w^B,z^{B\backslash(A/B))}]\!]$ is the same as before. For $[\![E_{\backslash_4}(z(w)^{A/B},y^B]\!]$:

$$[\![E_{\backslash_4}(z(w)^{A/B},y^B]\!] = W^{ll'}\, Z^{i'i}_{l'l,jj'}\, Y^{j'j}\,|_i\rangle_{\ulcorner A\urcorner}\langle_{i'}|.$$

Comparing the two derivations and interpretations, the conclusion is that

$$[\![E_{\setminus_4}(y, z(w))]\!] = [\![E_{\setminus_3}(z(w), \lambda x.x(y))]\!],$$

as expected.

3

PUTTING A SPIN ON LANGUAGE

ABSTRACT Extended versions of the Lambek Calculus currently used in computational linguistics rely on unary modalities to allow for the controlled application of structural rules affecting word order and phrase structure. These controlled structural operations give rise to derivational ambiguities that are missed by the original Lambek Calculus or its pregroup simplification. Proposals for compositional interpretation of extended Lambek Calculus in the compact closed category of FVect and linear maps have been made, but in these proposals the syntax-semantics mapping ignores the control modalities, effectively restricting their role to the syntax. Our aim is to turn the modalities into first-class citizens of the vectorial interpretation. Building on the directional density matrix semantics, we extend the interpretation of the type system with an extra spin density matrix space. The interpretation of proofs then results in ambiguous derivations being tensored with orthogonal spin states. Our method introduces a way of simultaneously representing co-existing interpretations of ambiguous utterances, and provides a uniform framework for the integration of lexical and derivational ambiguity.

3.1 INTRODUCTION

A cornerstone of formal semantics is Montague's [93] compositionality theory. Compositional interpretation, in this view, is a homomorphism, a structure-preserving map that sends types and derivations of a syntactic source logic to the corresponding semantic spaces and operations thereon. In the DisCoCat framework [33] compositionality takes a surprising new turn. Montague's abstract mathematical view on the syntax-semantics interface is kept, but the non-committed view on *lexical* meaning that one finds in formal semantics is replaced by a data-driven, distributional modelling, with finite dimensional vector spaces and linear maps as the target for the interpretation function. More recently density matrices and completely positive maps have been used to treat lexical ambiguity [109], word and sentence entailment [9, 119] and meaning updating [31].

Our goal in this paper is to apply the DisCoCat methodology to an extended version of the Lambek calculus where structural rules affecting word order and/or phrase structure are no longer freely available, but have to be explicitly licensed by unary control modalities [66, 94]. In particular, we adjust the interpretation homomorphism to assign appropriate semantic spaces to the modally extended type language, and show what their effect is on the derivational semantics. We choose to use density matrices as our interpretation spaces and show that, besides allowing for an integration of our model with other forms of ambiguity at the lexical level, it is key to preserve information about the ambiguity at phrase level.

The paper is structured as follows. In section 3.2 we recall the natural deduction rules of the simply typed Lambek Calculus, with the associated lambda terms under the proofs-as-programs interpretation. We extend the language with a residuated pair of unary modalities $\Diamond, \Box$ and show how these can be used to control structural reasoning, in particular reordering (commutativity). As an illustration, we show how the extended type logic allows us to capture derivational ambiguities that arise in Dutch relative clause constructions. In section 3.3 we set up the mapping from syntactic types to semantic spaces, adding an extra spin space to the previously used density matrix spaces. We motivate the introduction of this extra

space and relate the interpretation of the connectives in these spaces to the measurement and evolution postulates of quantum mechanics. In section 3.4 we show how the interpretation of the logical and structural inference rules of our extended type logic accommodates the spin space. In section 3.5 we make explicit the two-level spin space that we will use to store the ambiguity in the case of Dutch relative clauses. In section 3.6 we return to our example of derivational ambiguity and show how orthogonal spin states keep track of co-existing interpretations.

3.2 EXTENDED LAMBEK CALCULUS

By $\mathbf{NL}_\Diamond$ we designate the (non-associative, non-commutative, non-unital) pure residuation logic of [69], extended with a pair of unary type-forming operators $\Diamond, \Box$, also forming a residuated pair. Formulas are built over a set of atomic types $\mathcal{A}$ (here *s*, *np*, *n* for sentences, noun phrases and common nouns respectively) by means of a binary product $\bullet$ with its left and right residuals $/$, $\backslash$, and a unary $\Diamond$ with its residual $\Box$:

$$\mathcal{F} ::= \mathcal{A} \mid \Box\mathcal{F} \mid \Diamond\mathcal{F} \mid \mathcal{F}\backslash\mathcal{F} \mid \mathcal{F}/\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F}.$$

Figure 3.1 gives the (sequent-style) natural deduction presentation, together with the Curry-Howard term labelling[a]. Judgements are of the form $\Gamma \vdash B$, with B a formula and Γ a structure term with formulas at the leaves. Antecedent structures are built according to the grammar $\mathcal{S} ::= \mathcal{F} \mid (\mathcal{S} \cdot \mathcal{S}) \mid \langle \mathcal{S} \rangle$. The binary structure-building operation $(- \cdot -)$ is the structural counterpart of the connective $\bullet$ in the formula language. The unary structure-building operation $\langle - \rangle$ similarly is the counterpart of $\Diamond$ in the formula language.

With term labelling added, an antecedent term Γ with leaves $x_1 : A_1, \ldots, x_n : A_n$ becomes a typing environment giving type declarations for the variables x_i. These variables constitute the parameters for the program t associated with the proof of the succedent type B. Intuitively, one can see a term-

a We restrict to the simply typed fragment, ignoring the $\bullet$ operation.

labeled proof as an algorithm to compute a meaning t of type B with parameters x_i of type A_i. In parsing a particular phrase, one substitutes the meaning of the constants (i.e. words) that make up for the parameters of this algorithm.

Notice that the term language respects the distinction between / and \: we use the 'directional' lambda terms of [141] with left versus right abstraction and application. The inference rules for $\Box$ and $\Diamond$ are reflected in the term language by $^{\vee}, ^{\cup}$ (Elimination) and $^{\wedge}, ^{\cap}$ (Introduction) respectively.

$$\text{Terms:} \qquad t, u ::= x \mid \lambda^r x.t \mid \lambda^l x.t \mid t \triangleleft u \mid u \triangleright t \mid {}^{\cup} t \mid {}^{\cap} t \mid {}^{\vee} t \mid {}^{\wedge} t \mid {}^{c} t$$

Typing rules:

$$\frac{}{x : A \vdash x : A}\ Ax$$

$$\frac{\Gamma \cdot x : A \vdash t : B}{\Gamma \vdash \lambda^r x.t : B/A}\ I/ \qquad \frac{x : A \cdot \Gamma \vdash t : B}{\Gamma \vdash \lambda^l x.t : A\backslash B}\ I\backslash$$

$$\frac{\Gamma \vdash t : B/A \quad \Delta \vdash u : A}{\Gamma \cdot \Delta \vdash t \triangleleft u : B}\ E/ \qquad \frac{\Gamma \vdash u : A \quad \Delta \vdash t : A\backslash B}{\Gamma \cdot \Delta \vdash u \triangleright t : B}\ E\backslash$$

$$\frac{\langle \Gamma \rangle \vdash t : B}{\Gamma \vdash {}^{\wedge} t : \Box B}\ I\Box \qquad \frac{\Gamma \vdash t : B}{\langle \Gamma \rangle \vdash {}^{\cap} t : \Diamond B}\ I\Diamond$$

$$\frac{\Gamma \vdash t : \Box B}{\langle \Gamma \rangle \vdash {}^{\vee} t : B}\ E\Box \qquad \frac{\Delta \vdash t : \Diamond A \quad \Gamma[\langle x : A \rangle] \vdash u : B}{\Gamma[\Delta] \vdash u[{}^{\cup} t/x] : B}\ E\Diamond$$

Figure 3.1: NL$\Diamond$. Proofs and terms. Antecedent structure terms must be non-empty. Notation $\Gamma[\Delta]$ for structure term Γ with substructure Δ.

In addition to the logical rules for $\Diamond$ and $\Box$, we are interested in formulating options for structural reasoning keyed to their presence. Consider the postulates expressed by the categorical morphisms of (3.1), or the corresponding inference rules of (3.2) in the N.D. format of Figure 3.1. These represent controlled forms of associativity and commutativity, explicitly licensed by the presence of $\Diamond$ (or its structural counterpart $\langle - \rangle$ in the sequent rules).

$$\Diamond A \otimes (B \otimes C) \longrightarrow (\Diamond A \otimes B) \otimes C \quad \Diamond A \otimes (B \otimes C) \longrightarrow B \otimes (\Diamond A \otimes C) \tag{3.1}$$

$$\frac{\Gamma[(\langle \Delta_1 \rangle \cdot \Delta_2) \cdot \Delta_3] \vdash t : B}{\Gamma[\langle \Delta_1 \rangle \cdot (\Delta_2 \cdot \Delta_3)] \vdash t : B}\, Ass_\Diamond \quad \frac{\Gamma[\Delta_2 \cdot (\langle \Delta_1 \rangle \cdot \Delta_3)] \vdash t : B}{\Gamma[\langle \Delta_1 \rangle \cdot (\Delta_2 \cdot \Delta_3)] \vdash {}^c t : B}\, Comm_\Diamond \tag{3.2}$$

Controlled forms of structural reasoning of this type have been used to model the dependencies between question words or relative pronouns and 'gaps' (physically unrealized hypothetical resources) that follow them. We illustrate with Dutch relative clauses, and refer the reader to [97] for a vector-based semantic analysis. Dutch, like Japanese, has verb-final word order in embedded clauses as show in (3.3a) which translates as (3.3b). Now consider the relative clause (3.3c). It has two possible interpretations, expressed by the translations (3.3d) and (3.3e). With a typing $(n\backslash n)/(np\backslash s)$ for the relative pronoun 'die' we can capture only the (3.3d) interpretation; the improved typing $(n\backslash n)/(\Diamond\Box np\backslash s)$ creates a derivational ambiguity that covers both the (3.3d) and the (3.3e) interpretation, where the latter relies on the ability of the $\Diamond\Box np$ hypothesis to 'jump over' the subject by means of $Comm_\Diamond$.

a. (ik weet dat) Bob_{np} Alice_{np} $\text{bewondert}_{np\backslash(np\backslash s)}$
b. (I know that) Bob_{np} $\text{admires}_{(np\backslash s)/np}$ Alice_{np}
c. man_{n} $\text{die}_{??}$ de_hond_{np} $\text{bijt}_{np\backslash(np\backslash s)}$
d. man who bites the dog (= subject relativization)
e. man whom the dog bites (= object relativization)

(3.3)

The crucial subderivations for the (3.3c) example schematically rely on the following steps (working upward): $\backslash$ Introduction withdraws the $\Diamond\Box np$ hypothesis, $\Diamond$ Elimination followed by zero or more steps of structural reasoning bring the hypothesis to the position where it can actually be used as a 'regular' np, thanks to the $\Box$ Elimination proof of $\langle \Box np \rangle \vdash np$. The derived rule (*xleft*) in (3.4) telescopes this sequence of inference steps into a one-step inference, allowing for a succinct representation of the derivations.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z : \Box A \vdash z : \Box A}}{\langle z : \Box A\rangle \vdash {}^{\vee}z : A}\, E\Box}{\vdots}}{\Gamma[\langle z : \Box A\rangle \cdot \Delta] \vdash t : B}}{\vdots}}{x : \Diamond\Box A \vdash x : \Diamond\Box A \qquad \langle z : \Box A\rangle \cdot \Gamma[\Delta] \vdash {}^{c^n}t : B}\, (Ass_{\Diamond}, Comm_{\Diamond})^n \; E\Diamond}{\dfrac{x : \Diamond\Box A \cdot \Gamma[\Delta] \vdash {}^{c^n}t[{}^{\cup}x/z] : B}{\Gamma[\Delta] \vdash \lambda^l x.{}^{c^n}t[{}^{\cup}x/z] : \Diamond\Box A \backslash B}\, I\backslash} \qquad \dfrac{\dfrac{\dfrac{}{[y : A \vdash y : A]^n}}{\vdots}}{\Gamma[y : A \cdot \Delta] \vdash t : B}}{\Gamma[\Delta] \vdash \lambda^l x.{}^{c^n}t[{}^{\vee\cup}x/y] : \Diamond\Box A \backslash B}\, [xleft]^n \tag{3.4}$$

Here abbreviate the repeated application of the controlled commutativity rule on a single formula using the index n, where it serves a double purpose: indexing the hypothesis that will be extracted, and quantifying how many times the commutativity rule must be applied to licence this extraction. The proof term ${}^{c^n}t$ results from the nth application of this rule to the proof with conclusion term t, inductively defined with ${}^{c^0}t = t$ and ${}^{c^{n+1}}t = {}^{c}(c^n t)$.

Using our compiled inference rule, here are the derivations of both relativization readings, to be compared with those with the full uncompiled derivation in Appendix 3.A. On the proof of the subject relativization reading (3.3d), at the axioms, we show the constants (words) that will be substituted for the parameters of the proof term for the derivation. Also, in the structure terms on the left of the turnstile, we use these words instead of the parameter-type pairs to enhance legibility. This derivation uses the $\Diamond\Box np$ hypothesis as the subject of the relative clause body; it simply relies on $\Diamond$ and $\Box$ Elimination, and doesn't involve structural reasoning.

$$\dfrac{\dfrac{\text{man}}{y_0 : n}\,\ell \qquad \dfrac{\dfrac{\text{die}}{z_0 : (n\backslash n)/(\Diamond\Box np\backslash s)}\,\ell \qquad \dfrac{\dfrac{[\_ \vdash x : np]^0 \qquad \dfrac{\dfrac{\dfrac{\text{de}}{x_2 : np/n}\,\ell \qquad \dfrac{\text{hond}}{y_2 : n}\,\ell}{\text{de} \cdot \text{hond} \vdash (x_2 \lhd y_2) : np}\,[/E] \qquad \dfrac{\text{bijt}}{z_2 : np\backslash(np\backslash s)}\,\ell}{(\text{de} \cdot \text{hond}) \cdot \text{bijt} \vdash ((x_2 \lhd y_2) \rhd z_2) : np\backslash s}\,[\backslash E]}{\_ \cdot ((\text{de} \cdot \text{hond}) \cdot \text{bijt}) \vdash (x \rhd ((x_2 \lhd y_2) \rhd z_2)) : s}\,[\backslash E]}{(\text{de} \cdot \text{hond}) \cdot \text{bijt} \vdash \lambda^l x_1.{}^{c^0}({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2)) : \Diamond\Box np\backslash s}\,[xleft]^0}{\text{die} \cdot ((\text{de} \cdot \text{hond}) \cdot \text{bijt}) \vdash (z_0 \lhd \lambda^l x_1.{}^{c^0}({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2))) : n\backslash n}\,[/E]}{\text{man} \cdot (\text{die} \cdot ((\text{de} \cdot \text{hond}) \cdot \text{bijt})) \vdash (y_0 \rhd (z_0 \lhd \lambda^l x_1.{}^{c^0}({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2)))) : n}\,[\backslash E]$$

The index 0 in the rule *xleft* connects to the indexing of the hypothesis, reflecting that the hypothesis was already at the leftmost position. Therefore, no control rule is in need to be used. Contrast this with the derivation of the (3.3e) object relativization interpretation. In this case the $\Diamond\Box np$ hypothesis

is manoeuvred to the direct object position in the relative clause body thanks to the controlled commutativity option, used once as indicated by the index 1 in the *xleft* rule:

$$\dfrac{\dfrac{\dfrac{}{man}\,\ell}{y_0 : n} \quad \dfrac{\dfrac{}{die}\,\ell}{z_0 : (n\backslash n)/(\Diamond\Box np\backslash s)} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{de}\,\ell}{x_2 : np/n} \quad \dfrac{}{hond}\,\ell}{y_2 : n}\,[/E]}{de \cdot hond \vdash (x_2 \lhd y_2) : np} \quad \dfrac{\dfrac{[\_ \vdash x : np]^1 \quad \dfrac{}{bijt}\,\ell}{z_2 : np\backslash(np\backslash s)}\,[\backslash E]}{\_ \cdot bijt \vdash (x \rhd z_2) : np\backslash s}}{(de \cdot hond) \cdot (\_ \cdot bijt) \vdash (x_2 \lhd y_2) \rhd (x \rhd z_2) : s}\,[\backslash E]}{(de \cdot hond) \cdot bijt \vdash \lambda^l x_1.{}^{c^1}((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)) : \Diamond\Box np\backslash s}\,[xleft]^1}{die \cdot ((de \cdot hond) \cdot bijt) \vdash (z_0 \lhd \lambda^l x_1.{}^{c^1}((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2))) : n\backslash n}\,[/E]}{man \cdot (die \cdot ((de \cdot hond) \cdot bijt)) \vdash (y_0 \rhd (z_0 \lhd \lambda^l x_1.{}^{c^1}((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)))) : n}\,[\backslash E]$$

Our aim in the following sections is to provide a compositional interpretation of the control operators and the structural reasoning licensed by them that allows us to simultaneously represent the co-existing interpretations of ambiguous utterances such as (3.3c).

3.3 INTERPRETATION SPACES

Let us turn to the action of the interpretation homomorphism on the *types* of our extended Lambek calculus. In the approach introduced in [35], types are sent to density matrix spaces. These spaces are set up in a directionality-sensitive way, keeping in the semantics the distinction between left- or right-looking implications. Starting from the vector space V and its dual V^*, we use a modified Dirac notation to distinguish between two sets of basis of V, $\{|_{i'}\rangle\}$ and $\{|^{j}\rangle\}$, and two sets of basis of V^*, $\left\{\left\langle^{j'}\right|\right\}$ and $\{\langle_i|\}$, obeying the orthogonality conditions

$$\langle_i|_{i'}\rangle = d_{ii'}, \quad \left\langle^{j'}\middle|^{j}\right\rangle = d^{j'j}, \quad \left\langle_{i}\middle|^{j}\right\rangle = \delta_i^j, \quad \text{and} \quad \left\langle^{j}\middle|_{i}\right\rangle = \delta_i^j,$$

where a metric function d accounts for the eventual non-orthogonality between basis elements, and the Kronecker δ function defines the the relationship between dual basis elements. In general, the basis vector $\langle^{j'}|$ is obtained by the conjugate transposition of $|^{j}\rangle$. When the basis is not

orthogonal, this operation does not render the dual basis vector of $\left\langle j' \right|$ (which by definition is orthogonal to it and in our notation is represented by $|_{i'}\rangle$), but another vector $\left|^{j}\right\rangle$ that requires the metric tensor to describe this relationship. Compare this with the case with only one set of basis for each space, obtained in the standard way: $\left\langle j' \right|$ coincides with $|_{i'}\rangle$ so that all basis vectors are orthogonal to each other, and the metric is just δ

The *basic* building block for the interpretations is the density matrix space $\tilde{V} \equiv V \otimes V^*$. This space has density matrices as elements, which we will use as the starting representations of words, instead of vectors. Density matrices are 1) positive operators with 2) trace normalized to 1 [102]. In a physical system, this means that we can not only access the quantum properties of states, expressed as a linear combination of basis states of V or V^*, but we can also include the classical properties of a state, by constructing a basis of $\tilde{V}$ and describing the states as any linear combination formed with these basis elements that obeys conditions 1) and 2). Because the range of representations is enlarged, their use has been proposed for linguistic applications[9, 31, 109, 119], which we expand on here focusing on including the directionality of the calculus in this distributional representation. Defining the basis of V and V^* as we did before, we are able to construct a non-trivial basis for the density matrix space that carries over the structure of duality. For this space, we choose the basis formed by $|_i\rangle$ tensored with $\langle_{i'}|$, $\tilde{E} = \{|_i\rangle \langle_{i'}|\}$. We define the dual density matrix space $\tilde{V}^* \equiv V \otimes V^*$ and assign to the dual basis of this space the map that takes each basis element of $\tilde{V}$ and returns a scalar. That basis is formed by $\left\langle j \right|$ tensored with $\left|^{j'}\right\rangle$, $\tilde{D} = \left\{ \left|^{j'}\right\rangle \left\langle j \right| \right\}$, and is applied on the basis vectors of $\tilde{V}$ via the trace operation

$$\mathrm{Tr}\left(|_i\rangle \left\langle_{i'}\right|^{j'}\right\rangle \left\langle j \right|\right) = \sum_l \left\langle^{l}\right|_i\right\rangle \left\langle_{i'}\right|^{j'}\right\rangle \left\langle j \right|_l\right\rangle . \tag{3.5}$$

The *composite* spaces are formed via the binary operation $\otimes$ (tensor product) and the unary operation $()^*$ (dual functor) that sends the elements of a density matrix basis to its dual basis, using the metric tensor. In the notation, we use $\tilde{A}$ for density matrix spaces (basic or compound), and ρ, or

subscripted $\rho_x, \rho_y, \rho_z, \ldots \in \tilde{A}$ for elements of such spaces. The $()^*$ operation is involutive; it interacts with the tensor product as $(\tilde{A} \otimes \tilde{B})^* = \tilde{B}^* \otimes \tilde{A}^*$ and acts as identity on matrix multiplication.

The homomorphism that sends syntactic types to semantic spaces is the map $\lceil . \rceil$. For primitive types it acts as

$$\lceil s \rceil = \tilde{S} \quad \text{and} \quad \lceil np \rceil = \lceil n \rceil = \tilde{N},$$

with S the vector space for sentence meanings, N the space for nominal expressions (common nouns, full noun phrases). For compound types we have

$$\lceil A/B \rceil = \lceil A \rceil \otimes \lceil B \rceil^* \quad \text{and} \quad \lceil A \backslash B \rceil = \lceil A \rceil^* \otimes \lceil B \rceil.$$

This can be seen as an *operational* interpretation of formulae: a dualizing functor acting on one of the types, followed by a tensor product, also a functor, are identified with particular operations on elements, specifically by multiplying with the elements of a metric or by taking the trace [b].

3.3.1 *Translation of unary modalities*

We now turn to how to send the formulae decorated with unary modalities to semantic spaces, in a way that stays in this functorial/operational framework. Recall that in earlier work [96, 97] modally marked formulae are interpreted in the same space as their undecorated versions, i.e. $\lceil \Diamond A \rceil = \lceil \Box A \rceil = \lceil A \rceil$.

To build a non-trivial interpretation of the unary connectives, we expand the interpretation space using the description of quantum states, distinguishing between their *spatial* and *spin* degrees of freedom. Let the $\lceil . \rceil$ homomorphism give a description of the *spatial* components, encoding the numerically extracted distributional data. In addition to the spatial component, and commuting freely with the spatial parts, we introduce a new

b Equivalentely, in a categorical distributional framework this corresponds to establishing a basis and taking either tensor contraction or multiplication as the operations that represent the η and ϵ maps at the element level.

vector space, a density matrix space $\mathfrak{S}$, with dimension $(N+1) \times (N+1)$, where N the maximum value of index n in the *xleft* rule of eq.(3.4), where the *spin* components are encoded. We denote this by the *N-level spin space.* Here we do not distinguish between covariant and contravariant components, making the standard Dirac notation the appropriate one to deal with this space. Accordingly, the basis is orthonormal and has elements in $\{|a\rangle \langle a'|\}$, with the values of a and a' ranging from 0 to N.

To obtain the full translation from syntactic types to their distributional interpretation spaces, we introduce an extended interpretation homomorphism that tensors the $\lceil \cdot \rceil$ interpretation of *all* types with a density matrix space $\mathfrak{S}$ resulting in

$$\lfloor A \rfloor = \lceil A \rceil \otimes \mathfrak{S}. \tag{3.6}$$

For atoms and slash types, $\lceil \cdot \rceil$ stays as defined. For $\Diamond A$ and $\Box A$, we tensor $\lceil A \rceil$ with $\mathfrak{S} \otimes \mathfrak{S}^*$, the type for the matrix representation of the operators associated with $\Diamond$ and $\Box$, that is,

$$\lceil \Diamond A \rceil = \lceil \Box A \rceil = \lceil A \rceil \otimes \mathfrak{S} \otimes \mathfrak{S}^*. \tag{3.7}$$

The key idea here is that by tensoring every type with an extra spin space via $\lfloor \cdot \rfloor$, the marked types have representations that encode maps from $\mathfrak{S}$ to $\mathfrak{S}$ coming from $\lceil \cdot \rceil$. This justifies the use of the same spin space to interpret the two markers, as they act as endomorphisms on the $\mathfrak{S}$ space coming from $\lceil \cdot \rceil$, as in for lozenge $\lfloor \Diamond A \rfloor = \lceil \Diamond A \rceil \otimes \mathfrak{S}$ and similarly for box. At the *type* level, then, we find the structure to accommodate the operators $T_\Diamond, T_\Box \in \mathcal{L}(\mathfrak{S})$, for which the concrete distinct interpretations will be provided at the *term* level. The key point of this structure is to give us precise control over the spin space as we interpret the unary modalities. Note that our connectives' interpretations do not interfere either with the distributional data that is stored in the spacial spaces, which is compatible with the interpretation of these connectives in previous work [96, 97]. The interpretations we assign to the unary connectives consist of operations that only modify elements of an ancillary space. By enlarging the distributional

space with this new spin space, we can effectivelly find a distributional meaning for the unary connectives.

As an example, here is the $\lfloor\cdot\rfloor$ mapping for the relative pronoun type of (3.3c).

$$\begin{aligned}\lfloor (n\backslash n)/(\Diamond\Box np\backslash s)\rfloor &= \lceil (n\backslash n)/(\Diamond\Box np\backslash s)\rceil \otimes \mathfrak{S} \\ &= \lceil n\rceil^* \otimes \lceil n\rceil \otimes \lceil s\rceil^* \otimes \lceil np\rceil \otimes \underbrace{(\mathfrak{S}\otimes\mathfrak{S}^*)}_{T_\Diamond} \otimes \underbrace{(\mathfrak{S}\otimes\mathfrak{S}^*)}_{T_\Box} \otimes \mathfrak{S}\end{aligned} \tag{3.8}$$

3.4 OPERATIONAL INTERPRETATION OF LAMBEK RULES

Given the new semantic spaces for the syntactic types, we now turn to the interpretation of the syntactic *derivations*, as encoded by their lambda proof terms, proving the soundness of the calculus presented in section 3.2 with respect to the semantics of section 3.3. In spin space, the operations that interpret different syntactic maps relate with the quantum postulates describing measurement and evolution of quantum systems[102].

QUANTUM MEASUREMENT: Quantum measurements are described by a collection M_a of measurement operators, acting on the state space of the system being measured. The index a refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement then the probability that result a occurs is given by $p(a) = \mathrm{Tr}\left(M_a^\dagger M_a \rho\right)$ and the state of the system after the measurement is

$$\rho_a := \frac{M_a \rho M_a^\dagger}{p(a)}. \tag{3.9}$$

The measurement operators satisfy the completeness equation, $\sum_a M_a^\dagger M_a = I$. For an observable M with eigenvalues m and eigenvectors $|a\rangle$, a *projective* measurement is defined with $M_a = |a\rangle\langle a|$; in this context we say that a state has been projected onto $|a\rangle\langle a|$, and the quantum operator is then called a *projector*.

EVOLUTION The evolution of a closed quantum system is described by a unitary transformation. That is, the state ρ^i of the system at time t_i is related to state ρ^{i+1} of the system at time t_{i+1} by a unitary operator U which depends only on these times. The state ρ^{i+1} relates with the previous one ρ^i by $\rho^{i+1} = U\rho^i U^\dagger$.

This correspondence is established via a function $[\![\cdot]\!]_g$ that associates each term t of type A with a semantic value, i.e. an element of $\lceil A \rceil$, the semantic space where meanings of type A live. For proof terms, $[\![.]\!]$ is defined relative to an assignment function g, that provides a semantic value for the basic building blocks, viz. the variables that label the axiom leaves of a proof, in this case independently for the spatial (S) and spin ($\mathfrak{S}$) components. A particular assignment $g^S_{x,kk'}$ is used to interpret the lambda abstraction in the spatial spaces:

Definition 3.4.1. Given a variable x of type A, we write $g^S_{x,kk'}$ for the assignment exactly like g^S except for the variable x, which takes the value of the basis element of the interpreting space $|_k\rangle_{\lceil A \rceil}\langle_{k'}|$.

The elements of the spin space are given by

$$\rho^{\mathfrak{S}}_x = \sum_{a,a'=0}^{n-1} {}^{\mathfrak{S}}\mathbf{X}_{aa'} \, |a\rangle_{\mathfrak{S}} \langle a'| . \tag{3.10}$$

A pair of special assignment functions $g^{\mathfrak{S}}_{x,I}$ and $g^{\mathfrak{S}}_{x,y}$ is used to interpret the lambda abstraction in the spin space:

Definition 3.4.2. Given a variable x of type A, we write $g^{\mathfrak{S}}_{x,I}$ for the assignment exactly like $g^{\mathfrak{S}}$ except for the variable x, which takes the value of the normalized identity, $I = \sum_a \frac{1}{\dim \mathfrak{S}} \, |a\rangle_{\mathfrak{S}} \langle a|$.

Definition 3.4.3. Given a variable x of type A, we write $g^{\mathfrak{S}}_{x,y}$ for the assignment exactly like $g^{\mathfrak{S}}$ except for the variable x, which takes the value of variable y, also of type A.

The spatial interpretation of terms of types formed with binary connectives is as given in [35]. We reproduce here the main results, but focus on their interpretation in spin space. Further, we introduce the interpretation of the rules that introduce and eliminate unary connectives.

Some elimination rules will be interpreted in spin space using an instance of a projective measurement. Given a term u of type A and another term t of type B, we define a map $[\![t^A]\!]_{g^{\mathfrak{S}}} * [\![u^B]\!]_{g^{\mathfrak{S}}} : \mathfrak{S} \times \mathfrak{S} \to \mathfrak{S}$ acting on the interpretation of the terms in spin space:

$$[\![t^A]\!]_{g^{\mathfrak{S}}} * [\![u^B]\!]_{g^{\mathfrak{S}}} = \frac{\left([\![u^B]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![t^A]\!]_{g^{\mathfrak{S}}} \cdot \left([\![u^B]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left([\![u^B]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![t^A]\!]_{g^{\mathfrak{S}}} \cdot \left([\![u^B]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)}, \tag{3.11}$$

with $(.)^{\frac{1}{2}}$ such that when applied on an operator R we have that $(R)^{\frac{1}{2}} \cdot (R)^{\frac{1}{2}} = R$. Positive operators, such as density matrices, have a unique positive square root [7]. Physically, the spin split in its square-root acts as a measurement operator on the other input spin. Using normalization, the outcome is a well defined spin state. An unnormalized version of this operator is defined in [c]. An unnormalized version of this map is defined as the "phaser" in *Coecke and Meichanetzidis*[31].

3.4.1 *Axiom*

The axiom will be given by an element of the spatial spaces, tensored with an element of the spin space.

$$\left[\!\left[x^A\right]\!\right]_g = g(x^A) = \rho_x^{\lfloor A \rfloor} = \left[\!\left[x^A\right]\!\right]_{g^S} \otimes \left[\!\left[x^A\right]\!\right]_{g^{\mathfrak{S}}}, \tag{3.12}$$

where

c This a generalization of one of the Frobenius algebras already used in [9] in the category **CPM(FHilb)**, where, given the full density matrix representations of sentence, noun and verb, respectively $\rho(s)$, $\rho(n)$ and ρ, they relate by $\rho(s) = \rho(n)^{\frac{1}{2}}\rho(v)\rho(n)^{\frac{1}{2}}$.

$$\left[\!\left[x^A\right]\!\right]_{g^{\mathfrak{S}}} = \sum_{aa'} {}^{\mathfrak{S}}\mathbf{X}_{aa'} \left|a\right\rangle_{\mathfrak{S}} \left\langle a'\right| \quad \text{and} \quad \left[\!\left[x^A\right]\!\right]_{g^S} = \sum_{ii'} {}^{S}\mathbf{X}^{ii'} \left|_i\right\rangle_{\lceil A \rceil} \left\langle_{i'}\right|. \tag{3.13}$$

3.4.2 *Introduction and elimination of binary connectives*

ELIMINATION OF / AND \

$$\left[\!\left[(t \triangleleft u)^B\right]\!\right]_g \equiv \mathrm{Tr}_{\lceil A \rceil}\left(\left[\!\left[t^{B/A}\right]\!\right]_{g^S} \cdot \left[\!\left[u^A\right]\!\right]_{g^S}\right) \otimes [\![t^{B/A}]\!]_{g^{\mathfrak{S}}} * [\![u^A]\!]_{g^{\mathfrak{S}}}. \tag{3.14}$$

$$\left[\!\left[(u \triangleright t)^B\right]\!\right]_g \equiv \mathrm{Tr}_{\lceil A \rceil}\left(\left[\!\left[u^A\right]\!\right]_{g^S} \cdot \left[\!\left[t^{A\backslash B}\right]\!\right]_{g^S}\right) \otimes [\![t^{A\backslash B}]\!]_{g^{\mathfrak{S}}} * [\![u^A]\!]_{g^{\mathfrak{S}}}. \tag{3.15}$$

INTRODUCTION OF / AND \

$$\left[\!\left[(\lambda^r x.t)^{B/A}\right]\!\right]_g \equiv \sum_{kk'}\left([\![t^B]\!]_{g^S_{x,kk'}} \otimes \left|k'\right\rangle_{\lceil A \rceil^*}\left\langle k\right|\right) \otimes \left[\!\left[t^B\right]\!\right]_{g^{\mathfrak{S}}_{x,I}}. \tag{3.16}$$

$$\left[\!\left[\left(\lambda^l x.t\right)^{A\backslash B}\right]\!\right]_g \equiv \sum_{kk'}\left(\left|k'\right\rangle_{\lceil A \rceil^*}\left\langle k\right| \otimes [\![t^B]\!]_{g^S_{x,kk'}}\right) \otimes \left[\!\left[t^B\right]\!\right]_{g^{\mathfrak{S}}_{x,I}}. \tag{3.17}$$

Syntactic equalities like beta reduction are interpreted as equalities in this model, as is shown in appendix 3.D.

3.4.3 *Introduction and elimination of unary connectives*

As seen earlier in the example of eq.(3.8), at the term level the diamond introduction is interpreted by the map $T_\Diamond$ and box introduction is interpreted

by the map $T_{\Box}$, both consisting of maps $\mathfrak{S} \to \mathfrak{S}$. Two more operations need to be introduced, namely those that eliminate box, $T'_{\Box}$, and that eliminate diamond $T'\Diamond$. Since these are the maps applied in our proof, we next give their explicit form.

The operation $T'_{\Box}$ acting on elements of $\mathfrak{S}$ is the linear combination of projectors $T'^{a}_{\Box}$ onto pure states used as projectors $M_a = |a\rangle_{\mathfrak{S}}\langle a|$, generated by the eigenstates of an observable with $N+1$ different eigenvalues, specified for a particular unary modality, indexed by $a \in \{0, ..., N\}$. Applied on a state $\rho_x^{\mathfrak{S}}$, the general result is the following mixed state

$$T'_{\Box}(\rho_x^{\mathfrak{S}}) = \sum_{a=0}^{N} c_a T'^{a}_{\Box}(\rho_x^{\mathfrak{S}}) \equiv \sum_{a=0}^{N} c_a (\rho_x^{\mathfrak{S}} * |a\rangle_{\mathfrak{S}}\langle a|) = \sum_{a=0}^{N} c_a \left(\frac{M_a \rho_x^{\mathfrak{S}} M_a}{\mathrm{Tr}\,(M_a \rho_x^{\mathfrak{S}} M_a)} \right), \tag{3.18}$$

with $\sum_{a=0}^{N} c_a = 1$, $c_a \in R$. Defining the ordering of the eigenstates by the increasing value of their corresponding index a, rule $E_{\Box}$ will be interpreted in the spin components as the projection onto the lowest eigenstate, effectively with $c_0 = 1$ and $c_{a\neq 0} = 0$.

The operation $T'_{\Diamond}$ acts on elements by performing a unitary transformation, generated by the successive application of matrices $U_0 = \mathbb{1}$ and $U_b \in SU(N+1)$ on density matrices, for $b \in \{1, ..., N^2 + 2N\}$, represented as $T'^{b}_{\Diamond}$, for a particular representation and ordering. Again applied to the state $\rho_x^{\mathfrak{S}}$, the application of this operation is

$$\left(T'^{b}_{\Diamond}\left(\rho_x^{\mathfrak{S}}\right)\right)^{d_b} = \begin{cases} \rho_x^{\mathfrak{S}} & \text{if } d_b = 0 \\ U_b \rho_x^{\mathfrak{S}} U_b^{\dagger} & \text{if } d_b = 1 \end{cases} \tag{3.19}$$

$$T'_{\Diamond}(\rho_x^{\mathfrak{S}}) = \left(T'^{N^2+2N}_{\Diamond}\left(T'^{N^2+2N-1}_{\Diamond}\left(\ldots \left(T'^{0}_{\Diamond}\left(\rho_x^{\mathfrak{S}}\right)\right)^{d_0}\right)\right)^{d_{N^2+2N-1}}\right)^{d_{N^2+2N}} \tag{3.20}$$

where $()^\dagger$ indicates hermitian conjugation and $d_b \in \{0,1\}$ [d]. The rule $E_\Diamond$ is thus interpreted as performing a unitary transformation, using that $d_0 = 1$ and $d_{b\neq 0} 0$.

In the particular case where we interpret the introduction of a connective with the same operation of its connective, that is $T_\Box = T'_\Diamond$ and $T_\Diamond = T'_\Box$, the adjoint properties of the unary connectives are preserved. The implications $\Diamond\Box A \to A \to \Box\Diamond A$ are interpreted on space $\mathfrak{S}$ as

$$T_\Diamond\left(T_\Box\left(\mathfrak{S}\right)\right) \in \mathfrak{S} \in T_\Box\left(T_\Diamond\left(\mathfrak{S}\right)\right).$$

In the first inclusion we have a unitary transformation followed by a projection, which is inside the interpretation space of the state, the entire Bloch sphere. For second inclusion, any state inside of the Bloch sphere is inside the scope of projections followed by a unitary transformation. This is a consequence of the non-commutativity of the operations that interpret these connectives, measurement and evolution.

ELIMINATION OF $\Box$: $\quad \left[\!\left[(^\vee t)^B\right]\!\right]_g = \left[\!\left[t^{\Box B}\right]\!\right]_{g^S} \otimes T'^0_\Box\left(\left[\!\left[t^{\Box B}\right]\!\right]_{g^{\mathfrak{S}}}\right)$

ELIMINATION OF $\Diamond$:

$$\left[\!\left[{}^\cup t\right]\!\right]_{g^{\mathfrak{S}}} = T'^0_\Diamond\left(\left[\!\left[t^{\Diamond A}\right]\!\right]_{g^{\mathfrak{S}}}\right) \tag{3.21}$$

$$\left[\!\left[(u[{}^\cup t/x])^B\right]\!\right]_g = \mathrm{Tr}_{\lceil A\rceil}\left(\left[\!\left[t^{\Diamond A}\right]\!\right]_{g^S} \cdot \sum_{kk'} \left|k'\right\rangle_{\lceil A\rceil^*}\left\langle k\right| \otimes \left[\!\left[u^B\right]\!\right]_{g^S_{x,kk'}}\right) \otimes \left[\!\left[u^B\right]\!\right]_{g^{\mathfrak{S}}_{x,\cup_t}}. \tag{3.22}$$

INTRODUCTION OF $\Box$ AND $\Diamond$:

d Eq. (3.20) can possibly be extended with permutations over the order of application of $T'^b_\Diamond$.

$$\left[\!\left[(^{\wedge}t)^{\Box B}\right]\!\right]_g = \left[\!\left[t^B\right]\!\right]_{g^S} \otimes T^0_{\Box}\left(\left[\!\left[t^B\right]\!\right]_{g^{\mathfrak{S}}}\right), \quad \left[\!\left[(^{\cap}t)^{\Diamond B}\right]\!\right]_g = \left[\!\left[t^B\right]\!\right]_{g^S} \otimes T^0_{\Diamond}\left(\left[\!\left[t^B\right]\!\right]_{g^{\mathfrak{S}}}\right) \tag{3.23}$$

3.4.4 *Structural Reasoning*

To interpret the derived infererence rule, a *raising operator* S_+ acts on the input state and is applied as many times as nodes that need to be jumped to be in the right position to be extracted. We record that information by an index n on the substitution brackets of the proof term encoding the (*xleft*) inference. The index acts as a power on the raising operator, $(S_+)^n$, changing a state $\rho_a = |a\rangle_{\mathfrak{S}}\langle a|$ to $\rho_{a+n} = |a+n\rangle_{\mathfrak{S}}\langle a+n|$, where we use the convention that a matrix to the zeroth power is the identity matrix. Note that this is not a unitary operator, which means that the resulting state must be normalized after the application. Additionally the derived inference rule is interpreted using the previously given interpretations of $\Box$ and $\Diamond$.

DERIVED INFERENCE RULE

[*xleft*]^n: Premise t^B with subterm y^A at location n;
conclusion $(\lambda^l x.\left({}^{c^n}t\right)^B\left[{}^{\vee\cup}x/y\right]^n)^{\Diamond\Box A\backslash B}$:

$$\left[\!\left[\left({}^{\vee\cup}x\right)^A\right]\!\right]_{g^{\mathfrak{S}}} = T^0_{\Diamond}\left(T^0_{\Box}\left(\left[\!\left[x^{\Diamond\Box A}\right]\!\right]_{g^{\mathfrak{S}}}\right)\right) \tag{3.24}$$

$$
\begin{aligned}
&[\![(\lambda^l x.\left(c^n t\right)^B [^{\vee\cup}x/y])^{\Diamond\Box A\backslash B}]\!]_g = \\
&= \sum_{ll'} \left|l'\right\rangle_{\lceil A\rceil^*} \left\langle l\right| \otimes \left[\mathrm{Tr}_{\lceil A\rceil}\left(\left[\!\left[x^{\Diamond\Box A}\right]\!\right]_{g^S} \cdot \sum_{kk'} \left|k'\right\rangle_{\lceil A\rceil^*} \left\langle k\right| \otimes [\![t^B]\!]_{g^S_{y,kk'}}\right)\right]_{g^S_{x,ll'}} \\
&\otimes \frac{\left[(S_+)^n [\![t^B]\!]_{g^{\mathfrak{S}}_{y,^{\vee\cup}x}} \left((S_+)^\dagger\right)^n\right]_{g^{\mathfrak{S}}_{x,I}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left[(S_+)^n [\![t^B]\!]_{g^{\mathfrak{S}}_{y,^{\vee\cup}x}} \left((S_+)^\dagger\right)^n\right]_{g^{\mathfrak{S}}_{x,I}}\right)}
\end{aligned}
\tag{3.25}
$$

Here we can see clearly the physical meaning that the quantum interpretation gives to the application of the modal operators. In eq.(3.24), the combination of application of $T'^0_\Diamond$ and $T'^0_\Box$, interpreted as a projection and a unitary operation, respectively, takes the form of one of the possible outcomes of the quantum process $E = PU$ [102], applied on the state $\left[\!\left[x^{\Diamond\Box A}\right]\!\right]_{g^{\mathfrak{S}}}$, namely the one where the final state is $\langle 0|_{\mathfrak{S}}|0\rangle$. Having the unary connectives interpreted with the non-commutative operations of projection and unitary transformation correctly preserves the order of application of the connectives imposed at the syntactic level. The derivation of this interpretation from the extended version of *xleft* rule is explored in Appendix 3.B.

3.5 TWO-LEVEL SPIN SPACE

The structural ambiguity at hand will be treated using a two-level spin space, since we have two ambiguous readings. This space is used to encode spin states of fermionic particles, with spin 1/2, such as electrons and protons. A helpful geometric visualization of the states in this space is the *Bloch sphere*, in fig. 3.2.

To interpret the action of the unary connectives in the spin space, we suppose that the particles with spin, our words in this case, are subjected

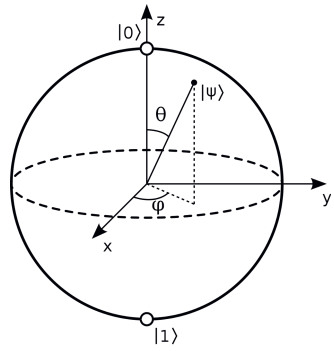


Figure 3.2: Bloch sphere representation of a two-level quantum state, also called a *qubit*. The general form of a state on the surface is $|\Psi\rangle = \left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right)e^{i\gamma}$. The global phase $e^{i\gamma}$ is not represented because it has no effect on the density matrix. A product of states ${}^{pure}\rho^{\mathfrak{S}} = |\Psi\rangle_{\mathfrak{S}}\langle\Psi|$ is called a *pure state*, represented on the surface of the sphere. Otherwise the states are called *mixed states* and live inside of the sphere.

to a uniform magnetic field pointing in the z direction. Using natural units, let

$$S_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

be the spin operator in the z direction. The eigenvectors of this operator are the orthogonal states $|0\rangle = (0,1)^\intercal$ and $|1\rangle = (1,0)^\intercal$, using the standard matrix representation. On the Bloch sphere, these states correspond to the north and south poles, respectively. The corresponding eigenvalues are $e_0 = -1/2$ and $e_1 = 1/2$. This is the operator that we will use to interpret our unary modality. Thus $T_\Diamond$ is the set formed by linear combinations of states $\rho_0 = |0\rangle_{\mathfrak{S}}\langle 0|$ and $\rho_1 = |1\rangle_{\mathfrak{S}}\langle 1|$, the states that lie on the z-axis inside the Bloch sphere.

To interpret controlled commutativity, we use that the raising operator is

$$S_+ = S_x + iS_y = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Once applied on ρ_0 the result is ρ_1, and a further application has a null result. Note that, together with with the lowering operator

$$S_- = S_x - iS_y = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

it obeys the completeness relation $S_+ (S_+)^\dagger + S_- (S_-)^\dagger = \mathbb{1}$.

3.6 GOING DUTCH AGAIN

To illustrate the interpretation process, we return to our Dutch relative clause example "man die de hond bijt", and show how we handle the derivational ambiguity. The lexicon below has the syntactic type assignments and the corresponding semantic spaces:

	syn type A	$\lfloor A \rfloor$
man'	n	$\tilde{N} \otimes \mathfrak{S}$,
die'	$(n\backslash n)/(\Diamond\Box np\backslash s)$	$\tilde{N}^* \otimes \tilde{N} \otimes \tilde{S}^* \otimes \tilde{N} \otimes (\mathfrak{S} \otimes \mathfrak{S}^*) \otimes (\mathfrak{S} \otimes \mathfrak{S}^*) \otimes \mathfrak{S}$,
de hond'	np	$\tilde{N} \otimes \mathfrak{S}$,
bijt'	$np\backslash np\backslash s$	$\tilde{N}^* \otimes \tilde{N}^* \otimes \tilde{S} \otimes \mathfrak{S}$.

In order to compute the interpretations given by the two above derivations, we start from the following primitive interpretations:

$$[\![\text{man}'^{n}]\!]_I = \sum_{rr',ii'} {}^{S}\mathbf{M}^{rr'} \, |r\rangle_{\lceil N \rceil} \langle r'| \otimes^{\mathfrak{S}} \mathbf{M}_{ii'} \, |i\rangle_{\mathfrak{S}} \langle i'|, \tag{3.26}$$

$$[\![\text{die}'^{(n\backslash n)/(\Diamond\Box np\backslash s)}]\!]_I = \sum_{kk',ll',mm',nn',ii'} {}^{S}\mathbf{D}_{k'k\,m'm}^{ll'\quad nn'} \left| {}^{k'\ m'}_{l\quad n} \right\rangle_{\lceil N \rceil^* \otimes \lceil N \rceil \otimes (\lceil S \rceil^* \otimes \lceil N \rceil)} \left\langle {}^{k\ m}_{l'\ n'} \right|$$

$$\otimes^{\mathfrak{S}} \mathbf{D}_{ii} \, |i\rangle_{\mathfrak{S}} \langle i|; \tag{3.27}$$

$$[\![\text{de hond}'^{np}]\!]_I = \sum_{jj',ii'} {}^{S}\mathbf{H}^{jj'} \, |_{j}\rangle_{\tilde{N}} \langle_{j'}| \otimes^{\mathfrak{S}} \mathbf{H}_{ii'} \, |i\rangle_{\mathfrak{S}} \langle i'|; \tag{3.28}$$

$$[\![\text{bijt}'^{np\backslash np\backslash s}]\!]_I = \sum_{oo',pp',qq',ii'} {}^{S}\mathbf{B}_{o'o,p'p}^{\quad qq'} \left| {}^{o'\,p'}_{\quad q} \right\rangle_{\lceil N \rceil^* \otimes \lceil N \rceil^* \otimes \lceil S \rceil} \left\langle {}^{op}_{\ q'} \right| \otimes^{\mathfrak{S}} \mathbf{B}_{ii'} \, |i\rangle_{\mathfrak{S}} \langle i'|. \tag{3.29}$$

To obtain the correct contractions in the spatial components, that are related either to the subject or object relativization readings, the role of the hypothesis x is crucial: interpreted as in eq.(3.13), it contracts with the interpretation of "bijt" as the interpretations of the slash elimination rules prescribe, either in subject or object position. Its most important role is in the latter, blocking "de hond" from taking the immediate object position contraction. After that, variable x is extracted using the *xleft* rule, in a way that keeps all the other contractions unchanged, and keeping the right form such that "die" can contract in the correct position. This process is worked out in Appendix 3.C.1.

With respect to the spin components, the goal is that a pure state is preserved as it interacts with other spin states via slash elimination. As the hypothesis of type $\Diamond\Box np$ is abstracted over, it attains the value of the identity matrix, onto which the box and diamond eliminations are applied, projecting it to the ρ_0 state. If the controlled commutativity rule is is applied, the raising operator brings this pure state to the orthogonal pure state ρ_1. In this way, each of the two readings is stored in one of orthogonal eigenstates of the S_z operator, which are necessarily pure states. As they interact with "man" using the $(.) * (.)$ map, we predict that the final spin states will remain pure, using the result of Lemma 4.1 on the phaser in *Coecke and Meichanetzidis*[31], since the spin state that represents "man"

interacts with a pure state in argument position. The full calculations are shown in Appendix 3.C.2.

The relative clause of the first reading has the interpretation

$$[\![\text{die\_de\_hond\_bijt}']\!]^1_I = \sum_{rr',ll',jj',mm',nn'} {}^S\mathbf{D}_{r'r\ m'm}^{\ ll'\ nn'}\ {}^S\mathbf{H}^{jj'}\ {}^S\mathbf{B}_{j'j,n'n}^{\ \ mm'} \left|\begin{matrix} r' \\ l \end{matrix}\right\rangle_{\lceil N \rceil} \langle {}^{r}_{l'}| \otimes |0\rangle_{\mathfrak{S}} \langle 0|, \tag{3.30}$$

while for the second reading the interpretation is it is

$$[\![\text{die\_de\_hond\_bijt}']\!]^2_I = sum_{rr',ll',jj'mm',nn'} {}^S\mathbf{D}_{r'r\ m'm}^{\ ll'\ nn'}\ {}^S\mathbf{H}^{jj'}\ {}^S\mathbf{B}_{n'n,j'j}^{\ \ mm'} \left|\begin{matrix} r' \\ l \end{matrix}\right\rangle_{\lceil N \rceil} \langle {}^{r}_{l'}| \otimes |1\rangle_{\mathfrak{S}} \langle 1|. \tag{3.31}$$

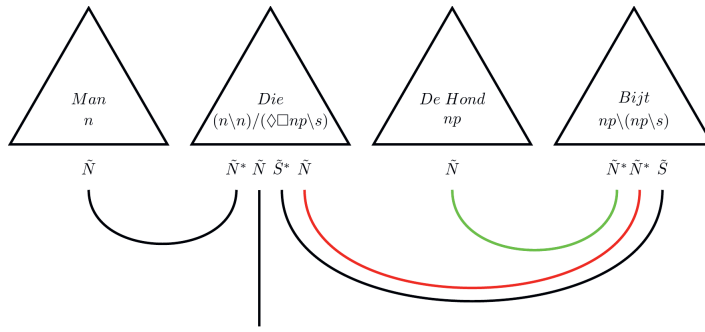


Figure 3.3: Representation of spatial contractions corresponding to the subject relativization reading of "man die de hond bijt", according to eq.(3.30).

The final interpretation of the ambiguous phrase is given by the direct sum of the two unambiguous interpretations, weighted by parameters p_1 and p_2 that express the likelihood of each reading:

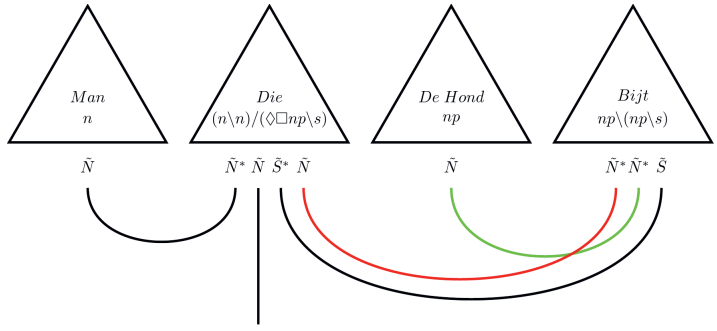


Figure 3.4: Representation of contractions corresponding to the object relativization reading of "man die de hond bijt", according to eq.(3.31).

$$\begin{aligned}&[\![man\_die\_de\_hond\_bijt]\!]_I \\ &= p_1[\![man\_die\_de\_hond\_bijt]\!]_I^1 \oplus p_2[\![man\_die\_de\_hond\_bijt]\!]_I^2. \end{aligned} \tag{3.32}$$

3.7 DISCUSSION AND CONCLUSION

In this paper we extended the interpretation space with a spin degree of freedom, showing how that can preserve extra information about the proof. We showed how interpreting the meanings of words directly as density matrices introduces a framework that can be used to encode higher-level content. This was done by interpreting the unary connectives as quantum operations in the spin space, such that the information about the readings is preserved via a quantum process. When more than two ambiguous readings are possible, it constitutes future work to show that our framework can be extended by using a larger spin space and an appropriate raising operator. Besides its usefulness to deal with ambiguity, in future work we want also to study how the spin degree of freedom is suitable to distinguish the representations of marked types in a multimodal setting, possibly by associating them with eigenstates of different operators. While in this work the spin degree of freedom plays no bigger role than an extra two-dimensional degree of freedom, when going to a multimodal setting

the interactions between the different spin eigenvectors will have quantum properties due to the non-commutativity of the operators. Interesting too is to relate our approach, where lambda terms are directly interpreted using elements and operations over them, with Kripke frames on vector spaces [50], defining the valuation sets with the accessibility relations that translate into our operations, unveiling a stronger connection with the logic of residuation. Also relevant would be to compare our take on interpreting certain logic connectives using quantum mechanical operations with the mirror field of quantum logic [32] that aims at interpreting quantum mechanics using logic tool, particularly modal logic [32] which is at the root of our unary connectives, where too an association between projections and the logic of possibility ($\Diamond$ in our notation) is suggested. Finally, further research will have to show how the probability coefficients can be extracted from derivational data, and whether it is possible to go from the subject relativization reading to the object relativization reading applying only permutation operators as is done in [35] for syntactic ambiguities and, in that case, what is precisely the connection with the derivation. Other interesting questions relate to finding the appropriate categorical interpretation of the spin space and operations that take place there, further helping us to relate our interpretation of control modalities with other logical operators that are also syntactic but do affect the meaning of a sentence, such as negation or quantification, but these are outside the scope of the present paper.

3.A COMPLETE PROOF TREES FOR DUTCH RELATIVIZATION CLAUSES

3.A.1 *Subject Relativization*

$$\dfrac{\dfrac{\text{man}}{y_0 : n}\,\ell \qquad \dfrac{\dfrac{\text{die}}{z_0 : (n\backslash n)/(\Diamond\Box np\backslash s)}\,\ell \qquad \dfrac{\dfrac{\dfrac{[\_ \vdash x_1 : \Diamond\Box np]^1 \qquad \dfrac{\dfrac{[\_ \vdash z_1 : \Box np]^2}{\langle\_\rangle \vdash {}^{\vee}z_1 : np}\,[\Box E] \qquad \dfrac{\dfrac{\dfrac{\text{de}}{x_2 : np/n}\,\ell \qquad \dfrac{\text{hond}}{y_2 : n}\,\ell}{\text{de}\cdot\text{hond} \vdash (x_2 \lhd y_2) : np}\,[/E] \qquad \dfrac{\text{bijt}}{z_2 : np\backslash(np\backslash s)}\,\ell}{(\text{de}\cdot\text{hond})\cdot\text{bijt} \vdash ((x_2 \lhd y_2) \rhd z_2) : np\backslash s}\,[\backslash E]}{\langle\_\rangle\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash ({}^{\vee}z_1 \rhd ((x_2 \lhd y_2) \rhd z_2)) : s}\,[\Diamond E]^2}{\_\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash ({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2)) : s}\,[\backslash I]^1}{(\text{de}\cdot\text{hond})\cdot\text{bijt} \vdash \lambda x_1.({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2)) : \Diamond\Box np\backslash s}\,[/E]}{\text{die}\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash (z_0 \lhd \lambda x_1.({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2))) : n\backslash n}\,[\backslash E]}{\text{man}\cdot(\text{die}\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt})) \vdash (y_0 \rhd (z_0 \lhd \lambda x_1.({}^{\vee\cup}x_1 \rhd ((x_2 \lhd y_2) \rhd z_2)))) : n}$$

3.A.2 *Object Relativization*

$$\dfrac{\dfrac{\text{man}}{y_0 : n}\,\ell \qquad \dfrac{\dfrac{\text{die}}{z_0 : (n\backslash n)/(\Diamond\Box np\backslash s)}\,\ell \qquad \dfrac{\dfrac{\dfrac{[\_ \vdash x_1 : \Diamond\Box np]^1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\text{de}}{x_2 : np/n}\,\ell \qquad \dfrac{\text{hond}}{y_2 : n}\,\ell}{\text{de}\cdot\text{hond} \vdash (x_2 \lhd y_2) : np}\,[/E] \qquad \dfrac{\dfrac{[\_ \vdash z_1 : \Box np]^2}{\langle\_\rangle \vdash {}^{\vee}z_1 : np}\,[\Box E] \qquad \dfrac{\text{bijt}}{z_2 : np\backslash(np\backslash s)}\,\ell}{\langle\_\rangle\cdot\text{bijt} \vdash ({}^{\vee}z_1 \rhd z_2) : np\backslash s}\,[\backslash E]}{(\text{de}\cdot\text{hond})\cdot(\langle\_\rangle\cdot\text{bijt}) \vdash ((x_2 \lhd y_2) \rhd ({}^{\vee}z_1 \rhd z_2)) : s}\,[Comm_{\Diamond}]}{\langle\_\rangle\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash ((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)) : s}\,[\Diamond E]^2}{\_\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash ((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)) : s}\,[\backslash I]^1}{(\text{de}\cdot\text{hond})\cdot\text{bijt} \vdash \lambda x_1.((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)) : \Diamond\Box np\backslash s}\,[/E]}{\text{die}\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt}) \vdash (z_0 \lhd \lambda x_1.((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2))) : n\backslash n}\,[\backslash E]}{\text{man}\cdot(\text{die}\cdot((\text{de}\cdot\text{hond})\cdot\text{bijt})) \vdash (y_0 \rhd (z_0 \lhd \lambda x_1.((x_2 \lhd y_2) \rhd ({}^{\vee\cup}x_1 \rhd z_2)))) : n}$$

3.A.3 *Formal semantics of relative pronouns*

To obtain the usual 'formal semantics' terms, one substitutes for the parameter z_0 the lexical program for the word 'die':

$$\textsc{die} = \lambda x\lambda y\lambda z.((y\ z) \wedge (x\ {}^{\cap\wedge}z))$$

which then, after β conversion and cap-cup and wedge-vee cancellation, reduces to

$$\lambda z.((\textsc{man}\ z) \wedge ((\textsc{bijt}\ (\textsc{de}\ \textsc{hond}))\ z)) \quad \text{(subject reading)}$$

$$\lambda z.((\textsc{man}\ z) \wedge ((\textsc{bijt}\ z)\ (\textsc{de hond}))) \quad \text{(object reading)}$$

3.B INTERPRETATION OF EXTENDED $[xleft]^n$ RULE

To arrive at the interpretation of the *xleft* rule, we compose the interpretations of the rules that it abreviates, explicit on the left part of 3.4. Additionally to the interpretations of $E_{\square}$, $E_{\Diamond}$ and $I_{\backslash}$, we only need to provide the interpretation for $Ass_{\Diamond}$ and $Comm_{\Diamond}$. Strucutral rules do not affect systematically the programme encoded by the associated lambda term. Howver, in this paper we go beyond the "bag of words" view and introduce a specification in the lambda term that results from the $Comm_{\Diamond}$ rule:

$$\frac{\Gamma[\Delta_2 \cdot (\langle \Delta_1 \rangle \cdot \Delta_3)] \vdash t : B}{\Gamma[\langle \Delta_1 \rangle \cdot (\Delta_2 \cdot \Delta_3)] \vdash^{c} t : B}\ Comm_{\Diamond} \tag{3.33}$$

The interpretation of $Comm_{\Diamond}$ is as follows:

$$[\![({}^{c}t)^{B}]\!]_g = [\![t^{B}]\!]_{g^{\mathcal{S}}} \otimes S_+ \left([\![t^{B}]\!]_{g^{\mathfrak{S}}}\right) (S_+)^{\dagger},$$

with S_+ the raising operator in the interpreting space, according to the discussion in sec. 3.4.4. If it is applied n times successively , it is takes the form and respective interpretation

$$[\![\left({}^{c^n}t\right)^{B}]\!]_g = [\![t^{B}]\!]_{g^{\mathcal{S}}} \otimes (S_+)^n \left([\![t^{B}]\!]_{g^{\mathfrak{S}}}\right) \left((S_+)^{\dagger}\right)^n.$$

This extends naturally to the case when the $Comm_{\Diamond}$ rule is never applied, in which case $n = 0$, where we have that $(S_+)^0 = I$.

In what follows we take the necessary steps to arrive at the interpretation of term $\lambda^{l} x.^{c^n} t[^{\cup} x/z]$ in spin space. First, we interpret the application of $Comm_{\Diamond}$:

$$[\![\left({}^{c^n}t\right)^{B}\ [^{\cup}x/z]]\!]_{g^{\mathfrak{S}}} = (S_+)^n\, [\![(t[^{\cup}x/z])^{B}]\!]_{g^{\mathfrak{S}}} \left((S_+)^{\dagger}\right)^n$$

Then we expand on the interpretation of $E_{\Diamond}$:

$$[\![\left(t[^{\cup}x/z]\right)^{B}]\!]_{g^{\mathfrak{S}}} = [\![t^{B}]\!]_{g^{\mathfrak{S}}_{z,^{\cup}x}},$$

which means that

$$[\![(^{\cup}x)^{\Box A}]\!]_{g^{\mathfrak{S}}} = T'^{0}_{\Diamond}\left([\![x^{\Diamond\Box A}]\!]_{g^{\mathfrak{S}}}\right) \tag{3.34}$$

will replace $[\![z^{\Box A}]\!]$ inside of t, appearing here already as the result of the application of $E_{\Box}$:

$$[\![(^{\vee}z)^{A}]\!]_{g^{\mathfrak{S}}} = T'^{0}_{\Box}\left([\![z^{\Box A}]\!]_{g^{\mathfrak{S}}}\right).$$

Finally, abstracting over variable x is interpreted as

$$[\![\lambda^{l}x.^{c^{n}}t[^{\cup}x/z]]\!]_{g^{\mathfrak{S}}} = [\![^{c^{n}}t[^{\cup}x/z]]\!]_{g^{\mathfrak{S}}_{x,I}},$$

such that the only instance of x has its interpretation subsituted by the indentity, namely in eq.(3.34). Putting all these elements together and normalizing, we arrive at the interpretation in eq.(3.25).

3.C CONCRETE INTERPRETATION OF RELATIVE CLAUSES

The derivations in 3.2 have a final term that depends on the variables y_0, z_0, x_2, y_2 z_2 and x_1. The latter is a *bound* variable (as well as the intermediate variable x), due to the lambda abstraction, and the former are *free* variables. Bound variables can be substituted by any free variable during the derivation, via beta reduction, and will take the value of that variable, contrasting with free variables that will be substituted by constants, and interpreted accordingly. An assignment function g assigns bound variables to a later-to-be-defined constant, and assigns free variables to specific constants, here our words. In our assignment, taken as an example, the assignment function gives $g(y_0) = man'$ but $g(x_1)$ remain in this form, until x_1 is substituted by a free variable. Alternatively we can represent the

free variables as bound variables using a lambda abstraction, applied on a constant: $\lambda y_0.y_0(man') \rightarrow man'$.

Looking at the interpretation of any variable stated in the interpretation of the axiom rule in eq.(3.13) and comparing with the interpretation of the constants in eqs.(3.26) to (3.29), we note that both represent the density matrix entries in a symbolic form, where we can apply directly operations like trace and matrix multiplication in the spatial components, or spin operators in the spin components. This permits that, when we perform these calculation step by step using each rule, we can perform them directly on the symbolic representations of interpretations of constants, in eqs.(3.26) to (3.29), as well as of variables that naturaly take the same form as states in eq.(3.13), since it can potentially take the value of *any* other constant.

Therefore, one can impose an assignment that will interpret our particular Dutch relative clause "man die de hond bijt" g that instantiates the free variables like so:

$$[\![(x_2 \triangleleft y_2)]\!]_g = [\![\text{de\_hond}'^{np}]\!]_I, \tag{3.35}$$

$$[\![(z_2)]\!]_g = [\![\text{bijt}'^{np\backslash np\backslash s}]\!]_I, \tag{3.36}$$

$$[\![z_0]\!]_g = [\![\text{die}'^{(n\backslash)/(np\backslash s)}]\!]_I, \tag{3.37}$$

$$[\![y_0]\!]_g = [\![\text{man}'^{n}]\!]_I \tag{3.38}$$

and instantiates the bound variable x according to eq.(3.13).

Substituting these directly in the derivations, we can, step by step, arrive at the final different readings. In what follows we give a full breakdown of these steps, splitting between spatial and spin components, and between subject and object relativization.

3.C.1 *Interpretations in* $\lceil . \rceil$*:*

Subject Relativization

The interpretation of this derivation starts by making use of the interpretation of $E_{\backslash}$ as given in eq.(3.15), substituting the variables by the assigned constants as described above.

$$\begin{aligned}
&[\![(x_2 \triangleleft y_2) \triangleright z_2]\!]_{g^S} = \mathrm{Tr}_{\tilde{N}}\left([\![(x_2 \triangleleft y_2)]\!]_{g^S} \cdot [\![z_2]\!]_{g^S}\right)\\
&\mathrm{Tr}_{\tilde{N}}\left(\sum_{jj'} {}^S\mathbf{H}^{jj'} \left|_j\right\rangle {}_{\tilde{N}}\left\langle_{j'}\right| \cdot \sum_{oo',pp',qq'} {}^S\mathbf{B}_{o'o,p'p}{}^{qq'} \left|{}^{o'\,p'}_{\quad q}\right\rangle_{\lceil N\rceil^*\otimes\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{o\,p}_{\quad q'}\right|\right)\\
&= \sum_{jj',pp',qq'} {}^S\mathbf{H}^{jj'}\,{}^S\mathbf{B}_{j'j,p'p}{}^{qq'} \left|{}^{p'}_{\quad q}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{p}_{\quad q'}\right| \qquad (3.39)
\end{aligned}$$

Then we use again eq.(3.15) and the interpret the variable x using axiom rule as in eq.(3.13).

$$\begin{aligned}
&[\![x \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)]\!]_{g^S} = \mathrm{Tr}_{\tilde{N}}\left([\![x]\!]_{g^S} \cdot [\![(x_2 \triangleleft y_2) \triangleright z_2]\!]_{g^S}\right)\\
&= \mathrm{Tr}_{\tilde{N}}\left(\sum_{ii'} {}^S\mathbf{X}^{ii'} \left|_i\right\rangle {}_{\tilde{N}}\left\langle_{i'}\right| \cdot \sum_{jj',pp',qq'} {}^S\mathbf{H}^{jj'}\,{}^S\mathbf{B}_{j'j,p'p}{}^{qq'} \left|{}^{p'}_{\quad q}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{p}_{\quad q'}\right|\right)\\
&= \sum_{ii',jj',qq'} {}^S\mathbf{X}^{ii'}\,{}^S\mathbf{H}^{jj'}\,{}^S\mathbf{B}_{j'j,i'i}{}^{qq'} \left|q\right\rangle_{\lceil S\rceil} \left\langle q'\right| \qquad (3.40)
\end{aligned}$$

To use the *xleft* rule, we first interpret the previous term in the assignment $g^S_{x,ll'}$, as described in Def.3.4.1. recalculating the previous interpretation using the basis of its interpretation space instead of eq.(3.13).

$$\begin{aligned}\llbracket x \triangleright ((x_2 \triangleleft y_2) \triangleright z_2) \rrbracket_{g^S_{x,ll'}} &= \mathrm{Tr}_{\tilde{N}} \left(|l\rangle_{\lceil N \rceil} \langle l'| \cdot \llbracket (x_2 \triangleleft y_2) \triangleright z_2 \rrbracket_{g^S} \right) \\ &= \sum_{jj',qq'} {}^S\mathbf{H}^{jj'S}\mathbf{B}_{j'j,l'l}{}^{qq'} \left| q \right\rangle_{\lceil S \rceil} \left\langle q' \right| \end{aligned} \tag{3.41}$$

We simplify the spatial interpretation of *xleft* as given in eq.(3.25), using that x and y are interpreted both interpreted in $\lceil A \rceil$, since $\lceil \Diamond\Box A \rceil = \lceil A \rceil$:

$$\begin{aligned}&\llbracket (\lambda^l x.{}^{c^0} t[{}^{\vee\cup} x/y])^{\Diamond\Box A \backslash B} \rrbracket_{g^S} = \\ &= \sum_{ll'} \left| l' \right\rangle_{\lceil A \rceil^*} \left\langle l \right| \otimes \left[\mathrm{Tr}_{\lceil A \rceil} \left(\left[\!\left[x^{\Diamond\Box A} \right]\!\right]_{g^S} \cdot \sum_{kk'} \left| k' \right\rangle_{\lceil A \rceil^*} \left\langle k \right| \otimes \llbracket t^B \rrbracket_{g^S_{y,kk'}} \right) \right]_{g^S_{x,ll'}} \\ &= \sum_{ll'} \left| l' \right\rangle_{\lceil A \rceil^*} \left\langle l \right| \otimes \mathrm{Tr}_{\lceil A \rceil} \left(|l\rangle_{\lceil A \rceil} \langle l'| \cdot \sum_{kk'} \left| k' \right\rangle_{\lceil A \rceil^*} \left\langle k \right| \otimes \llbracket t^B \rrbracket_{g^S_{y,kk'}} \right) \\ &= \sum_{ll'} \left| l' \right\rangle_{\lceil A \rceil^*} \left\langle l \right| \otimes \llbracket t^B \rrbracket_{g^S_{y,ll'}}. \end{aligned} \tag{3.42}$$

Using this simplified form, we see that multiplying with the dual basis of the space that interprets both x and x_1 results in an expression that will take any value of a variable of that type, precisely the goal of the lambda abstraction.

$$\begin{aligned}&\llbracket \lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)) \rrbracket_{g^S} = \sum_{ll'} \left| l' \right\rangle_{\lceil N \rceil^*} \left\langle l \right| \otimes \llbracket x \triangleright ((x_2 \triangleleft y_2) \triangleright z_2) \rrbracket_{g^S_{x,ll'}} \\ &= \sum_{ll'} \left| l' \right\rangle_{\lceil N \rceil^*} \left\langle l \right| \otimes \sum_{jj',qq'} {}^S\mathbf{H}^{jj'S}\mathbf{B}_{j'j,l'l}{}^{qq'} \left| q \right\rangle_{\lceil S \rceil} \left\langle q' \right| \\ &= \sum_{ll',jj',qq'} {}^S\mathbf{H}^{jj'S}\mathbf{B}_{j'j,l'l}{}^{qq'} \left| \begin{matrix} l' \\ q \end{matrix} \right\rangle_{\lceil N \rceil^* \otimes \lceil S \rceil} \left\langle \begin{matrix} l \\ q' \end{matrix} \right| \end{aligned} \tag{3.43}$$

To finalize, the next two steps consist in the application of the interpretations of $E_/$ in eq.(3.14) and $E_\backslash$ (eq.3.15), respectively, resulting in the spatial part of eq. 3.30.

$$
\begin{aligned}
&[\![z_0 \triangleleft \lambda^l x_1.{}^{c^0} t(^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^S} = \\
&\mathrm{Tr}_{\tilde{S}} \left(\mathrm{Tr}_{\tilde{N}} \left([\![z_0]\!]_{g^S} . [\![\lambda^l x_1.{}^{c^0} t(^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^S} \right) \right) \\
&= \mathrm{Tr}_{\tilde{S}} \left(\mathrm{Tr}_{\tilde{N}} \left(\sum_{kk',tt',mm',nn'} {}^S\mathbf{D}_{k'k\ m'm}^{\ tt'\ \ nn'} \left| {}^{k'\ m'}_{\ t\ \ n} \right\rangle_{\lceil N\rceil^* \otimes \lceil N\rceil \otimes (\lceil S\rceil^* \otimes \lceil N\rceil)} \left\langle {}^{k\ m}_{t'\ n'} \right| \right. \right. \\
&\left. \left. \cdot \sum_{ll',jj',qq'} {}^S\mathbf{H}^{jj'\,S}\mathbf{B}_{j'j,l'l}^{\ \ \ \ qq'} \left| {}^{l'}_{q} \right\rangle_{\lceil N\rceil^* \otimes \lceil S\rceil} \left\langle {}^{l}_{q'} \right| \right) \right) \\
&= \sum_{kk',tt',mm',nn',jj'} {}^S\mathbf{D}_{k'k\ m'm}^{\ tt'\ \ nn'} {}^S\mathbf{H}^{jj'\,S}\mathbf{B}_{j'j,n'n}^{\ \ \ \ mm'} \left| {}^{k'}_{t} \right\rangle_{\lceil N\rceil^* \otimes \lceil N\rceil} \left\langle {}^{k}_{t'} \right| \qquad (3.44)
\end{aligned}
$$

$$
\begin{aligned}
&[\![y_0 \triangleright (z_0 \triangleleft \lambda^l x_1.{}^{c^0} t(^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)))]\!]_{g^S} = \\
&= \mathrm{Tr}_{\tilde{N}} \left([\![y_0]\!]_{g^S} \cdot [\![z_0 \triangleleft \lambda^l x_1.{}^{c^0} t(^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^S} \right) \\
&= \mathrm{Tr}_{\tilde{N}} \left(\sum_{rr'} {}^S\mathbf{M}^{rr'} \left| r \right\rangle_{\lceil N\rceil} \left\langle r' \right| \right. \\
&\left. \cdot \sum_{kk',tt',mm',nn',jj'} {}^S\mathbf{D}_{k'k\ m'm}^{\ tt'\ \ nn'} {}^S\mathbf{H}^{jj'\,S}\mathbf{B}_{j'j,n'n}^{\ \ \ \ mm'} \left| {}^{k'}_{t} \right\rangle_{\lceil N\rceil^* \otimes \lceil N\rceil} \left\langle {}^{k}_{t'} \right| \right) \\
&= \sum_{rr',tt',mm',nn',jj'} {}^S\mathbf{M}^{rr'\,S}\mathbf{D}_{r'r\ m'm}^{\ \ tt'\ \ nn'} {}^S\mathbf{H}^{jj'\,S}\mathbf{B}_{j'j,n'n}^{\ \ \ \ mm'} \left| t \right\rangle_{\lceil N\rceil} \left\langle t' \right| \qquad (3.45) \\
&= [\![\text{man\_die\_de\_hond\_bijt'}]\!]^1_{I^S} \qquad (3.46)
\end{aligned}
$$

Object relativization

This derivation is very similar to the previous, except that on the first application of $E_\backslash$ the bound variable x is introduced as the argument of

z_2, and only on the next application of the rule is $(x_2 \triangleleft y_2)$ taken as an argument.

$$
\begin{aligned}
&[\![x \triangleright z_2]\!]_{g^S} = \mathrm{Tr}_{\tilde{N}}\left([\![x]\!]_{g^S} \cdot [\![z_2]\!]_{g^S}\right) \\
&= \mathrm{Tr}_{\tilde{N}}\left(\sum_{ii'} {}^S\mathbf{X}^{ii'} \, |_i\rangle_{\tilde{N}} \langle_{i'}| \cdot \sum_{oo',pp',qq'} {}^S\mathbf{B}_{o'o,p'p}^{\;\;qq'} \left|{}^{o'p'}_{\;\;q}\right\rangle_{\lceil N\rceil^*\otimes\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{op}_{\;\;q'}\right|\right) \\
&= \sum_{ii',pp',qq'} {}^S\mathbf{X}^{ii'\,S}\mathbf{B}_{i'i,p'p}^{\;\;qq'} \left|{}^{p'}_{\;\;q}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{p}_{\;\;q'}\right| \qquad (3.47)
\end{aligned}
$$

$$
\begin{aligned}
&[\![(x_2 \triangleleft y_2) \triangleright (x \triangleright z_2)]\!]_{g^S} = \mathrm{Tr}_{\tilde{N}}\left([\![(x_2 \triangleleft y_2)]\!]_{g^S} \cdot [\![x \triangleright z_2]\!]_{g^S}\right) \\
&= \mathrm{Tr}_{\tilde{N}}\left([\![(x_2 \triangleleft y_2)]\!]_{g^S} \cdot \mathrm{Tr}_{\tilde{N}}\left([\![x]\!]_{g^S} \cdot [\![z_2]\!]_{g^S}\right)\right) \\
&= \mathrm{Tr}_{\tilde{N}}\left(\sum_{jj'} {}^S\mathbf{H}^{jj'} \, |_j\rangle_{\tilde{N}} \langle_{j'}| \cdot \sum_{ii',pp',qq'} {}^S\mathbf{X}^{ii'\,S}\mathbf{B}_{i'i,p'p}^{\;\;qq'} \left|{}^{p'}_{\;\;q}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle{}^{p}_{\;\;q'}\right|\right) \\
&= \sum_{jj',ii',qq'} {}^S\mathbf{H}^{jj'\,S}\mathbf{X}^{ii'\,S}\mathbf{B}_{i'i,j'j}^{\;\;qq'} \left|q\right\rangle_{\lceil S\rceil} \left\langle q'\right| \qquad (3.48)
\end{aligned}
$$

Note at this point that, due to changing the ordering of contraction, when compared with the subject relativization reading, the matrix indices are contracted differently from eq.3.40. We see now what the role of the hypotheses x is: to block $(x_2 \triangleright y_2)$ from contracting inevitably as the first argument of z_2. Now that the contraction is in line with what we want for an object relativization reading, we will extract variable x via *xleft*. To do that, we first reinterpret the previous term using the assignment $g^S_{x,ll'}$. To substitute the interpretation of x by that of its basis elements we need to go further into de proof, when compared with the subject relativization reading.

$$\begin{aligned}
&[\![(x_2 \triangleleft y_2) \triangleright (x \triangleright z_2)]\!]_{g^S_{x,ll'}} = \mathrm{Tr}_{\tilde{N}}\left([\![(x_2 \triangleleft y_2)]\!]_{g^S} \cdot \mathrm{Tr}_{\tilde{N}}\left(|l\rangle_{\lceil N\rceil}\langle l'| \cdot [\![z_2]\!]_{g^S}\right)\right) \\
&= \mathrm{Tr}_{\tilde{N}}\left(\sum_{jj'} {}^S\mathbf{H}^{jj'} \,|j\rangle_{\tilde{N}}\langle j'| \cdot \sum_{pp',qq'} {}^S\mathbf{B}_{l'l,p'p}{}^{qq'} \left|\begin{smallmatrix} p' & \\ & q\end{smallmatrix}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle\begin{smallmatrix} p & \\ & q'\end{smallmatrix}\right|\right) \\
&= \sum_{jj',qq'} {}^S\mathbf{H}^{jj'}{}^S\mathbf{B}_{l'l,j'j}{}^{qq'} \left|q\right\rangle_{\lceil S\rceil}\left\langle q'\right| \qquad (3.49)
\end{aligned}$$

The following steps are as before, with the final result referring to eq.3.31.

$$\begin{aligned}
&[\![\lambda^l x_1.{}^{c^1} t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^S} = \sum_{ll'} \left|l'\right\rangle_{\lceil N\rceil^*}\left\langle l\right| \otimes [\![((x_2 \triangleleft y_2) \triangleright (x \triangleright z_2))]\!]_{g^S_{x,ll'}} \\
&= \sum_{ll'} \left|l'\right\rangle_{\lceil N\rceil^*}\left\langle l\right| \otimes \sum_{jj',qq'} {}^S\mathbf{H}^{jj'}{}^S\mathbf{B}_{l'l,j'j}{}^{qq'} \left|q\right\rangle_{\lceil S\rceil}\left\langle q'\right| \qquad (3.50) \\
&= \sum_{ll',jj',qq'} {}^S\mathbf{H}^{jj'}{}^S\mathbf{B}_{l'l,j'j}{}^{qq'} \left|\begin{smallmatrix} l' & \\ & q\end{smallmatrix}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle\begin{smallmatrix} l & \\ & q'\end{smallmatrix}\right| \qquad (3.51)
\end{aligned}$$

$$\begin{aligned}
&[\![z_0 \triangleleft \lambda^l x_1.{}^{c^1} t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^S} \\
&= \mathrm{Tr}_{\tilde{S}}\left(\mathrm{Tr}_{\tilde{N}}\left([\![z_0]\!]_{g^S}.[\![\lambda^l x_1.{}^{c^1} t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^S}\right)\right) \\
&= \mathrm{Tr}_{\tilde{S}}\left(\mathrm{Tr}_{\tilde{N}}\left(\sum_{kk',tt',mm',nn'} {}^S\mathbf{D}_{k'k\,m'm}^{tt'\quad nn'} \left|\begin{smallmatrix} k' & m' \\ t & n\end{smallmatrix}\right\rangle_{\lceil N\rceil^*\otimes\lceil N\rceil\otimes(\lceil S\rceil^*\otimes\lceil N\rceil)} \left\langle\begin{smallmatrix} k & m \\ t' & n'\end{smallmatrix}\right|\right.\right. \\
&\left.\left. \cdot \sum_{ll',jj',qq'} {}^S\mathbf{H}^{jj'}{}^S\mathbf{B}_{l'l,j'j}{}^{qq'} \left|\begin{smallmatrix} l' & \\ & q\end{smallmatrix}\right\rangle_{\lceil N\rceil^*\otimes\lceil S\rceil} \left\langle\begin{smallmatrix} l & \\ & q'\end{smallmatrix}\right|\right)\right) \\
&= \sum_{kk',tt',mm',nn',jj'} {}^S\mathbf{D}_{k'k\,m'm}^{tt'\quad nn'}{}^S\mathbf{H}^{jj'}{}^S\mathbf{B}_{n'n,j'j}{}^{mm'} \left|\begin{smallmatrix} k' & \\ & t\end{smallmatrix}\right\rangle_{\lceil N\rceil^*\otimes\lceil N\rceil} \left\langle\begin{smallmatrix} k & \\ & t'\end{smallmatrix}\right| \qquad (3.52)
\end{aligned}$$

$$
\begin{aligned}
&[\![y_0 \triangleright (z_0 \triangleleft \lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2)))]\!]_{g^S} \\
&= \mathrm{Tr}_{\tilde{N}}\left([\![y_0]\!]_{g^S} \cdot [\![z_0 \triangleleft \lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^S}\right) \\
&= \mathrm{Tr}_{\tilde{N}}\left(\sum_{rr'} {}^{S}\mathbf{M}^{rr'} \, |_r\rangle_{\lceil N\rceil}\langle_{r'}| \right. \\
&\left. \cdot \sum_{kk',tt',mm',nn',jj'} {}^{S}\mathbf{D}_{k'k\ m'm}^{tt'\ \ nn'}\,{}^{S}\mathbf{H}^{jj'}\,{}^{S}\mathbf{B}_{n'n,j'j}^{\ \ mm'} \left|{}^{k'}_{t}\right\rangle_{\lceil N\rceil^*\otimes\lceil N\rceil}\left\langle{}^{k}_{t'}\right| \right) \\
&= \sum_{rr',tt',mm',nn',jj'} {}^{S}\mathbf{M}^{rr'}\,{}^{S}\mathbf{D}_{r'r\ m'm}^{tt'\ \ nn'}\,{}^{S}\mathbf{H}^{jj'}\,{}^{S}\mathbf{B}_{n'n,j'j}^{\ \ mm'} \, |_t\rangle_{\lceil N\rceil}\langle_{t'}| \qquad (3.53) \\
&= [\![\text{man\_die\_de\_hond\_bijt'}]\!]^2_{I^S}. \qquad (3.54)
\end{aligned}
$$

3.c.2 *Interpretations in $\mathfrak{S}$:*

Subject Relativization

We start by using the interpretations of variables in the interpretation of $E_{\backslash}$ as given in eq. 3.15, which are particular forms of eq. 3.11. The variables can have any value with the only requirement that it is neither ρ_0 nor ρ_1. This is because the resulting states must have a non-zero probability of being projected on either of these states, which is necessary for the following step.

$$
\begin{aligned}
&[\![(x_2 \triangleleft y_2) \triangleright z_2]\!]_{g^{\mathfrak{S}}} = [\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}} \\
&= \frac{\left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![z_2]\!]_{g^{\mathfrak{S}}} \cdot \left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![z_2]\!]_{g^{\mathfrak{S}}} \cdot \left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)}. \qquad (3.55)
\end{aligned}
$$

$$\begin{aligned}\llbracket x \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)\rrbracket_{g^{\mathfrak{S}}} &= \left(\llbracket z_2\rrbracket_{g^{\mathfrak{S}}} * \llbracket x_2 \triangleleft y_2\rrbracket_{g^{\mathfrak{S}}}\right) * \llbracket x\rrbracket_{g^{\mathfrak{S}}} \\ &= \frac{\left(\llbracket x\rrbracket_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot \left(\llbracket z_2\rrbracket_{g^{\mathfrak{S}}} * \llbracket x_2 \triangleleft y_2\rrbracket_{g^{\mathfrak{S}}}\right) \cdot \left(\llbracket x\rrbracket_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left(\llbracket x\rrbracket_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot \left(\llbracket z_2\rrbracket_{g^{\mathfrak{S}}} * \llbracket x_2 \triangleleft y_2\rrbracket_{g^{\mathfrak{S}}}\right) \cdot \left(\llbracket x\rrbracket_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)}\end{aligned} \tag{3.56}$$

Looking at the interpretation of *xleft* in eq. 3.25, we first work out eq. 3.24 with $\llbracket x\rrbracket_{g^{\mathfrak{S}}}$ substituted by $\llbracket {}^{\vee\cup}x_1\rrbracket = T'^{0}_{\square}\left(T'^{0}_{\Diamond}\left(\llbracket x_1\rrbracket_{g^{\mathfrak{S}}}\right)\right)$ because of assignment $g^{\mathfrak{S}}_{x,{}^{\vee\cup}x_1}$, and with $\llbracket x_1\rrbracket_{g^{\mathfrak{S}}}$ substituted by I in its turn, because of the assignment $g^{\mathfrak{S}}_{x_1,I}$. Recall that in our definitions $U_0 = \mathbb{1}$. Since controlled commutativity is not used, $n = 0$ and $(S_+)^0 = \mathbb{1}$. In both steps below, pure state ρ_0 will be preserved, taking into account that

$$\llbracket t^A\rrbracket_{g^{\mathfrak{S}}} * \llbracket u^B\rrbracket_{g^{\mathfrak{S}}} = \llbracket u^B\rrbracket_{g^{\mathfrak{S}}}, \tag{3.57}$$

when $\llbracket u^B\rrbracket_{g^{\mathfrak{S}}}$ equals ρ_0 or ρ_1. To show this, take $\llbracket t^A\rrbracket_{g^{\mathfrak{S}}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and $\llbracket u^B\rrbracket_{g^{\mathfrak{S}}} = |0\rangle_{\mathfrak{S}}\langle 0| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$,

$$\llbracket t^A\rrbracket_{g^{\mathfrak{S}}} * \llbracket u^B\rrbracket_{g^{\mathfrak{S}}} = \frac{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}{\mathrm{Tr}\left(\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\right)} = \frac{\begin{pmatrix} 0 & 0 \\ 0 & d \end{pmatrix}}{d} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \tag{3.58}$$

and similarly for $\llbracket u^B\rrbracket_{g^{\mathfrak{S}}} = \rho_1$.

Therefore, the concrete interpretation of the *xleft* rule uses

$$\llbracket {}^{\vee\cup}x_1\rrbracket = T'^{0}_{\square}\left(T'^{0}_{\Diamond}(I)\right) = I * |0\rangle_{\mathfrak{S}}\langle 0| = |0\rangle_{\mathfrak{S}}\langle 0|, \tag{3.59}$$

which substituted in eq.3.56 gives

$$\begin{aligned}
&[\![\lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^{\mathfrak{S}}} = \\
&= S^0_+ \left(\left([\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}} \right) * |0\rangle_{\mathfrak{S}} \langle 0| \right) \left(S^0_+\right)^\dagger \\
&= |0\rangle_{\mathfrak{S}} \langle 0| \qquad (3.60)
\end{aligned}$$

In the following two steps, the interpretations of rules $E_/$ and $E_\backslash$ are used. In the last step of 3.61, we refer again to eq.3.30.

$$\begin{aligned}
&[\![z_0 \triangleleft \lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^{\mathfrak{S}}} = [\![z_0]\!]_{g^{\mathfrak{S}}} * [\![\lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2))]\!]_{g^{\mathfrak{S}}} \\
&= [\![z_0]\!]_{g^{\mathfrak{S}}} * |0\rangle_{\mathfrak{S}} \langle 0| = \frac{(|0\rangle_{\mathfrak{S}} \langle 0|)^{\frac{1}{2}} \cdot [\![z_0]\!]_{g^{\mathfrak{S}}} \cdot (|0\rangle_{\mathfrak{S}} \langle 0|)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}} \left((|0\rangle_{\mathfrak{S}} \langle 0|)^{\frac{1}{2}} \cdot [\![z_0]\!]_{g^{\mathfrak{S}}} \cdot (|0\rangle_{\mathfrak{S}} \langle 0|)^{\frac{1}{2}} \right)} = |0\rangle_{\mathfrak{S}} \langle 0| \\
&= [\![\text{die_de_hond_bijt'}]\!]^1_{I^{\mathfrak{S}}} \qquad (3.61)
\end{aligned}$$

$$\begin{aligned}
&[\![y_0 \triangleright (z_0 \triangleleft \lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)))]\!]_{g^{\mathfrak{S}}} \\
&= [\![(z_0 \triangleleft \lambda^l x_1.{}^{c^0} t({}^{\vee\cup} x_1 \triangleright ((x_2 \triangleleft y_2) \triangleright z_2)))]\!]_{g^{\mathfrak{S}}} * [\![y_0]\!]_{g^{\mathfrak{S}}} \\
&= |0\rangle_{\mathfrak{S}} \langle 0| * [\![y_0]\!]_{g^{\mathfrak{S}}} = \frac{\left([\![y_0]\!]_{g^{\mathfrak{S}}} \right)^{\frac{1}{2}} \cdot |0\rangle_{\mathfrak{S}} \langle 0| \cdot \left([\![y_0]\!]_{g^{\mathfrak{S}}} \right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}} \left(\left([\![y_0]\!]_{g^{\mathfrak{S}}} \right)^{\frac{1}{2}} \cdot |0\rangle_{\mathfrak{S}} \langle 0| \cdot \left([\![y_0]\!]_{g^{\mathfrak{S}}} \right)^{\frac{1}{2}} \right)} \\
&= [\![\text{man_die_de_hond_bijt'}]\!]^1_{I^{\mathfrak{S}}}. \qquad (3.62)
\end{aligned}$$

Object Relativization

Just as in the previous derivations, once more we use the interpretations of $E_\backslash$ in the two first steps.

$$[\![x \triangleright z_2]\!]_{g^{\mathfrak{S}}} = [\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x]\!]_{g^{\mathfrak{S}}} = \frac{\left([\![x]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![z_2]\!]_{g^{\mathfrak{S}}} \cdot \left([\![x]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left([\![x]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot [\![z_2]\!]_{g^{\mathfrak{S}}} \cdot \left([\![x]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)}. \quad (3.63)$$

$$\begin{aligned}[\![(x_2 \triangleleft y_2) \triangleright (x \triangleright z_2)]\!]_{g^{\mathfrak{S}}} &= \left([\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x]\!]_{g^{\mathfrak{S}}}\right) * [\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}} \\ &= \frac{\left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot \left([\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x]\!]_{g^{\mathfrak{S}}}\right) \cdot \left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot \left([\![z_2]\!]_{g^{\mathfrak{S}}} * [\![x]\!]_{g^{\mathfrak{S}}}\right) \cdot \left([\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)}\end{aligned} \quad (3.64)$$

In the application of the interpretation of $xleft$ in eq.3.4) is the same as in the previous reading, except that controlled commutation is used once, so that $m = 1$, meaning that $(S_+)^1 = S_+$, $U_0 = \mathbb{1}$:

$$[\![^{\vee\cup}x_1]\!] = T'^0_{\square}\left(T'^0_{\Diamond}(I)\right) = I * |0\rangle_{\mathfrak{S}}\langle 0| = |0\rangle_{\mathfrak{S}}\langle 0|, \quad (3.65)$$

which substituted in 3.56 gives

$$\begin{aligned}&[\![\lambda^l x_1.^{c^1} t(x_2 \triangleleft y_2) \triangleright (^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^{\mathfrak{S}}} = \\ &= \frac{S_+\left(\left([\![z_2]\!]_{g^{\mathfrak{S}}} * |0\rangle_{\mathfrak{S}}\langle 0|\right) * [\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)(S_+)^\dagger}{\mathrm{Tr}_{\mathfrak{S}}\left(S_+\left(\left([\![z_2]\!]_{g^{\mathfrak{S}}} * |0\rangle_{\mathfrak{S}}\langle 0|\right) * [\![x_2 \triangleleft y_2]\!]_{g^{\mathfrak{S}}}\right)(S_+)^\dagger\right)} \\ &= |1\rangle_{\mathfrak{S}}\langle 1|, \end{aligned} \quad (3.66)$$

since

$$\frac{S_+[\![t^A]\!]^{\mathfrak{S}}_g S_+^\dagger}{\mathrm{Tr}_{\mathfrak{S}}\left(S_+[\![t^A]\!]^{\mathfrak{S}}_g S_+^\dagger\right)} = \frac{\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}a&b\\c&d\end{pmatrix}\begin{pmatrix}0&0\\1&0\end{pmatrix}}{\mathrm{Tr}\left(\begin{pmatrix}0&1\\0&0\end{pmatrix}\begin{pmatrix}a&b\\c&d\end{pmatrix}\begin{pmatrix}0&0\\1&0\end{pmatrix}\right)} = \frac{\begin{pmatrix}d&0\\0&0\end{pmatrix}}{d}$$

$$= \begin{pmatrix}1&0\\0&0\end{pmatrix} = |1\rangle_{\mathfrak{S}}\langle 1|. \tag{3.67}$$

Finally, for the interpretations of $E_/$ and $E_\backslash$:

$$\begin{aligned}
&[\![z_0 \triangleleft \lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^{\mathfrak{S}}} = \\
&[\![z_0]\!]_{g^{\mathfrak{S}}} * [\![\lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^{\mathfrak{S}}} \\
&= \frac{(|1\rangle_{\mathfrak{S}}\langle 1|)^{\frac{1}{2}} \cdot [\![z_0]\!]_{g^{\mathfrak{S}}} \cdot (|1\rangle_{\mathfrak{S}}\langle 1|)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left((|1\rangle_{\mathfrak{S}}\langle 1|)^{\frac{1}{2}} \cdot [\![z_0]\!]_{g^{\mathfrak{S}}} \cdot (|1\rangle_{\mathfrak{S}}\langle 1|)^{\frac{1}{2}}\right)} = |1\rangle_{\mathfrak{S}}\langle 1| \\
&= [\![\text{die_de_hond_bijt'}]\!]^2_{I^{\mathfrak{S}}}.
\end{aligned} \tag{3.68}$$

$$\begin{aligned}
&[\![y_0 \triangleright (z_0 \triangleleft \lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2)))]\!]_{g^{\mathfrak{S}}} \\
&= [\![z_0 \triangleleft \lambda^l x_1.{}^{c^1}t((x_2 \triangleleft y_2) \triangleright ({}^{\vee\cup}x_1 \triangleright z_2))]\!]_{g^{\mathfrak{S}}} * [\![y_0]\!]_{g^{\mathfrak{S}}} \\
&= \frac{\left([\![y_0]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot |1\rangle_{\mathfrak{S}}\langle 1| \cdot \left([\![y_0]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}}{\mathrm{Tr}_{\mathfrak{S}}\left(\left([\![y_0]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}} \cdot |1\rangle_{\mathfrak{S}}\langle 1| \cdot \left([\![y_0]\!]_{g^{\mathfrak{S}}}\right)^{\frac{1}{2}}\right)} \\
&= [\![\text{man_die_de_hond_bijt'}]\!]^2_{I^{\mathfrak{S}}}.
\end{aligned} \tag{3.69}$$

3.D PROOF TRANSFORMATION: BETA REDUCTION

The β-reduction is one of the rewrite rules of the λ-calculus. It asserts that applying a term with a lambda-bound variable to a certain argument is equivalent to substituting that argument directly in the original term, before introducing the lambda. In proof-theoretic terms, if an introduction rule is used followed by an elimination rule, the derivation is not minimal. To elucidate this point, below is the skeleton of a derivation where a term of type A is proved twice, by axiom and by an unknown proof:

$$\dfrac{\dfrac{\vdots}{\Delta \vdash n : A} \quad \dfrac{\dfrac{\dfrac{}{x : A \vdash x : A}\,axiom}{\vdots}}{\dfrac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda^l x.m : A \backslash B}\backslash_I}\backslash_E}{(\Gamma, \Delta) \vdash n \triangleright (\lambda^l x.m) : B}.$$

The β reduction consists of substituting the unknown proof of the term of type A in place of the axiom, reducing the need for the double proof of that term, and consequently the size of the proof:

$$\dfrac{\dfrac{\vdots}{\Delta \vdash n : A}}{\dfrac{\vdots}{\Delta, \Gamma \vdash m[x/n] : B}}.$$

Through this reduction, a map from one conclusion to the other can be obtained, which has to be an equality regarding their interpretations:

$$[\![n \triangleright (\lambda^l x.m)]\!]_g = [\![m[x/n]]\!]_g, \forall g.$$

This equality will be used to check that the density matrix construction interpretation is consistent with the λ-calculus. Below a concrete symbolic derivation before the reduction is shown:

$$\dfrac{\dfrac{\dfrac{}{w:B\vdash w:B}\,ax \quad \dfrac{}{z:B\backslash(A/B)\vdash z:B\backslash(A/B)}\,ax}{w:B,z:B\backslash(A/B)\vdash(w\triangleright z):A/B}\backslash_{E_2} \quad \dfrac{\dfrac{\dfrac{}{x:A/B\vdash x:A/B}\,ax \quad \dfrac{}{y:B\vdash y:B}\,ax}{x:A/B,y:B\vdash(x\triangleleft y):A}/_{E_1}}{y:B\vdash\lambda^l x.(x\triangleleft y):(A/B)\backslash A}\backslash_{I_1}}{(w:B,z:B\backslash(A/B),u:B)\vdash(w\triangleright z)\triangleright(\lambda^l x.(x\triangleleft y)):A}\backslash_{E_3}\,.$$

The interpretation in the spatial space S of the several steps of the proof is given below, following the numbering in the proof:

$$E_{/_1}:\ [\![(x\triangleleft y)]\!]_{g^S}=\sum_{ii',jj'}{}^S\mathbf{X}^{i'i}_{jj'}\,{}^S\mathbf{Y}^{j'j}\,|_i\rangle_{\lceil A\rceil}\langle_{i'}|,$$

$$I_{\backslash_1}:\ [\![\lambda^l x.(x\triangleleft y)]\!]_{g^S}=\sum_{ii',jj'}\left|{}^{i'}_{j}\right\rangle_{\lceil B\rceil\otimes\lceil A\rceil^*}\left\langle{}^{i}_{j'}\right|\otimes{}^S\mathbf{Y}^{j'j}\,|_i\rangle_{\lceil A\rceil}\langle_{i'}|,$$

$$E_{\backslash_2}:\ [\![(w\triangleright z)]\!]_g=\sum_{ll',mm',nn'}{}^S\mathbf{W}^{ll'}\,{}^S\mathbf{Z}^{m'm}_{l'l,nn'}\left|{}^{n'}_{m}\right\rangle_{\lceil A\rceil\otimes\lceil B\rceil^*}\langle_{m'}^{n}|,$$

$$E_{\backslash_3}:\ [\![(w\triangleright z)\triangleright(\lambda^l x.(x\triangleleft y))]\!]_{g^S}=\sum_{ii',jj',ll'}{}^S\mathbf{W}^{ll'}\,{}^S\mathbf{Z}^{i'i}_{l'l,jj'}\,{}^S\mathbf{Y}^{j'j}\,|_i\rangle_{\lceil A\rceil}\langle_{i'}|.$$

In spin space $\mathfrak{S}$ the interpretation of the proof steps is as follows:

$$E_{/_1}:\ [\![(x\triangleleft y)]\!]_{g^{\mathfrak{S}}}=[\![x]\!]_{g^{\mathfrak{S}}}*[\![y]\!]_{g^{\mathfrak{S}}}$$

$$I_{\backslash_1}:\ [\![\lambda^l x.(x\triangleleft y)]\!]_{g^{\mathfrak{S}}}=I*[\![y]\!]_{g^{\mathfrak{S}}}=[\![y]\!]_{g^{\mathfrak{S}}}$$

$$E_{\backslash_2}:\ [\![(w\triangleright z)]\!]_g=[\![z]\!]_{g^{\mathfrak{S}}}*[\![w]\!]_{g^{\mathfrak{S}}}$$

$$E_{\backslash_3}: \ [\![(w \triangleright z) \triangleright (\lambda^l x.(x \triangleleft y))]\!]_{g^S} = [\![y]\!]_{g^{\mathfrak{S}}} * \left([\![z]\!]_{g^{\mathfrak{S}}} * [\![w]\!]_{g^{\mathfrak{S}}} \right)$$

A similar treatment is done for the derivation after the reduction:

$$\dfrac{\dfrac{\dfrac{}{w : B \vdash w : B}\, ax \quad \dfrac{}{z : B\backslash(A/B) \vdash z : B\backslash(A/B)}\, ax}{w : B, z : B\backslash(A/B) \vdash (w \triangleright z) : A/B}\, \backslash_{E_2} \quad \dfrac{}{y : B \vdash y : B}\, ax}{w : B, z : B\backslash(A/B), u : B \vdash ((w \triangleright z) \triangleleft y) : A}\, \backslash_{E_4}.$$

The value of $[\![(w \triangleright z)\rangle]\!]_g$ is the same as before. For $[\![((w \triangleright z) \triangleleft y)]\!]_{g^S}$:

$$E_{\backslash_4}: \ [\![((w \triangleright z) \triangleleft y)]\!]_{g^S} = \sum_{ii',jj',ll'} {}^S\mathbf{W}^{ll'} {}^S\mathbf{Z}_{l'l,jj'}^{i'i} {}^S\mathbf{Y}^{j'j} \, |_i\rangle \, \ulcorner_A \urcorner \langle_{i'}|.$$

On the spin space, we have

$$E_{\backslash_4}: \ [\![((w \triangleright z) \triangleleft y)]\!]_{g^{\mathfrak{S}}} = [\![y]\!]_{g^{\mathfrak{S}}} * \left([\![z]\!]_{g^{\mathfrak{S}}} * [\![w]\!]_{g^{\mathfrak{S}}} \right)$$

Comparing the two derivations and interpretations, the conclusion is that

$$[\![E_{\backslash_4}(y, z(w))]\!]_{g^S} = [\![E_{\backslash_3}(z(w), \lambda x.x(y))]\!]_{g^S},$$

as expected, and

$$[\![E_{\backslash_4}(y, z(w))]\!]_{g^{\mathfrak{S}}} = [\![E_{\backslash_3}(z(w), \lambda x.x(y))]\!]_{g^{\mathfrak{S}}}.$$

4

COMPARING IN CONTEXT

ABSTRACT Cosine similarity is a widely used measure of the relatedness of pre-trained word embeddings, trained on a language modeling goal. Datasets such as WordSim-353 and SimLex-999 rate how similar words are according to human annotators, and as such are often used to evaluate the performance of language models. Thus, any improvement on the word similarity task requires an improved word representation. In this paper, we propose instead the use of an extended cosine similarity measure to improve performance on that task, with gains in interpretability. We explore the hypothesis that this approach is particularly useful if the word-similarity pairs share the same context, for which distinct contextualized similarity measures can be learned. We first use the dataset of Richie et al. (2020) to learn contextualized metrics and compare the results with the baseline values obtained using the standard cosine similarity measure, which consistently shows improvement. We also train a contextualized similarity measure for both SimLex-999 and WordSim-353, comparing the results with the corresponding baselines, and using these datasets as independent test sets for the all-context similarity measure learned on the contextualized dataset, obtaining positive results for a number of tests.

4.1 INTRODUCTION

Cosine similarity has been largely used as a measure of word relatedness, since vector space models for text representation appeared to automatically optimize the task of information retrieval [122]. While other distance measures are also commonly used, such as Euclidean distance [146], for cosine similarity only the vector directions are relevant, and not their norms. More recently, pre-trained word representations, also referred to as embeddings, obtained from neural network language models, starting from word2vec (W2V) [89], emerged as the main source of word embeddings, and are subsequently used in model performance evaluation on tasks such as word similarity [133]. Datasets such as SimLex-999 [58] and WordSim-353 [44], which score similarity between word-pairs according to the assessment of several humans annotators, have become the benchmarks for the performance of a certain type of embedding on the task of word similarity [8, 39, 114, 132].

For $\vec{n}_a$ and $\vec{n}_b$, the vector representations of two distinct words w_a and w_b, cosine similarity takes the form

$$cos_{ab} = \frac{\vec{n}_a \cdot \vec{n}_b}{||\vec{n}_a||\,||\vec{n}_b||}, \tag{4.1}$$

with the Euclidean *inner product* between any two vectors $\vec{n}_a$ and $\vec{n}_b$ given as

$$\vec{n}_a \cdot \vec{n}_b = \sum_i \vec{n}_a^i \vec{n}_b^i, \tag{4.2}$$

and the *norm* of a vector $\vec{n}_a$ given as

$$||\vec{n}_a|| = \sqrt{\vec{n}_a \cdot \vec{n}_a}, \tag{4.3}$$

dependent on the inner product [7].

Using this measure of similarity, improvements can only take place if the vectors that represent the words change. However, the assumption that the vectors interact using a Euclidean inner product becomes less plausible

when it comes to higher order vectors. If, differently, we consider that the vector components are not described in a Euclidean basis, then we enlarge the possible relationships between the vectors. Specifically in the calculation of the inner product, on which the cosine similarity depends, we can use an intermediary *metric* tensor. By challenging the assumption that the underlying metric is Euclidean, cosine similarity values can be improved *without changing vector representations*.

We identify two main motivations to search for improved cosine similarity measures. The first motivation has to do with the cost of training larger and more refined language models [13]. By increasing the performance on a task simply by changing the evaluation measure without changing the pre-trained embeddings, we expect that better results can be achieved with more efficient and interpretable methods. This is particularly true of contextualized datasets, with benefits not only for tasks such as word similarity, but also others that use cosine similarity as a measure of relatedness, such as content based recommendation systems [125], and where it can be particularly interesting to explore the different metrics that emerge as representations of vector relatedness.

The second motivation comes from compositional distributional semantics, where words of different syntactic types are represented by tensors of different ranks, and representations of larger fragments of text are produced via tensor contraction [11, 33, 51, 52, 90, 103]. This framework has proved to be a valuable tool for low resource languages, enhancing the scarce available data with a grammatical structure for composition, providing embeddings of complex expressions [2]. As these contractions depend on an underlying metric that is usually taken to be Euclidean, improvements have only been achieved, once again, by modifying word representations [144]. As proposed by Correia u. a. [35], another way to improve on these results consists in using a different metric to mediate tensor contractions. Metrics obtained in tasks such as word similarity can be transferred to tensor contraction, and thus we expect this work to open new research avenues on the compositional distributional framework, providing a better integration with (contextual) language models.

This paper is organized as follows. In §4.2 we introduce an extended cosine similarity measure, motivating the introduction of a metric on the hypothesis that it can optimize the relationships between the vectors. In §4.3 we explain our experiment on contextualized and non-contextualized datasets to test whether improvements can be achieved. In §4.4 we present the results obtained in our experiments and in §5.5 we discuss these results and propose further work.

Our contributions are summarized below:

- ◇ Use of contextualized datasets to explore contextualized dynamic embeddings and evaluate the viability of contextualized similarity measures;
- ◇ Expansion of the notion of cosine similarity, motivating our model theoretically, contributing to a conceptual simplification that yields interpretable improvements.

4.1.1 *Related Literature*

Variations on similarity metrics on the contextualized dataset of Richie u. a. [116] have been first explored in Richie und Bhatia [115], but only on static vector representations and diagonal metrics. Other analytical approaches to similarity learning have been identified in Kulis u. a. [65]. The notion of soft cosine similarity of Sidorov u. a. [129] presents a relevant extension theoretically similar to ours, but motivated and implemented differently. Using count-base vector space models with words and n-grams as features, the authors extract a similarity score between features, using external semantic information, that they use as a distance matrix that can be seen as a metric; however, they do not implement it as in Eq. (4.4), but instead they transform the components by creating a higher dimensional vector space where each entry is the average of the components in two features, multiplied by the metric, whereas we, by contrast, learn the metric automatically and apply it to the vectors directly. Hewitt und Manning [57] also use a modified metric for inner product to probe the syntactic structure

of the representations, showing that syntax trees are embedded implicitly in deep models' vector geometry.

Context dependency in how humans evaluate similarity, which we based our study on, has been widely supported in the psycholinguistic literature. Tversky [136] shows that similarity can be expressed as a linear combination of properties of objects, Barsalou [12] looks at how context-dependent and context-independent properties influence similarity perception, Medin u. a. [82] explore how similarity judgments are constrained by the very fact of being requested, and Goldstone u. a. [49] test how similarity judgments are influenced by context that can either be explicit or perceived.

4.2 MODEL

A metric is a tensor that maps any two vectors to an element of the underlying field $\mathbb{K}$, which in this case will be the field of real numbers $\mathbb{R}$. This element is what is known as the *inner product*. To this effect, the metric tensor can be represented as a function, not necessarily linear, over each of the coordinates of the vectors it acts on. In geometric terms, the metric characterizes the underlying geometry of a vector space, by describing the projection of the underlying manifold of a non-Euclidean geometry to a Euclidean geometry $\mathbb{R}^n$ [140]. The inner product between two vectors is informed by the metric in a precise way, and is representative of how the distance between two vectors should be calculated.

A standard example consists of two unit vectors on a sphere, which is an S^2 manifold that can be mapped onto $\mathbb{R}^3$. If the vectors are represented in spherical coordinates, which are a map from S^2 to $\mathbb{R}^3$, the standard method of computing the angle between the vectors using Eq. (4.1) will fail to give the correct value. The vectors need to be transformed by the appropriate non-linear metric to the Euclidean basis in $\mathbb{R}^3$ before a contraction of the coordinates can take place. To illustrate this, take as an example a triangle drawn on the surface of a sphere S^2. If it is projected onto a planisphere $\mathbb{R}^3$, a naive measurement of its internal angles will exceed the known 180 degrees, which corresponds to a change in the inner product between the

vectors tangents to the triangle corners (see Levart [74] for a demonstration). To preserve this inner product, and thus recover the equivalence between a triangle on a spherical surface and a triangle on a Euclidean plane, the coordinates need to be properly transformed by the appropriate metric before they are contracted.

By the same token, we explore here the possibility that the shortcomings of the values obtained using cosine similarity when compared with human similarity ratings are not due to poor vector representations, but to a measure that fails to assess the distance between the vectors adequately. To test this hypothesis, we generalize the inner product of Eq. (4.2) to accommodate a larger class of relationships between vectors, modifying it using a metric represented by the distance matrix d, once a basis is assumed, that defines the inner product between two vectors as

$$\vec{n}_a \cdot_d \vec{n}_b = \sum_{ij} \vec{n}_a^i d^{ij} \vec{n}_b^j, \tag{4.4}$$

where $\vec{n}_a^i$ is the ith component of $\vec{n}_a$. Using a metric of this form, the best we can achieve is a linear rescaling of the components of the vectors, which entails the existence of a non-orthogonal basis. The metric d is required to be bilinear and symmetric, which is satisfied if

$$d^{sym} = B^T B, \tag{4.5}$$

such that Eq. (4.4) can be rewritten as

$$\vec{n}_a \cdot_d \vec{n}_b = (B\vec{n}_a)^T \cdot (B\vec{n}_b)\,. \tag{4.6}$$

We can thus learn the components of a metric for a certain set of vectors by fitting it to the goal of preserving a specified inner product. In the case of word similarity, the matrix B can be learned supervised on human similarity judgments, towards the goal that a contextualized cosine similarity applied to a set of word embeddings, using Eq. (4.6), returns the correct human assessment. An advantage of this approach is that the cosine is symmetric

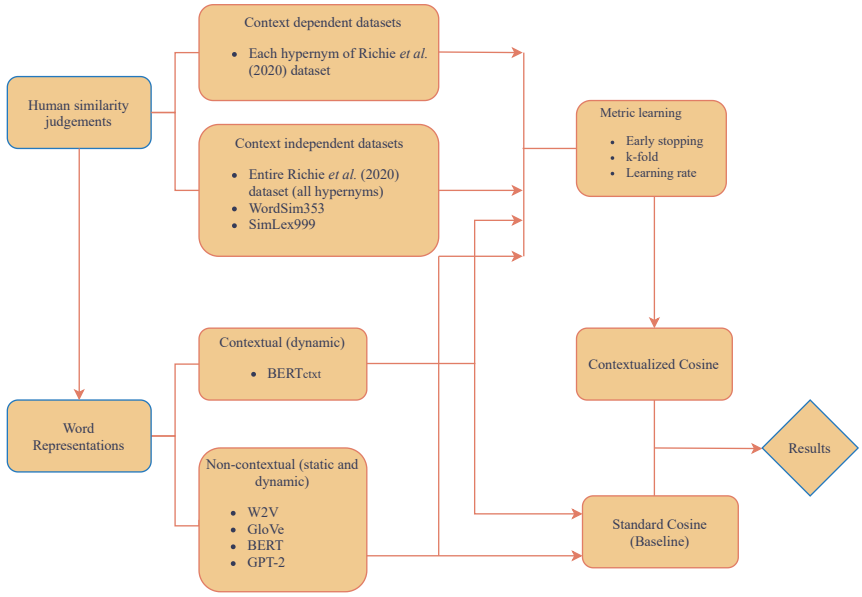


Figure 4.1: Schematic representation of the experiment leading up to the results in Tables 4.4 and 4.5.

with respect to its inputs, which is a nice property that this extension preserves by requiring that symmetry of the metric.

4.3 METHODS

The general outline of our experiment is as follows. First, we learn contextualized cosine similarity measures for related (contextualized) pairs of words, and afterwards for unrelated (non-contextualized) pairs of words. A schematic representation can be found in Fig. 4.1. We then test whether these learned measures are transferable and provide improvements on word pairs that were not seen during training, when compared with the standard cosine similarity baseline.

4.3.1 *Datasets*

For a contextualized assessments of word similarity, we use the dataset of Richie u. a. [116], where 365 participants were asked to judge the similarity between English word-pairs that are co-hyponyms of eight different hypernyms (Table 4.1). Participants were assigned a specific hypernym and were asked to rate the similarity between each co-hyponym pair from 1 to 7, with the highest rating indicating the words to be maximally similar. The number of annotators varies per hypernym, but each word-pair is rated by around 30 annotators, such that for the largest categories each annotator only saw a fraction of the totality of the word-pairs. As examples from the hypernym 'Clothing', the word-pair 'hat/overalls' was rated by 32 of the 61 annotators, resulting in an average similarity of 1.469, while 'coat/gloves' had an average similarity rating of 3.281 and 'coat/jacket' of 6.438, also by 32 annotators. The average similarity was computed for all word-pairs and rescaled to a value between 0 and 1, to be used as the target for supervised learning.

Besides trying to fit a contextualized similarity measure to each hypernym, we also considered the entire all-hypernyms dataset, in order to test whether training on the hypernyms separately would result in a better cosine measure compared with when the hypernym information was disregarded.

To test whether similarity measures can be learned if the similarity of words is not assessed within a specific context, we use the WordSim-353 (WS353) [44] and part of the SimLex-999 (SL999) [58] datasets, where the word-pairs bear no specific semantic relation. From the SL999 dataset only the nouns were included, resulting in a dataset of 666 word-pairs. Additionally, we use these datasets to verify whether the similarity metric learned by training on the whole dataset of [116] can be transferred to other, more general, datasets.

Hypernym	Words	Pairs	Annotators
Birds	30	435	54
Clothing	29	406	61
Professions	28	378	67
Sports	28	378	61
Vehicles	22	231	28
Fruit	21	210	31
Furniture	20	190	33
Vegetables	20	190	30
All	198	2418	365

Table 4.1: Number of words, word-pairs and human annotators per hypernym.

Representation	**Corpus**	**Corpus size**	**Dim**
word2vec	Google News	100B	300
GloVe	GigaWord Corpus & Wikipedia	6B	200
$\text{BERT}_{\text{base-uncased}}$	BooksCorpus & English Wikipedia	3.3B	768
$\text{GPT-2}_{\text{medium}}$	8 million web pages	$\sim$ 40 GB	768

Table 4.2: Pre-trained embeddings obtained from different source language models, with BERT and GPT-2 implemented using the Huggingface Transformers library.

4.3.2 *Word embeddings*

To fine-tune the cosine similarity measure, we start from different pre-trained word representations. We do that for two classes of embeddings, static and dynamic.

Static embeddings were obtained from a pre-trained word2vec (W2V) model [89] and a pre-trained GloVe model [106], each used to encode each word in the pair. Dynamic embeddings were obtained from two Transformers-based models, pre-trained BERT [38] and GPT-2 models [111] (see Table 4.2). Here the representation of each word was taken to be the average representation of sub-word tokens when necessary, excluding the [CLS] and [SEP] tokens.

The token representations provided by the BERT model, as a bidirectional dynamic language model, can change depending on the surrounding con-

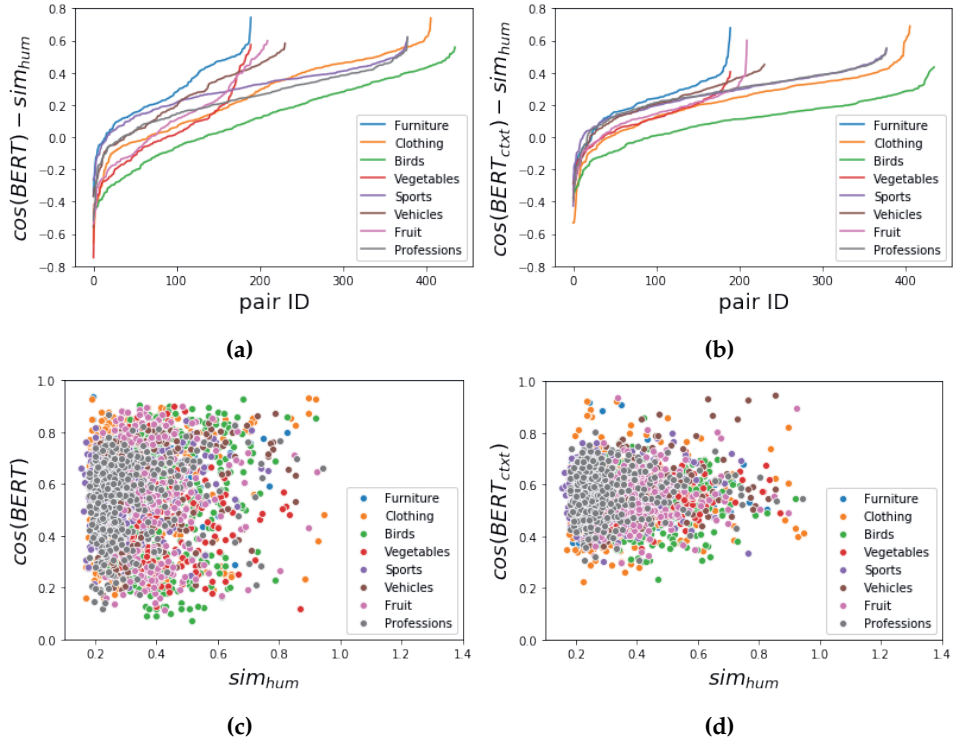


Figure 4.2: Distributions of pairwise human similarity judgments sim_{hum} and cosine similarity measures using either BERT representations (cos(BERT)) or contextualized BERT representations (cos(BERT_{ctxt})). In (a) and (b) the absolute difference of scores, ordered per hypernym, is shown. while (c) and (d) represent the distribution of different similarity scores with respect to each other. Comparing (a) and (b), we can see a regularization effect by contextualizing the representations, and between (c) and (d) we can see a clustering effect.

Hypernym	Context words
Birds	small, migratory, other, water, breeding
Clothing	cotton, heavy, outer, winter, leather
Professions	health, legal, engineering, other, professional
Sports	youth, women, men, ea, boys
Vehicles	military, agricultural, motor, recreational, commercial
Fruit	citrus, summer, wild, sweet, passion
Furniture	wood, furniture, modern, antique, office
Vegetables	some, wild, root, fresh, green

Table 4.3: Five most likely words for masked token preceding hypernym token using BERT.

text tokens. As such, additional contextualized embeddings were retrieved, BERT$_{ctxt}$, to test whether performance could be improved relative to the baseline cosine metric by using the hypernym information, as well as when compared with the hypernym cosine metric learned on non-contextualized representations. In this way we test whether leveraging the contextual information intrinsic to this dataset can in itself improve similarity at the baseline level, without the need of further training.

The contextualized vectors of BERT$_{ctxt}$ were obtained by first having BERT predict the five most likely adjectives that precede each hypernym using ([MASK] <hypernym>), and then using those adjectives to obtain five contextualized embeddings for each co-hyponym, subsequently averaged over. Most of the predicted words were adjectives, and the few cases that were not were filtered out. For instance, for the category 'Clothing', the most likely masked tokens were 'cotton', 'heavy', 'outer', 'winter' and 'leather'. The contextualized representation of each hyponyms of 'Clothing' was thus calculated as its average representation in the context of each of the

adjectives, so that, for instance, for 'coat' we first obtained its contextualized representation in 'cotton coat', 'heavy coat', 'outer coat', 'winter coat', and 'leather coat', performing a final averaging. The full list of context words can be found in Table 4.3. Figs. 4.2a and 4.2b show that this transformation reduces the absolute extreme values of the difference between the values of the standard cosine similarity and the corresponding human similarity assessments, while regularizing the bulk of the differences closer to the desired value of 0. We tested other forms of contextualizing, such as (`<hypernym> is/are [MASK]`), but the resulting representations did not show as much improvement.

The WS353 and SL999 datasets were only trained with non-contextualized embeddings, since we cannot obtain contextualized embeddings for the nouns in these datasets using the same method. For consistency, the models that were learned with contextualized representations were not tested on these datasets at the final step of our experiment.

4.3.3 *Model*

A linear model was implemented on the PyTorch machine learning framework to learn the parameters of B, without a bias, such that a word initially represented by input_a is transformed to $\text{input'}_a = B\text{input}_a$. The forward function of this model takes two inputs and returns

$$\frac{(\text{input'}_a)^T \cdot \text{input'}_b}{\sqrt{(\text{input'}_a)^T \cdot \text{input'}_1}\sqrt{(\text{input'}_b)^T \cdot \text{input}_b}}, \tag{4.7}$$

where a and b correspond to the indices of the words of a given word-pair[a].

a https://github.com/maradf/Contextualized-Cosine

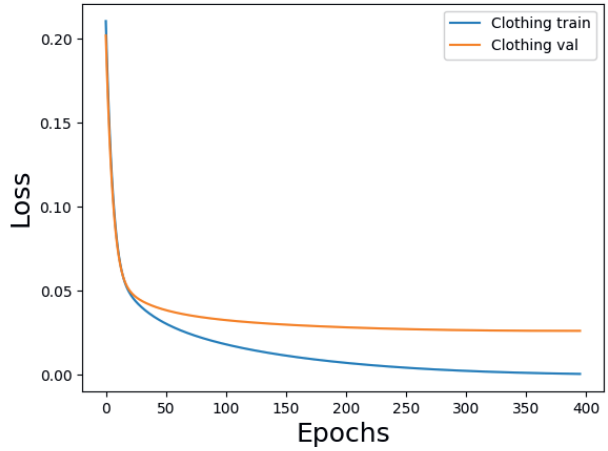


Figure 4.3: Example of learning curve, showing losses over epochs, from a fold training on the hypernym **Clothes** on the GloVe embeddings. In this case, training was stopped early at 397 epochs.

4.3.4 *Cross-validation*

The number of co-hyponyms per hypernym is small when compared with the number of parameters in B to be trained, which depends on the square of the dimension **Dim** of each representation. To ensure that the models did not overfit, a k-fold cross-validation was used during training [113], which divided each dataset in k training sets and non-overlapping development sets. Additionally, early stopping of training was implemented in the event that the validation loss increased for ten consecutive epochs after it dropped below 0.1 [17].

4.3.5 *Hyperparameter selection*

Per each dataset h (each hypernym, all hypernyms, WS353 or SL999) and learning rate l_r, k models B^h_{i,l_r} were trained, with $i \in \{1, ..., k\}$ and with k

corresponding validation sets val_i. The training was done using two 16 cores (64 threads) Intel Xeon CPU at 2.1 GHz.

A fixed seed was used to find the best combination of the learning rate l_r (1×10^{-5}, 1×10^{-6}, and 1×10^{-7}) and the number of folds (5, 6 and 7) for the k-fold cross-validation. The regression to the best metric was done using the mean square error loss function and the Adam optimizer. The maximum number of training epochs was set to 500, as most models converged at that point as per preliminary learning curve inspection (Fig.4.3). The implementation of early stopping resulted in *de facto* variation of the number of epochs required to train each model.

4.3.6 *Testing the model*

Each one of the B^h_{i,l_r} models was tested on the corresponding holdout validation set val_i, resulting in two correlation scores between the models' predicted similarity scores and the human judgment scores: a Pearson correlation score $r^h_{i,l_r}(val^h_i)$ and a Spearman correlation score $\rho^h_{i,l_r}(val^h_i)$. A final score per k and l_r was calculated using the average performance on the validation sets as

$$r^h_{k,l_r} = \frac{1}{k}\sum_{i=1}^{k} r^h_{i,l_r}(val^h_i), \tag{4.8}$$

$$\rho^h_{k,l_r} = \frac{1}{k}\sum_{i=1}^{k} \rho^h_{i,l_r}(val^h_i). \tag{4.9}$$

The baseline results were obtained in a similar form, but with the model B^{std} corresponding to the identity matrix, returning the standard cosine similarity rating as

$$r_k^{h,std} = \frac{1}{k}\sum_{i=1}^{k} r^{std}(val_i^h), \tag{4.10}$$

$$\rho_k^{h,std} = \frac{1}{k}\sum_{i=1}^{k} \rho^{std}(val_i^h). \tag{4.11}$$

The model results shown in Table 4.4 correspond to the best correlation values obtained using Eqs. (4.8) and (4.9), with the baselines given as in Eqs. (4.10) and (4.11). The hyperparameters corresponding to the best results can be found in Table 4.5, along with the relative change in correlation performance. As the seed was fixed, the differences in performance achieved by models trained on each hypernym and on all-hypernyms of the contextualized dataset were not due to randomization errors. The final correlation per fold on the entire all-hypernyms dataset was found by first calculating the correlation per hypernym and then averaging over all eight hypernyms.

To test the transferability of the metric learned on the all-hypernyms dataset to other datasets, the model that returned the best correlation scores on the validation datasets of the all-hypernyms dataset was tested on the entire WS353 and SL999 datasets. As the best performing model consists in fact of k models, each one of these was tested on the entire datasets, as

$$r_{k,l_r}^{h,test} = \frac{1}{k}\sum_{i=1}^{k} r_{i,l_r}^{All-hyp}(test^h), \tag{4.12}$$

$$\rho_{k,l_r}^{h,test} = \frac{1}{k}\sum_{i=1}^{k} \rho_{i,l_r}^{All-hyp}(test^h), \tag{4.13}$$

with $h \in \{\text{WS353}, \text{SL999}\}$.

The baselines for these results were obtained by applying B^{std} to the entire WS353 and SL999 datasets as

$$r^{h,std} = r^{std}(test^h), \tag{4.14}$$

$$\rho^{h,std} = \rho^{std}(test^h). \tag{4.15}$$

As the correlation functions are not linear, the results from Eqs. (4.10) and (4.11) for the WS353 and SL999 datasets are expected to differ from those obtained using Eqs. (4.14) and (4.15) for the same datasets.

4.4 RESULTS

The validation results on Table 4.4 show consistent improvements over the baselines, with statistical significance. This confirms that the modification introduced to the cosine measure worked in a principled way, and consistent with the results found by Richie und Bhatia [115]. On the individual hypernym datasets, 'Vehicles' showed the best correlations, except for the Pearson correlation in GPT-2, in spite of not being the largest hypernym dataset. On the contrary, the smallest categories showed the lowest correlations. In general, the relative performance of hypernyms according to the baselines extends to the model correlations, although with better performance. With some exceptions, mainly in the 'Birds' hypernym, the best performing representation was GPT-2, followed by W2V, but the relative increase as shown in Table 4.5 was clearly superior for the dynamic representations. An important observation that we make is that the model trained on all hypernyms had a better performance than the average performance on the individual hypernyms. As the seed was fixed, this means that the performance on the hypernym-specific validation sets increased if at training time the models saw more examples, from different categories, indicating that a similarity relationship was learned and transferred across different contexts. Improvements over baseline also took place if a metric was learned on datasets where the word pairs did not share a context, as was the case with WS353 and SL999, but the percentual increase was lower, as seen in Table 4.5.

Comparing the results of BERT contextualized and non-contextualized, the baseline values of the contextualized representations were worse than those obtained with the contextualized embeddings, although without statistical significance, while the improvement after training was consistently better and significant for all datasets with the contextualized representations. Figs. 4.2c and 4.2d, show that the distribution of points using the contextualized embeddings is more concentrated and collinear, making it more likely that a metric that acts in the same way for all points in the dataset will rotate and rescale them into a positive correlation. The percentual increases also show that BERT contextualized had the greatest increases from before to after training, suggesting that there was a cumulative effect in considering the context both in the representations and in the similarity measure.

Table 4.6 shows the results of applying the best model learned on all hypernyms to the WS353 and SL999 datasets. The baseline values for the static representations are comparable with the existing literature [133]. We see that our model was capable of improving on the correlation scores on the datasets, for some representations. Although the improvements did not happen across the board, they show clear evidence that the notion of similarity in the form of a modified cosine measure can be learned in one dataset and applied with positive results to an independent dataset.

4.5 CONCLUSION AND OUTLOOK

In this paper we tested whether a contextualized notion of cosine similarity could be learned, improving the similarity not only of the results for the datasets where it was learned, but of unrelated similarities. We showed that this metric improved the correlations above baseline, and that, when learned on a contextualized similarity dataset, it had an advantage when compared to one learned on a dataset with unrelated word-pairs. We furthermore showed that this framework has the potential to generalize the notion of similarity to word-pairs it has not seen during training. An important future research line towards interpretability consists in understanding the properties of the metrics that yielded the best results, particularly

in identifying the distinctive features of the best metrics, such as their eigensystems. Other further directions include applying these metrics to distributional compositional contractions, including with dependency enhancements [64], testing this framework on larger contextualized datasets and trying out more complex, non-linear, metric forms.

ACKNOWLEDGEMENTS

All authors would like to thank Juul A. Schoevers for contributions made during the early stages of the project. A.D.C. would like to thank Gijs Wijnholds, Konstantinos Kogkalidis, Michael Moortgat and Henk T.C. Stoof for the many exchanges during this research. This work is supported by the UU Complex Systems Fund, with special thanks to Peter Koeze.

(a) Pearson correlations.

Dataset (h)	BERT		BERT_{ctxt}		GPT-2		word2vec		GloVe	
	Model	Base	Model	Base	Model	Base	Model	Base	Model	Base
Birds	**0.311**	0.098	**0.316**	0.042	**0.200**	-0.023	**0.293**	0.213	**0.215**	0.194
Clothing	**0.550**	0.141	**0.515**	0.065	**0.501**	0.349	**0.529**	0.417	**0.574**	0.364
Professions	**0.501**	0.193	**0.601**	0.073	**0.651**	0.542	**0.635**	0.566	0.529	0.529
Sports	**0.452**	0.175	**0.543**	0.139	**0.556**	0.324	**0.532**	0.418	**0.580**	0.386
Vehicles	**0.496**	0.218	**0.616**	0.123	**0.645**	0.385	**0.738**	0.719	**0.703**	0.567
Fruit	**0.315**	0.016	**0.378**	-0.037	**0.333**	0.203	**0.361**	0.239	**0.571**	0.392
Furniture	**0.353**	-0.018	**0.539**	-0.035	**0.568**	0.399	**0.368**	0.333	**0.470**	0.462
Vegetables	**0.211**	-0.059	**0.293**	-0.044	**0.378**	0.144	**0.577**	0.281	**0.562**	0.290
All	**0.434**	0.100	**0.542**	0.040	**0.508**	0.287	**0.483**	0.400	**0.539**	0.397
WS353	**0.517**	0.238	-	-	**0.651**	0.647	0.637	0.654	**0.622**	0.568
SL999	**0.403**	0.161	-	-	**0.555**	0.504	**0.495**	0.455	**0.510**	0.408

(b) Spearman correlations.

Dataset (h)	BERT		BERT_{ctxt}		GPT-2		word2vec		GloVe	
	Model	Base	Model	Base	Model	Base	Model	Base	Model	Base
Birds	**0.260**	0.102	**0.299**	0.052	**0.190**	-0.054	**0.250**	0.211	**0.238**	0.201
Clothing	**0.436**	0.184	**0.467**	0.059	**0.445**	0.276	**0.510**	0.414	**0.513**	0.384
Professions	**0.501**	0.248	**0.578**	0.170	**0.560**	0.473	**0.518**	0.410	0.482	0.486
Sports	**0.391**	0.174	**0.526**	0.142	**0.540**	0.291	**0.458**	0.339	**0.478**	0.325
Vehicles	**0.518**	0.238	**0.601**	0.056	**0.626**	0.288	**0.709**	0.687	**0.680**	0.596
Fruit	**0.265**	-0.014	**0.333**	-0.103	**0.365**	0.173	**0.368**	0.277	**0.491**	0.342
Furniture	**0.353**	-0.032	**0.491**	-0.120	**0.527**	0.393	**0.442**	0.402	**0.464**	0.451
Vegetables	**0.217**	-0.028	**0.305**	0.015	**0.363**	0.089	**0.587**	0.290	**0.528**	0.228
All	**0.407**	0.111	**0.504**	0.034	**0.504**	0.242	**0.446**	0.379	**0.477**	0.377
WS353	**0.543**	0.267	-	-	**0.715**	0.705	0.675	0.701	**0.624**	0.579
SL999	**0.416**	0.180	-	-	**0.566**	0.513	**0.475**	0.445	**0.500**	0.374

Table 4.4: Best correlation scores between human similarity judgments and similarity scores found by the trained model, compared with baseline cosine metric values of the same hyperparameters. The underlined correlation values are the statistical significant values with a $p < 0.05$, and the bold values correspond to model correlations that were higher than base correlations.

(a) Pearson correlations.

Dataset (h)	BERT		BERT_{ctxt}		GPT-2		W2V		GloVe	
	%	lr, k	%	lr, k	%	lr, k	%	lr, k	%	lr, k
Birds	217	10^{-6},5	652	10^{-6}, 5	**770**	10^{-5}, 5	38	10^{-5}, 5	11	10^{-5}, 7
Clothing	290	10^{-6},5	**692**	10^{-6}, 6	44	10^{-5}, 6	27	10^{-5}, 7	58	10^{-6}, 5
Professions	160	10^{-6}, 5	**723**	10^{-6}, 6	20	10^{-5}, 5	12	10^{-5}, 7	0	10^{-5}, 5
Sports	158	10^{-5}, 6	**291**	10^{-6}, 6	72	10^{-5}, 6	27	10^{-5}, 6	50	10^{-6}, 7
Vehicles	128	10^{-6}, 6	**401**	10^{-5}, 7	68	10^{-5}, 5	3	10^{-5}, 5	24	10^{-6}, 6
Fruit	**1869**	10^{-5}, 7	922	10^{-6}, 6	64	10^{-5}, 7	51	10^{-6}, 5	46	10^{-7}, 7
Furniture	**1861**	10^{-5}, 7	1440	10^{-6}, 6	42	10^{-5}, 7	11	10^{-5}, 6	2	10^{-5}, 6
Vegetables	258	10^{-5}, 7	**566**	10^{-6}, 6	163	10^{-5}, 5	105	10^{-6}, 7	94	10^{-6}, 5
All	334	10^{-5}, 5	**1255**	10^{-6}, 7	77	10^{-5}, 6	21	10^{-5}, 6	36	10^{-7}, 6
WS353	**117**	10^{-6}, 7	-	-	1	10^{-5}, 7	-3	10^{-5}, 6	10	10^{-5}, 5
SL999	**150**	10^{-6}, 7	-	-	10	10^{-5}, 6	9	10^{-5}, 6	25	10^{-6}, 5

(b) Spearman correlations.

Dataset (h)	BERT		BERT_{ctxt}		GPT-2		W2V		GloVe	
	%	lr, k	%	lr, k	%	lr, k	%	lr, k	%	lr, k
Birds	155	10^{-6}, 5	**475**	10^{-6}, 5	252	10^{-5}, 7	18	10^{-5}, 5	18	10^{-7}, 5
Clothing	137	10^{-6}, 5	**692**	10^{-6}, 6	61	10^{-5}, 7	23	10^{-5}, 7	34	10^{-6}, 5
Professions	102	10^{-6}, 7	**240**	10^{-6}, 5	18	10^{-5}, 5	26	10^{-5}, 7	-1	10^{-7}, 6
Sports	125	10^{-5}, 6	**270**	10^{-6}, 6	86	10^{-5}, 6	35	10^{-5}, 6	47	10^{-6}, 6
Vehicles	118	10^{-6}, 6	**973**	10^{-6}, 6	117	10^{-5}, 7	3	10^{-5}, 5	14	10^{-6}, 6
Fruit	**1793**	10^{-6}, 7	223	10^{-6}, 6	111	10^{-5}, 6	33	10^{-6}, 6	44	10^{-7}, 7
Furniture	**1003**	10^{-6}, 6	309	10^{-6}, 5	34	10^{-5}, 5	10	10^{-5}, 6	3	10^{-6}, 7
Vegetables	675	10^{-5}, 7	**1933**	10^{-6}, 6	308	10^{-5}, 5	102	10^{-6}, 7	132	10^{-6}, 5
All	267	10^{-5}, 5	**1382**	10^{-6}, 7	108	10^{-5}, 6	18	10^{-5}, 6	27	10^{-6}, 5
WS353	**103**	10^{-6}, 5	-	-	1	10^{-5}, 7	-4	10^{-6}, 5	8	10^{-5}, 5
SL999	**131**	10^{-6}, 7	-	-	10	10^{-5}, 6	7	10^{-5}, 6	34	10^{-6}, 5

Table 4.5: Change (%) in correlation from Table 4.4, given by $(|\text{Model}| - |\text{Base}|)/|\text{Base}|$, at corresponding best hyperparameters (lr, k). Values in bold indicate the highest increase on a given dataset.

		Pearson		Spearman	
		WS353	SL999	WS353	SL999
BERT	Model	**0.487**	**0.375**	**0.519**	**0.384**
	Base	0.239	0.151	0.267	0.172
GPT-2	Model	0.635	**0.507**	0.676	0.513
	Base	0.647	0.504	0.709	0.520
W2V	Model	0.613	0.472	0.632	**0.457**
	Base	0.653	0.460	0.700	0.452
GloVe	Model	**0.593**	**0.431**	0.558	**0.392**
	Base	0.578	0.408	0.578	0.376
SOTA		0.704	0.658	0.828	0.76

Table 4.6: Best model trained on all hypernyms, tested on SimLex-999 and WordSim-353 datasets. Bold values indicate correlation scores above baseline, and underlining indicates statistical significance. State of the art from Banjade u. a. [8], Dobó und Csirik [39], Recski u. a. [114], Speer u. a. [132].

5

QUANTUM COMPUTATIONS FOR DISAMBIGUATION AND QUESTION ANSWERING

ABSTRACT Automatic text processing is now a mature discipline in computer science, and so attempts at advancements using quantum computation have emerged as the new frontier, often under the term of Quantum Natural Language Processing. The main challenges consist in finding the most adequate ways of encoding words and their interactions on a quantum computer, considering hardware constraints, as well as building algorithms that take advantage of quantum architectures, so as to show improvement on the performance of natural language tasks. In this chapter, we introduce a new framework that starts from a grammar that can be interpreted by means of tensor contraction, to build word representations as quantum states that serve as input to a quantum algorithm. We start by introducing an operator measurement to contract the representations of words, resulting in the representation of larger fragments of text. We then go on to develop pipelines for the tasks of sentence-meaning disambiguation and question answering that take advantage of quantum features. For the first task, we show that our contraction scheme deals with syntactically ambiguous phrases storing the various different meanings in quantum superposition, a solution not available on a classical setting. For the second task, we obtain a question representation that contains all possible answers in equal quantum superposition, and we implement Grover's quantum search algorithm to find the correct answer, agnostic to the specific question, an implementation with the potential of delivering a result with quadratic speedup.

5.1 INTRODUCTION

Recent developments in quantum computation have given rise to new and exciting applications in the field of Natural Language Processing (NLP). Pioneering work in this direction is the DisCoCat framework [30, 33], which introduces a compositional mapping between types and derivations of Lambek's typelogical grammars [68, 72] and a distributional semantics [135] based on vector spaces, linear maps and tensor products. In this framework, the interpretations of large text fragments are obtained by performing a tensor contraction between the tensor interpretations of individual words. To interpret text fragments taking into account their grammatical features, while staying in the vector space semantics, the dimension of the representation quickly scales, as it depends on the complexity of the syntactic type, which has been a limiting feature in vector-based semantics implementations [145]. This motivates a representation of words as quantum states, counting on the potential of quantum computers to outperform the limitations of classical computation both in terms of memory use [48] and in processing efficiency [6]. In this setting, words are represented as multipartite quantum states, with the theory predicting that, when contracted with one another, the meaning of larger text fragments is encoded in the resulting quantum states.

The challenge is now in implementing these contractions on quantum circuits. Circumventing this issue, DisCoCirc [28] introduces a different way of representing the meaning of a sentence, where certain words are seen as quantum gates that act as operators on input states representing other words. The DisCoCirc approach uses quantum machine learning algorithms [16] for NLP [29, 83] where circuit parameters, related to word representations, are then learned by classical optimization and used to predict different binary labels statistically, such as the answers to *yes-no* questions [84], topics of phrases, or the distinction between subject and object relative clauses [77].

Although these implementations can play an important role in speeding up NLP tasks based on current machine-learning ideas and techniques, they do not go beyond the current paradigm in terms of classification tasks.

Furthermore, a number of theoretical advances using the tensor contractions from DisCoCat cannot be directly reproduced, since the mapping from a phrase to a circuit requires extra steps that deviate from the original grammatical foundation, not treating every word as an input at the same level. We refer here to the work done in expanding the toolbox of word representations with density matrices [109], so as to achieve good results on discerning different word and phrase senses [9, 85, 119], and in entertaining simultaneously different possible interpretations of texts, either by looking at an incremental interpretation of the parsing process [128], or by considering a single representation for the multiple readings of syntactic ambiguities [35, 36]. This presents a strong incentive to find an alternative quantum-circuit implementation that sticks to the original grammatical formulation, preserving the previous achievements, where all words are taken as input on an equal footing. In addition, it is our belief that a quantum framework can contribute a great deal to the reestablishment of rule-based NLP, as a desirable alternative to large-scale statistical approaches [13], since certain computations become more efficient if we use the appropriate quantum algorithms, as we will illustrate in the case of question-answering where quadratic quantum speedup can be achieved.

The paper is structured as follows. In Sec. 5.2 we develop the grammatical framework and quantum state interpretation thereof, setting the stage for the types of linguistic problems we will deal with here. Here we introduce the idea that words are represented as vectors, matrices, and higher-rank tensors, depending on their grammatical function, that contract with each other following grammatical rules, explaining how we can arrive at the interpretations of larger fragments of text. In Sec. 5.3 we put forward an approach where the words are interpreted as quantum states, and we show how the contractions between word representations can be implemented on a quantum computer as the measurement of a permutation operator. We elaborate on how this setting permits the simultaneous treatment of ambiguous phrases in English. In Sec. 5.4 we apply Grover's algorithm to question-answering, using the framework developed in the previous section to turn the representation of the question and answers into the input of the algorithm, together with an oracle that identifies that correct answers.

Finally, in Sec. 5.5 we give an overview of the developments introduced and discuss further work.

5.2 SYNTAX-SEMANTICS INTERFACE

In this section we introduce the grammatical framework that we will be working with. It consists of a categorial grammar as the syntactic front end, together with a compositional mapping that sends the types and derivations of the syntax to a vector-based distributional interpretation. This is necessary to understand the type of linguistic problems that we can address and how they can be solved using a quantum circuit.

5.2.1 *Type logic as syntax*

The key idea of categorial grammar formalisms is to replace the parts of speech of traditional grammars (nouns, adjectives, (in)transitive verbs, etc.) by logical formulas or types; a deductive system for these type formulas then determines their valid combinations. The idea can be traced back to Ajdukiewicz [5], but Lambek's Syntactic Calculus [68] is the first full-fledged formulation of a categorial type logic that provides an algorithm to effectively decide whether a phrase is syntactically well formed or not.

Let us briefly discuss types and their combinatorics. We start from a small set of primitive types, for example s for declarative sentences, n for noun phrases, w for open-ended interrogative sentences, etc. From these primitive types, compound types are then built with the aid of three operations: multiplication $\bullet$, left division $\backslash$ and right division $/$. Intuitively, a type $A \bullet B$ stands for the concatenation of a phrase of type A and a phrase of type B ("A and then B"). Concatenation is not commutative ("A and then B" $\neq$ "B and then A"). Hence we have left *vs* right division matching the multiplication: $A \backslash B$ can be read as "give me a phrase A to the left, and I'll return a phrase B"; B / A is to be interpreted as "give me a phrase A to the right, and I'll return a phrase B". We can codify this informal interpretation in the rules below, where $A_1 \bullet \cdots \bullet A_n \vdash B$ means that from

the concatenation of phrases of type $A_1, \ldots, A_n$ one can derive a phrase of type B. Hence,

$$B/A \bullet A \vdash B, \tag{5.1}$$

$$A \bullet A\backslash B \vdash B. \tag{5.2}$$

As examples of simple declarative sentences, consider *Alice talks*, or *Bob listens*. In the former case, we assign the type n to *Alice* and the type $n\backslash s$ to the intransitive verb *talks*. We start by multiplying the word types in the order by which the words appear, forming $n \bullet n\backslash s$. Then, it suffices to apply rule (5.2), with $A = n$ and $B = s$, to show that $n \bullet n\backslash s$ derives s, i.e. constitutes a well-formed sentence. Conversely, the lack of a derivation of s from $n\backslash s \bullet n$ (*talks Alice*) allows us to conclude that this not a well-formed sentence. These, and the later examples, illustrate only the simplest ways of combining types, but these will suffice for the purposes of this paper. To obtain a deductive system that is sound and complete with respect to the intended interpretation of the type-forming operations, Lambek's Syntactic Calculus also includes rules that allow one to infer $A \vdash C/B$ and $B \vdash A\backslash C$ from $A \bullet B \vdash C$. Moreover, to deal with linguistic phenomena that go beyond simple concatenation, Lambek's type logic has been extended in a number of ways that keep the basic mathematical structure intact but provide extra type-forming operations for a finer control over the process of grammatical composition. See Ref. [95] for a survey, and Ref. [36] for a quantum interpretation of such structural control operations.

Syntactic ambiguities

To see rule (5.1) in action, consider adjectives in English. An adjective is expecting a noun to its right, and, once it is composed with a noun, it must derive something that can be used, for instance, as the argument of an intransitive verb, which, as we have seen, is of type n. Thus, an adjective must be of type n/n, and we can use rule (5.1) to prove that, as an example, *rigorous mathematicians* is a well-formed phrase of type n.

For certain phrases, there is more than one way of deriving the target type, with each derivation corresponding to a distinct interpretation. As an example, consider the noun phrase *rigorous mathematicians and physicists*, an ambiguous structure that has already been studied in the context of vector representations in Ref. [35]. Here the conjunction *and* gets the type $(n\backslash n)/n$; for the complete phrase, we want to show that the following judgement holds:

$$n/n \bullet n \bullet (n\backslash n)/n \bullet n \vdash n. \tag{5.3}$$

There are two possible interpretations: a first one, where the adjective *rigorous* has scope over *mathematicians and physicists*, and a second one, where it only has scope over *mathematicians*. Each of these interpretations is connected to a different way of deriving the goal formula n. The first reading is obtained by applying the rules in the following order

$$\underbrace{n/n \bullet n \bullet \overbrace{(n\backslash n)/n \bullet n}^{(5.1)\,\vdash n\backslash n}}_{(5.1)\,\vdash n}, \tag{5.4}$$

while for the second reading the rules apply in a different order as

$$\underbrace{\underbrace{n/n \bullet n}_{(5.1)\,\vdash n} \bullet \underbrace{(n\backslash n)/n \bullet n}_{(5.1)\,\vdash n\backslash n}}_{(5.2)\,\vdash n}. \tag{5.5}$$

Our goal is to treat both readings simultaneously until further information allows us to clarify which of the readings is the intended one.

Question answering

Question answering (Q&A) is one of the most common tasks in NLP [131]. Questions can be close ended, having “yes" or “no" for an answer, or open

ended, starting by “who", “why" or “what", also referred to as *wh*-questions. For P possible answers, it is always possible to turn *wh*-questions into close-ended questions. If we know that either Alice, Bob, Carol or Dave is talking, we can turn “Who talks?" into a series of four questions “Does *[name]* talk?". Thus, for P possible answers, there are P closed-ended questions that we need to check[a]. We would like to find the answer to the open-ended questions directly, without this mapping. Syntactically, *wh*-questions are open-endend interrogative sentences, and as such are assigned their own type w. For a subject question, the type of the word *who* is thus $w/(n\backslash s)$, since, when applied to an intransitive verb using rule (5.2), it derives the interrogative type w.

5.2.2 *Vectors as semantics*

In the context of automatic processing of text, the most widely used form of representing a word is by a unique array of values, referred to as a “word embedding". Seen as vectors, we can cluster or compare them using varied geometric tools [75, 100, 101]. Representing the meanings of words as such is widely known as “distributional semantics" [20]. In earlier work, vector entries were related with how often a word would appear next to other words [117], following the “distributional hypothesis" that states that words that appear in similar contexts are themselves similar [54]. Nowadays, word embeddings are extracted using language models, targeted on the prediction of the most likely next word [89**?**]. This presents a problem for the representation of larger fragments, since they are less likely to appear in a text, making their distributional array rather sparse and thus not particularly meaningful. Larger fragments can nevertheless receive an embedding, but a direct connection with grammatical composition is lost.

a This is a common way of turning Q&A into a classification problem, where each close-ended question gets a binary label, depending on whether the answer is true or false. Binary classification problems are some of the most well established applications of machine learning. After finding a way of representing the question statements, usually as single vectors, a number of these labeled statements is used to predict the labels of the hold-out statements.

To tackle this problem, the authors in Ref. [33] propose that the arrays representing different words depend on their syntactic types, namely having a dimensionality that mirrors their type complexity. This introduces a way of composing the meanings of the individual words that is homomorphic to the syntactic derivations, generating a representation of larger fragments from the representation of smaller ones. For completeness, the mapping between the syntax and the semantics is done using the formalism of vector spaces. Each syntactic type A is mapped to its semantic type via $\lceil A \rceil$. Each semantic type is then interpreted as a vector space, where the particular words are represented. Let there be three basic semantic spaces $\{S, N, I\}$. The simple syntactic types n and s are mapped respectively to $\lceil n \rceil = N$ and $\lceil s \rceil = S$. Each individual word is an element of the semantic space that interprets its syntactic type. For instance, the interpretation of the word *physicists* is now seen as a vector in N, this being the vector space where the distributional information of nouns is stored. Similarly, *Alice talks* is represented by a vector in S, that has as basis elements two orthogonal states corresponding to "true" and "false". The interrogative type w is mapped to $\lceil w \rceil = I \otimes N \otimes I \otimes S$. The vector space I ("index") has basis elements that are in one-to-one correspondence to the nouns that can be used as answers to the interrogative sentence, providing an enumeration of the noun vectors of N. This will be useful later when we need to index the quantum states associated with each possible answer.

The vector spaces that translate the directional and multiplicative types are obtained recursively as

$$\lceil A \backslash B \rceil = \lceil A / B \rceil = \lceil A \bullet B \rceil = \lceil A \rceil \otimes \lceil B \rceil, \tag{5.6}$$

where $\otimes$ forms a tensor product space, inductively starting from $A, B, C \in \{n, s, w\}$. Note that the tensor is commutative, such that $\lceil A \rceil \otimes \lceil B \rceil \cong \lceil B \rceil \otimes \lceil A \rceil$. We perform tensor contractions as the interpretations of the rules in Eqs. (5.1) and (5.2). Thus, an intransitive verb is represented as a matrix in $N \otimes S$, that when acting on a vector of type N returns a vector in S. Using the notation $[\![.]\!]$ to represent the tensor interpretation of a word, and assuming an orthogonal basis $\{\hat{n}_i\}$ of N and an orthogonal basis $\{\hat{s}_i\}$ of S, the composition of the vectorial interpretations of *Alice* and *talks* leads

to the interpretation of the entire sentence as a vector in S. The word meanings for this sentence are represented as

$$[\![\text{Alice}]\!] = \sum_p A_p \ \hat{n}_p \tag{5.7}$$

$$[\![\text{talks}]\!] = \sum_{qr} t_{qr} \ \hat{n}_q \otimes \hat{s}_r, \tag{5.8}$$

and the full sentence meaning as

$$[\![\text{Alice talks}]\!] = [\![\text{Alice}]\!] \cdot [\![\text{talks}]\!] = \sum_{pr} A_p t_{pr} \hat{s}_r. \tag{5.9}$$

A more refined treatment of the translation from the Lambek types to tensor spaces has been given in Ref. [35].

Similarly, the semantic space for an adjective can be seen as a matrix in $N \otimes N$. Note that here the analogy between a matrix modifying a vector and the adjective as a noun modifier is the clearest. Let us look at the meanings of *rigorous* and *mathematicians*, which can be represented as

$$[\![\text{rigorous}]\!] = \sum_{ij} r_{ij} \ \hat{n}_i \otimes \hat{n}_j \tag{5.10}$$

$$[\![\text{mathematicians}]\!] = \sum_k m_k \ \hat{n}_k. \tag{5.11}$$

The meaning of *rigorous mathematicians* will be given by the application of the translation of rule (5.1) to tensors. At the components level, it is the matrix multiplication between the *rigorous* matrix and the *mathematicians* vector, which gives, consistently with n being the syntactic type of this fragment, a vector in N, as

$$\begin{aligned}&[\![\text{rigorous mathematicians}]\!]\\&= [\![\text{rigorous}]\!].[\![\text{mathematicians}]\!] = \sum_{ij} r_{ij}\, m_j\, \hat{n}_i. \end{aligned} \tag{5.12}$$

The different order of application of Lambek rules in Eqs. (5.4) and (5.5) translates into different vectors that represent the two readings of *rigorous mathematicians and physicists*. The words *and* and *physicists* are given the vector representations

$$[\![\text{and}]\!] = \sum_{lmn} a_{lmn}\, \hat{n}_l \otimes \hat{n}_m \otimes \hat{n}_n, \tag{5.13}$$

$$[\![\text{physicists}]\!] = \sum_{o} p_o\, \hat{n}_o. \tag{5.14}$$

The reading from Eq. (5.4) is represented by the vector

$$[\![\text{rigorous mathematicians and physicists}]\!]_1 = \sum_{ijln} r_{ij}\, m_l\, a_{ljn}\, p_n\, \hat{n}_i, \tag{5.15}$$

whereas the reading from Eq. (5.5) is encoded in the vector

$$[\![\text{rigorous mathematicians and physicists}]\!]_2 = \sum_{jlmn} r_{lj}\, m_j\, a_{lmn}\, p_n\, \hat{n}_m, \tag{5.16}$$

which are of the same form as the results in Ref. [35].

For interrogative sentences, the word *who* will have the semantic function of "lifting" an intransitive verb with representation in space $N \otimes S$ to a representation in $I \otimes N \otimes I \otimes S$, since

$$\begin{aligned}&\lceil w/(n\backslash s)\rceil = I \otimes N \otimes I \otimes S \otimes N \otimes S\\&\cong I \otimes N \otimes I \otimes S \otimes S \otimes N. \end{aligned} \tag{5.17}$$

An element of this space contracts with an element of the representation space of intransitive verbs, $N \otimes S$, associating the index of every possible answer, in I, with both its representation in N and its truth value in S.

5.3 IMPLEMENTATION

In this section we motivate a passage from vectors to quantum states and we introduce them as inputs of quantum circuits that calculate contractions between word representations.

5.3.1 *Quantum states as inputs of a quantum circuit*

We now switch to a representation of word embeddings as vectors in complex-valued inner product vector spaces, i.e. Hilbert spaces. Our atomic semantic spaces N, S and I will now be replaced by their quantum counterparts as the interpretation spaces. We thus have the Hilbert spaces $\mathcal{H}^N$, $\mathcal{H}^S$ and $\mathcal{H}^{\otimes p}$ respectively, with $\mathcal{H}^{\otimes p}$ the p-qubit Hilbert space corresponding to the complex-valued realization of the semantic type I, where we assume that $P = 2^p$. For instance, with $\{|n_i\rangle\}$ the basis of $\mathcal{H}^N$, we now have

$$[\![\text{Alice}]\!] = |\text{Alice}\rangle = \sum_p A_p \ |n_p\rangle \,. \tag{5.18}$$

Note that this space allows us to expand our representations with complex-valued entries, and a proper contraction between the words will require the conjugation of some of the components, i.e.

$$[\![\text{Alice}]\!]^* = \langle \text{Alice}| = \sum_p A_p^* \ \langle n_p| \,. \tag{5.19}$$

Let the input of a circuit be the product of the states that interpret each word in the language fragment in question. Our running example of a

noun subject and an intransitive verb *Alice talks* is now represented as the input

$$|\text{Alice}\rangle\,|\text{talks}\rangle \in \mathcal{H}^N \otimes \mathcal{H}^N \otimes \mathcal{H}^S.$$

The basis of $\mathcal{H}^S$, $\{|s_i\rangle\}$, are the single-qubit spin states $|0\rangle$ and $|1\rangle$, where the former represents a sentence that is false, and the later one that is true. In this setting, it is also possible to establish a probability distribution over the truthfullness of a sentence.

Each of the elements of the interpreting spaces will be represented by a labeled quantum wire, thus rewriting the input state as

$$|\text{Alice}\rangle\,|\text{talks}\rangle \in \mathcal{H}^1_N \otimes \mathcal{H}^2_N \otimes \mathcal{H}^3_S, \tag{5.20}$$

used as the input of a quantum circuit, as shown in Fig. 5.1.

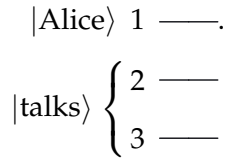


Figure 5.1: Quantum circuit with intransitive sentence input.

The ambiguous fragment *rigorous mathematicians and physicists* will be initially represented as a unique state in the tensor space, formed by numbered copies of the $\mathcal{H}^N$ space as

$$\begin{aligned}&|\text{rigorous}\rangle\,|\text{physicists}\rangle\,|\text{and}\rangle\,|\text{mathematicians}\rangle\\ &\in \mathcal{H}^N_1 \otimes \mathcal{H}^N_2 \otimes \mathcal{H}^N_3 \otimes \mathcal{H}^N_4 \otimes \mathcal{H}^N_5 \otimes \mathcal{H}^N_6 \otimes \mathcal{H}^N_7,\end{aligned}$$

forming the input of a quantum circuit as in Fig. 5.2.

From here, different contractions, corresponding to the two possible readings, will be represented by different circuits acting on this same input, as we show in detail below.

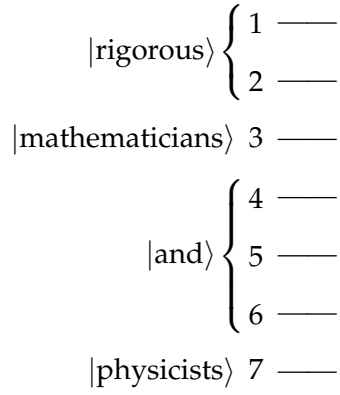


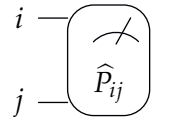
Figure 5.2: Quantum circuit with syntactically ambiguous input.

5.3.2 *Contraction as measurement of permutation operator*

To compute the desired contraction using the quantum circuit, we calculate the expectation value of the permutation operator $\widehat{P}_{ij}$ on the input states/wires indexed by i, j. These correspond to the spaces with elements that we want to contract, following the syntactic rules. For two states $|\phi_1\rangle_i$ and $|\phi_2\rangle_j$ belonging to two numbered copies of a Hilbert space, respectively $\mathcal{H}_i^{\lceil A \rceil}$ and $\mathcal{H}_j^{\lceil A \rceil}$, we refer to the following quantity as the *measurement of the expectation value of the permutation operator*:

$$\begin{aligned} &\langle\phi_1|_i \langle\phi_2|_j \widehat{P}_{ij} |\phi_1\rangle_i |\phi_2\rangle_j = \\ &= \langle\phi_1|\phi_2\rangle_i \langle\phi_2|\phi_1\rangle_j \equiv |\langle\phi_1|\phi_2\rangle|^2. \end{aligned} \tag{5.21}$$

In general, to obtain this quantity on a quantum circuit, one must perform repeated measurements of the input states $|\phi_1\rangle_i$ and $|\phi_2\rangle_j$, on a basis that diagonalizes the permutation operator, summing the frequency of outcomes, using the respective operator's eigenvalues as weights. We introduce the following circuit notation to indicate the measurement of the permutation operator:



If the measuring device is only capable of performing measurements in the standard basis, then we must additionally apply to the input states the inverse of the transformation that diagonalizes the permutation operator, before performing the repeated measurements. In appendix 5.A we show how this can be achieved using the inverse transformation to the Bell basis in the case of two-qubit inputs. In this case, the measurement of the expectation value can be understood as the map between the SWAP operator, that represents the permutation operator in that case, and the projection operator on the maximally entangled state $|\beta_{00}\rangle$, which, although not a homomorphism, will be diagonal in the same basis, since both operators share an algebra.

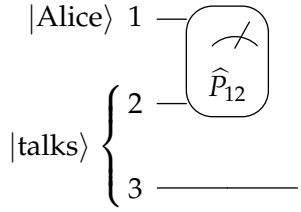


Figure 5.3: Quantum circuit that measures the permutation operator $\widehat{P}_{12}$ on an intransitive sentence input.

We now show in two ways that the final representation of a simple sentence such as *Alice talks* is stored as an effective state $|\psi\rangle$ in $\mathcal{H}_3^S$, after measuring $\widehat{P}_{12}$ not normalizing for clarity, with input as given in Eq. (5.20).

USING OPERATORS. Assume that an operator $\widehat{O}_3$ is being measured in space $\mathcal{H}_3^S$. Its expectation value is given by $\langle\psi|\,\widehat{O}_3\,|\psi\rangle$, after measuring $\widehat{P}_{12}$, with

$$\begin{aligned}&\langle \text{Alice}| \langle \text{talks}| \, \widehat{P}_{12} \otimes \widehat{O}_3 \, |\text{Alice}\rangle \, |\text{talks}\rangle \\ &\equiv \langle \psi | \, \widehat{O}_3 \, |\psi\rangle \, . \end{aligned} \tag{5.22}$$

The left-hand side unfolds as follows:

$$\begin{aligned}&\langle \text{Alice}| \langle \text{talks}| \;\; \widehat{P}_{12} \otimes \widehat{O}_3 \;\; |\text{Alice}\rangle \, |\text{talks}\rangle \, = \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \, \langle n_p n_q s_r | \;\; \widehat{P}_{12} \otimes \widehat{O}_3 \\ &= \; A_{p'} t_{q'r'} \, |n_{p'} n_{q'} s_{r'}\rangle \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \, \langle n_p n_q s_r | \;\; \widehat{O}_3 \;\, A_{p'} t_{q'r'} \, |n_{q'} n_{p'} s_{r'}\rangle \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \, \langle s_r | \;\; \widehat{O}_3 \;\, A_{p'} t_{q'r'} \, |s_{r'}\rangle \, \delta_{pq'} \delta_{qp'} \\ &= \sum_{qr,q'r'} A_q t_{qr}^* \, \langle s_r | \;\; \widehat{O}_3 \;\, A_{q'}^* t_{q'r'} \, |s_{r'}\rangle \, . \end{aligned} \tag{5.23}$$

To uniquely determine $|\psi\rangle$, we need to solve $m = d(d-1)/2$ independent equations, where d is the dimension of $\mathcal{H}^3$, of the form below

$$\sum_{qr,q'r'} A_q t_{qr}^* \, \langle s_r | \;\; \widehat{O}_3 \;\, A_{q'}^* t_{q'r'} \, |s_{r'}\rangle = \langle \psi | \, \widehat{O}_3 \, |\psi\rangle \, . \tag{5.24}$$

Any operator $\widehat{O}_3$ can be decomposed as a sum of m linearly independent operators $\widehat{O}_3^a$, with $1 < a < m$. Since Eq. (5.24) holds for any operator, then it holds for each $\widehat{O}_3^a$, thus generating m independent equations, necessary and sufficient to solve for $|\psi\rangle$. In particular, if $|\psi\rangle$ is expressed in the basis $|s_{r'}\rangle$, the components of $|\psi\rangle$ are given precisely by the respective components of the left-hand side of Eq. (5.24), from which we can immediately conclude that the effective state in $\mathcal{H}_S^3$ is

$$|\psi\rangle = \sum_{q'r'} A_{q'}^* t_{q'r'} \, |s_{r'}\rangle \equiv [\![\text{Alice talks}]\!]. \qquad \blacksquare \tag{5.25}$$

Similarly, density matrices can be used to confirm not only that the state in $\mathcal{H}_3^S$ corresponds to Eq. (5.25) after the measurement, using partial tracing, but also that the outcome of this operation is a pure state.

USING DENSITY MATRICES. Assume that the sentence *Alice talks* is being represented by the pure state density matrix

$$\widehat{\rho} = |\text{Alice}\rangle\,|\text{talks}\rangle\,\langle\text{Alice}|\,\langle\text{talks}|\,. \tag{5.26}$$

We want to show that the density matrix $\widehat{\rho}_3$ that we obtain in space $\mathcal{H}_3^S$ after the measurement of $\widehat{P}_{12}$ is in fact a pure state. We do that by taking the partial trace in spaces 1 and 2 of $\widehat{P}_{12}\widehat{\rho}$:

$$\begin{aligned}
\widehat{\rho}_3 &= \text{Tr}_{12}\left(\widehat{P}_{12}\widehat{\rho}\right) = \\
&= \sum_{ab}\sum_{pqr,p'q'r'} \langle n_a n_b|\,\widehat{P}_{12} A_{p'} t_{q'r'}\,|n_{p'} n_{q'} s_{r'}\rangle\,\langle n_p n_q s_r|\,A^*_p t^*_{qr}\,|n_a n_b\rangle \\
&= \sum_{ab}\sum_{pqr,p'q'r'} \langle n_a n_b|\,A_{p'} t_{q'r'}\,|n_{q'} n_{p'} s_{r'}\rangle\,\langle n_p n_q s_r|\,A^*_p t^*_{qr}\,|n_a n_b\rangle \\
&= \sum_{r,p'q'r'} A_{p'} t_{q'r'}\,|s_{r'}\rangle\,\langle s_r|\,A^*_{q'} t^*_{p'r} \\
&= \sum_{q'r'} A^*_{q'} t_{q'r'}\,|s_{r'}\rangle \sum_{p'r} \langle s_r|\,A_{p'} t^*_{p'r} = |\psi\rangle\,\langle\psi|\,.
\end{aligned} \tag{5.27}$$

This thus proves that the resulting state in space $\mathcal{H}_3^S$ is pure and equal to Eq. 5.25. It also proves that $\langle\psi|\psi\rangle = \langle\widehat{P}_{12}\rangle$, as expected from $\langle\psi|\,\widehat{O}\,|\psi\rangle = \langle\widehat{O}\rangle\langle\widehat{P}_{12}\rangle$. ■

Note that here the index contraction is equivalent to that of Eq. (5.9), enhanced with the conjugation of some components, which remain as an informative feature from the directionality of language, which would be lost otherwise. The circuit that calculates Eq. (5.25) is given in Fig. 5.3.

An alternative way of contracting the two-qubit spaces has been proposed in Ref. [29], where the Bell effect $\langle\beta_{00}| \equiv (\langle 00| + \langle 11|)/\sqrt{2} \in \mathcal{H}_N^1 \otimes \mathcal{H}_N^2$ is measured instead as

$$
\begin{aligned}
&[\![\text{Alice talks}]\!] = \langle \beta_{00} | \text{Alice} \rangle \, | \text{talks} \rangle \\
=&\frac{1}{\sqrt{2}} \sum_{p'q'r'} \left(\langle 00| + \langle 11| \right) A_{p'} t_{q'r'} \left| n_{p'} n_{q'} s_{r'} \right\rangle \\
=&\frac{1}{\sqrt{2}} \sum_{r'} \left(A_0 t_{0r'} + A_1 t_{1r'} \right) | s_{r'} \rangle \, .
\end{aligned} \tag{5.28}
$$

Measuring the permutation operator as we do in Eq. (5.23) is manifestly a more general way of contracting the representations of words than what is done in Eq. (5.28). On the one hand, it allows each interpretation space to have more than two basis states, that is, each quantum wire can represent something more general than one qubit. On the other hand, it accommodates correctly the existence of complex numbers in the quantum mechanical representations. Importantly, it has also one more important feature that we will make use of now: it allows us to integrate a quantum superposition of conflicting readings.

5.3.3 *Ambiguous readings on a quantum circuit*

The quantum states of the two readings in Eqs. (5.15) and (5.16), that result from contracting the individual word states, can be written as

$$
[\![\text{rigorous mathematicians and physicists}]\!]_1 = \sum_{ijln} r_{ij} \, m_l \, a^*_{ljn} \, p^*_n \;\; | n_i \rangle \tag{5.29}
$$

and

$$
[\![\text{rigorous mathematicians and physicists}]\!]_2 = \sum_{jlmn} r^*_{lj} \, m_j \, a_{lmn} \, p^*_n \;\; | n_m \rangle \, . \tag{5.30}
$$

These can be represented by the two different circuits in Figs. 5.4 and 5.5 respectively, coming from the two different contraction schemes, as obtained in Ref. [35]. Also in this reference, an analysis of how to express

the two readings syntactic ambiguities simultaneously is developed, which we here implement. We can go from the first reading to the second by applying two wire swappings. First, we swap wires 3 and 5 on the second reading, which turns the measurement of $\widehat{P}_{23}$ into the measurement of $\widehat{P}_{25}$. Next, by swapping wires 5 (which now contains the information from wire 3) and 1, we effectively turn the measurement of $\widehat{P}_{14}$ into the measurement of $\widehat{P}_{34}$. In this way, the circuit in Fig. 5.6 is equivalent to that of the first reading. If we control the application of this set of swap gates on an extra qubit $|c\rangle = c_1 |1\rangle + c_2 |0\rangle$, we entangle the states of this qubit with the two possible readings. The first reading is stored in the quantum wire 5 with probability $|c_1|^2$, while the second reading is stored in that same quantum wire with probability $|c_2|^2$. In total we have what is represented in the circuit of Fig. 5.7.

The innovation that this implementation brings is that we are now able to deal with both interpretations simultaneously, that later contractions, with the representations of other words, have the potential to disambiguate.

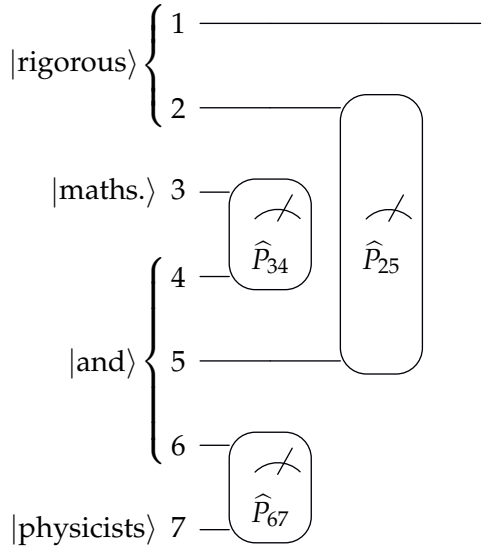


Figure 5.4: Quantum circuit for the first interpretation of the syntactically ambiguous phrase.

5.4 APPLICATION

In this section we apply Grover's quantum search algorithm to obtain the answer to a *wh*-question with quantum speedup, using the quantum circuits for sentence representation developed in the previous section.

5.4.1 *Grover's quantum search algorithm*

Grover's quantum search algorithm aims at finding the correct answer to a query, by taking a state with an equal superposition of orthogonal states representing the answers as the input, and outputting a state in which the only basis states that have any probability of being measured correspond to correct answers. For $P = 2^p$ possible solutions, the first step is to generate a linear superposition of P unique states, with its index x corresponding to one of the possible solutions. In the original proposal [53], this input state

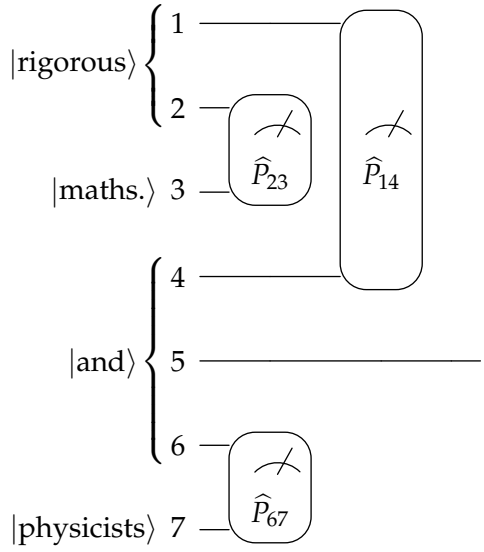


Figure 5.5: Quantum circuit for the second reading of the syntactically ambiguous phrase.

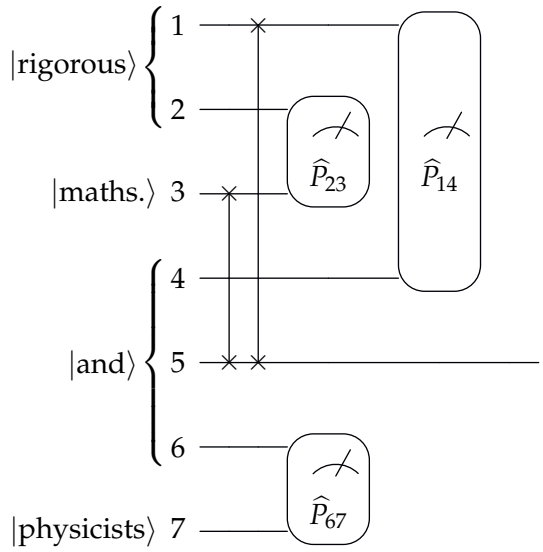


Figure 5.6: Quantum circuit that computes the first reading from the contractions of the second, and is therefore equivalent to Fig. 5.4.

is obtained by acting on $|0\rangle^{\otimes p}$ qubit states with the $H^{\otimes p}$ gate, where H is the one-qubit Hadamard gate, which generates

$$|\Psi\rangle = \frac{1}{\sqrt{P}} \sum_{x=0}^{P-1} |x\rangle \,. \tag{5.31}$$

Then, a sequence of gates, the *Grover iteration*, is repeatedly applied to this input state, until a correct answer is the guaranteed outcome of the measurement of the initial qubits. For Q correct solutions, only $O(\sqrt{P/Q})$ iterations are necessary, representing a quadratic speedup compared to a classical search algorithm, which requires checking all P possible answers. Each Grover iteration G has two main components: first, an *oracle* operation O, and then an *inversion about the mean* operation, formed by applying the unitary transformation that generates $|\Psi\rangle$ to $2\,|0\rangle\,\langle 0| - 1$, in this case

$$H^{\otimes p}(2\,|0\rangle\,\langle 0| - 1)H^{\otimes p} \equiv 2\,|\Psi\rangle\,\langle\Psi| - 1, \tag{5.32}$$

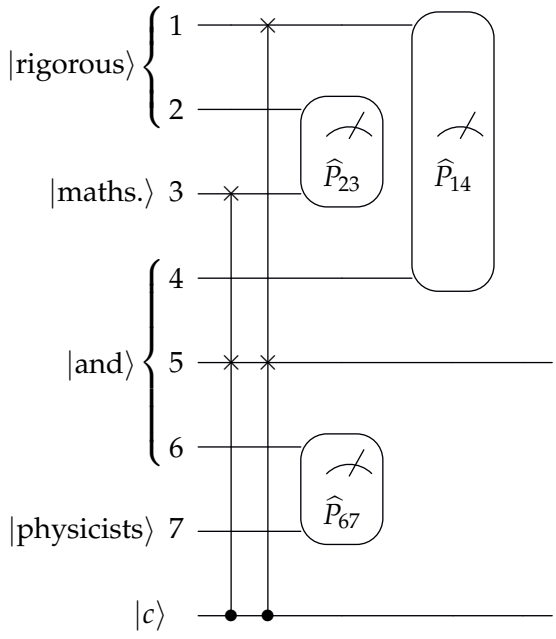


Figure 5.7: Quantum circuit that computes simultaneously the two readings from the ambiguous input.

which can easily be shown to be unitary. The heart of the algorithm is the oracle, as it is able to distinguish the answers that are correct from those that are not. It is a unitary operation that works by flipping the sign of the answer state if $|x\rangle$ is correct, that is,

$$\begin{cases} O(|x\rangle) = -|x\rangle & \text{if } |x\rangle \text{ is a correct answer,} \\ O(|x\rangle) = |x\rangle & \text{otherwise.} \end{cases}$$

To achieve this, more qubits might be necessary, and those constitute the "oracle workspace". In the original setup, it is the oracle that depends on the query at hand, as well as the form of the inputs, while the inverse operation has a universal form. By representing our *wh*-question query as quantum states and contractions therein, we will see that we can use Grover's algorithm with the problem-dependence of its parts reversed:

instead it is the oracle that is universal, and the rotation is query-dependent, obtained from a unitary transformation on $|0\rangle$.

5.4.2 *Input-state preparation for question answering*

The question statement and possible answers hold the key for the search algorithm to identify the correct answers in our application. This will happen as a consequence of the contractions of the possible solutions with the question predicate. We will use our previous construction as the input of the first Grover iteration. To this end, suppose that we want to know the answer to the question *Who talks?*, and that we have P possible answers, of which Q are absolutely correct, and $P - Q$ are, on the contrary, definitely wrong. For the oracle to identify the correct answers, they must be produced from the contraction with the verb, and they must be in a superposition equivalent to Eq. (5.31).

The more complex mapping of w to the semantics, when compared with the syntactic types s and n, can be attributed to the particular semantics of questions and our application. In standard terms, the meaning of a question is taken as the map that sends answers, which belong to the interpretation space of nouns, to truth values, which are elements of the interpretation space of declarative sentences. We want to keep track of which word provides a correct answer in our quantum circuit, and a map like the latter, upon performing contractions, would only give us a count of how many correct and wrong answers there are. To see this, suppose that the word *who* is represented in the space

$$\lceil w/(n\backslash s)\rceil = \mathcal{H}_1^N \otimes \mathcal{H}_2^S \otimes \mathcal{H}_3^S \otimes \mathcal{H}_4^N,$$

and semantic representation as

$$\begin{aligned} |\text{who}\rangle &= \sum_{ab,ij,kl} |n_i\rangle_1 \left|s_j\right\rangle_2 |s_k\rangle_3 |n_l\rangle_4 \, \delta_{il}\delta_{jk} \\ &= \sum_{i,l} |n_i\rangle_1 \left|s_j\right\rangle_2 \left|s_j\right\rangle_3 |n_i\rangle_4 \,. \end{aligned} \tag{5.33}$$

and the intransitive verb *talks*

$$|\text{talks}\rangle = \sum_{mn} t_{mn} \ |n_m\rangle_5 \, |s_n\rangle_6 \, . \tag{5.34}$$

Following the contraction schemes presented in Sec. 5.3.2, the contraction between these two words states results in the state

$$|\text{who}\rangle \, |\text{talks}\rangle = \sum_{ij} t_{ij} \ |n_i\rangle_1 \, |s_j\rangle_2 \, , \tag{5.35}$$

and representing the answers as

$$|\text{answers}\rangle = \sum_{p} W_p \, |n_p\rangle_7 \, , \tag{5.36}$$

their final contraction results in

$$|\text{who}\rangle \, |\text{talks}\rangle \, |\text{answers}\rangle = \sum_{ij} W_i^* t_{ij}^* \ |s_j\rangle_2 \, . \tag{5.37}$$

This shows that a contraction just on the S and N spaces gives only a count of how many correct or incorrect answers there are, but not which ones are which.

As such, the map of the *wh*-word needs to be furthermore tensored with elements of $\mathcal{H}^{\otimes p}$, of which each of the basis elements corresponds to the unique indexing of the possible answers. This provides an entanglement between the distributional representation of a noun, its corresponding truth value and an enumerable representation in the quantum circuit. The word textitwho thus belongs to the following semantic space, in the image of Eq. (5.17),

$$\lceil w/(n\backslash s) \rceil = \mathcal{H}_1^{\otimes p} \otimes \mathcal{H}_2^N \otimes \mathcal{H}_3^{\otimes p} \otimes \mathcal{H}_4^S \otimes \mathcal{H}_5^S \otimes \mathcal{H}_6^N ,$$

with semantic representation given as

$$
\begin{aligned}
|\text{who}\rangle &= \sum_{ab,ij,kl} |a\rangle_1 |n_i\rangle_2 |b\rangle_3 |s_j\rangle_4 |s_k\rangle_5 |n_l\rangle_6 \delta_{ab}\delta_{il}\delta_{jk} \\
&= \sum_{a,i,l} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 |s_j\rangle_5 |n_i\rangle_6 . \qquad (5.38)
\end{aligned}
$$

The intransitive verb *talks* has the same representation as before, now with the adapted labeling of wires

$$
|\text{talks}\rangle = \sum_{mn} t_{mn} \ |n_m\rangle_7 |s_n\rangle_8 . \qquad (5.39)
$$

For clarity, we flesh out the computation of the contractions that involve the extra index space. The semantic contraction of *who* with *talks*, following the interpretation of the syntactic contraction, results in

$$
\begin{aligned}
&\langle\text{who}| \langle\text{talks}| \ \widehat{P}_{67} \otimes \widehat{P}_{58} \otimes \widehat{O}_{1234} \ |\text{who}\rangle |\text{talks}\rangle \\
&= \sum_{a'i'j'm'n'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 \langle n_{i'}|_6 t^*_{m'n'} \langle n_{m'}|_7 \langle s_{n'}|_8 \\
&\cdot \widehat{O}_{1234} \sum_{aijmn} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 |s_n\rangle_5 |n_m\rangle_6 t_{mn} |n_i\rangle_7 |s_j\rangle_8 \\
&= \sum_{a'i'j'm'n'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 t^*_{m'n'} \widehat{O}_{1234} \ |a\rangle_1 |n_{m'}\rangle_2 |a\rangle_3 |s_{n'}\rangle_4 t_{i'j'} \\
&= \sum_{a'i'j'} t_{i'j'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 \widehat{O}_{1234} \sum_{ail} t^*_{ij} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 . \qquad (5.40)
\end{aligned}
$$

From this we read off the question representation, rewriting the indices:

$$
[\![\text{who talks}]\!] = \sum_{aij} t^*_{ij} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 . \qquad (5.41)
$$

To be the input of the quantum search algorithm, the full input needs to correspond to an equal superposition of all possible answers. This can be achieved by entangling the distributional representation of the answers with the corresponding index

$$|\text{answers}\rangle = \sum_{bp} W_p^b \, |n_p\rangle_9 \, |b\rangle_{10} \,, \tag{5.42}$$

followed by a contraction with the question representation, which happens strictly at the semantic level. Hence

$$\begin{aligned}
&\langle \text{who talks}| \, \langle \text{answers}| \, \widehat{P}_{29} \otimes \widehat{P}_{1,10} \otimes \widehat{O}_{34} \, |\text{who talks}\rangle \, |\text{answers}\rangle = \\
&\sum_{a'i'l'} t_{i'j'} \, \langle a'|_1 \, \langle n_{i'}|_2 \, \langle a'|_3 \, \langle s_{j'}|_4 \sum_{b'p'} W_{p'}^{*b'} \, \langle n_{p'}|_9 \, \langle b'|_{10} \\
&\cdot \widehat{O}_{34} \sum_{ail} t_{ij}^* \, |b\rangle_1 \, |n_p\rangle_2 \, |a\rangle_3 \, |s_j\rangle_4 \sum_{bp} W_p^b \, |n_i\rangle_9 \, |a\rangle_{10} \\
&= \sum_{a'i'j'} W_{i'}^{a'} t_{i'j'} \, \langle a'|_3 \, \langle s_{j'}|_4 \, \widehat{O}_{34} \sum_{amn} W_i^{*a} t_{ij}^* \, |a\rangle_3 \, |s_j\rangle_4 \,,
\end{aligned} \tag{5.43}$$

such that the effective input state to the Grover's algorithm is

$$|\Psi_{initial}\rangle = \sum_{aij} W_i^{*a} t_{ij}^* \, |a\rangle_3 \, |s_j\rangle_4 \,, \tag{5.44}$$

which represents an entanglement between the indices of possible answers and truth values. This process is represented by the circuit in Fig. 5.8.

5.4.3 *Oracle and inversion*

Grover's algorithm requires a normalized state as initial input. Since the amplitudes in the state in Eq. (5.44) have information about whether a certain answer is correct, they uniquely associate each word indexed by a with one of the basis states $|s_j\rangle$, in such a way that $\sum_i W_i^{*a} t_{ij}^*$ is null if the combination between word index a and truth value j is not correct, and otherwise equal to one. Since every word should be either true or false, that leaves us with precisely P independent and equally summed states. Therefore, if the indices aj are abbreviated by one index x, the normalized state is given as

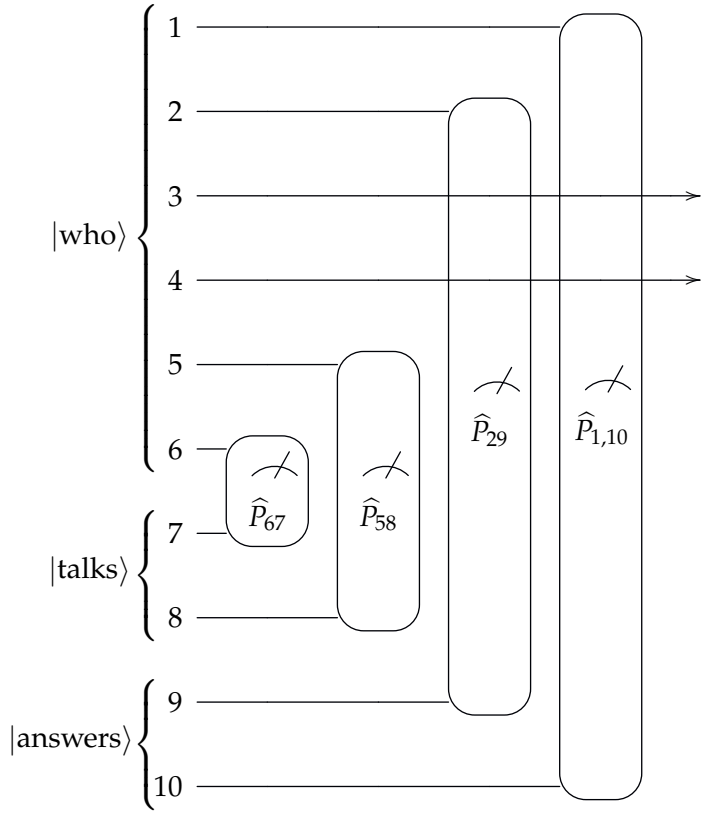


Figure 5.8: Quantum circuit that generates the input of the first Grover iteration for question-answering.

$$|\Psi_{initial}\rangle = \frac{1}{\sqrt{P}} \sum_{x=0}^{P-1} |x\rangle \,, \tag{5.45}$$

with

$$|x\rangle = \sum_i W_i^{*a} t_{ij}^* \, |a\rangle_3 \, |s_j\rangle_4 \,. \tag{5.46}$$

The oracle applied to this input state takes the form of the circuit in Fig. 5.9.

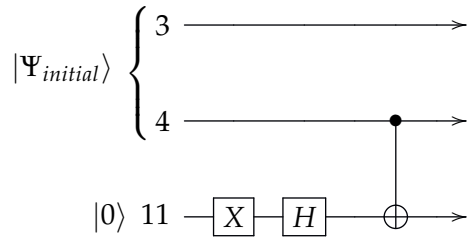


Figure 5.9: Oracle for question-answering.

The states $|a\rangle$ on the first p qubits, being in one-to-one correspondence with each of the possible solutions, are the complete set of states that build up the equal superposition $|\Psi\rangle = H^{\otimes p}|0\rangle^{\otimes p}$, as in Eq. (5.31). In terms of the states that correspond to words that are correct or incorrect, we can rewrite $|\Psi\rangle$ as

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle_3 + \sin\left(\frac{\theta}{2}\right)|\beta\rangle_3\,, \tag{5.47}$$

with $|\alpha\rangle$ the normalized sum of all states that correspond to words that are not solutions, and $|\beta\rangle$ to the normalized sum of those that correspond to words that are solutions. Using this notation, the $|\Psi_{initial}\rangle$ state that we obtain using the contractions can be expressed as

$$|\Psi_{initial}\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle_3|0\rangle_4 + \sin\left(\frac{\theta}{2}\right)|\beta\rangle_3|1\rangle_4\,. \tag{5.48}$$

Though there is entanglement between the first p qubits and the last one, this is a pure state in the $p+1$ qubit space, as it results from the measurement of the permutation operators, as shown in Sec. 5.3.2. As such, there is a unitary transformation that generates it from $|0\rangle^{\otimes p+1}$ as

$$|\Psi_{initial}\rangle = U|0\rangle^{p+1}\,. \tag{5.49}$$

Using this, we can construct the rotation part of the Grover algorithm as

$$U(2\left|0\right\rangle\left\langle 0\right| - 1)U^{\dagger} = 2\left|\Psi_{initial}\right\rangle\left\langle\Psi_{initial}\right| - 1. \tag{5.50}$$

It follows that the Grover iteration applied on the input state in Eq. (5.48) using the oracle and the rotation in Eq. (5.50) gives the desired outcome of

$$\begin{aligned}&(2\left|\Psi_{initial}\right\rangle\left\langle\Psi_{initial}\right| - 1)O(\left|\Psi_{initial}\right\rangle) = \\ &\cos\left(\frac{3\theta}{2}\right)\left|\alpha\right\rangle_3\left|0\right\rangle_4 + \sin\left(\frac{3\theta}{2}\right)\left|\beta\right\rangle_3\left|1\right\rangle_4.\end{aligned} \tag{5.51}$$

This is precisely what we expect to obtain. For the second iteration, the oracle acts as desired, and so does the rotation. After a number of iterations only the states associated with $\left|1\right\rangle_4$ have positive amplitude, which means that we are certain to measure the index of a word that corresponds to a correct answer when we make a measurement on the first p qubits. Thus we have obtained a correct answer with quadratic speedup due to the quantum search algorithm.

5.5 CONCLUSION AND OUTLOOK

In this paper we introduced two main developments in the application of quantum computation to natural language processing. The first one is a tensor-contraction scheme on quantum circuits. Taking quantum states as the representations of all input words, contractions are then identified with the expectation value of an appropriate permutation operator. Doing this, are we not only able to reproduce previous analytical results, but we also allow for complex values and create quantum circuits that are equipped to deal with the syntactic ambiguities in Ref. [35]. With this setup, each reading of an ambiguous phrase corresponds to a particular circuit, and different readings are interchangeable upon the application of a number of swap gates. Controlling on these swap gates, we can obtain a true quantum superposition of the multiple readings. This covers the problem of how to deal with multiple readings in real time, without the need to assume any contextualization. While this addresses the question

of syntactic ambiguities by making use of the quantum superposition principle, ambiguities at the word level can be immediately accommodated for by using density matrices [9, 86, 109, 119], instead of the pure states we use here for simplicity. A generalization to other sentence-level ambiguities constitutes further work, in the expectation that the use of different controls allows for different readings simultaneously in the output state. Note that, in terms of a concrete implementation, the permutation between two qubits used to generate an equal superposition of readings from an ambiguous input takes the form of a Fredkin gate, or a CSWAP gate, which might add considerable circuit complexity, but this is expected to be compensated by the fact that the number of two-qubit operations only scales linearly with an increasing number of readings, since for these types of ambiguities all permutations can be generated via sets of SWAP operations. We leave a more robust exploration of these technical constraints to future work.

The second development builds on this quantum framework, and consists of a quantum search algorithm that is able to find the answer to a *wh*-question with quantum speedup. As input, the algorithm takes a multipartite state in quantum superposition, representing a *wh*-question and its possible answers, and performs a series of tensor contractions as established previously. A series of gates then acts repeatedly on the post-contraction state, guaranteeing that a correct answer to the question is obtained upon a single final measurement. Our algorithm takes advantage of intrinsic quantum features to identify and deliver a correct answer with quantum speedup of quadratic order, when compared to the classical alternative of checking every possible answer. We are thus able to provide a correct answer using information given directly by the tensor contractions of representations of words as proposed by DisCoCat, and without needing to hand-feed any labels nor to learn the answers to other questions. Our approach thus shows how quantum circuit implementations can break with the widely accepted "learning paradigm" of current NLP approaches to question-answering and other tasks used in Ref. [84], providing a scalable approach to open-ended questions. Our approach differs from that of Ref. [28] also in the sense that we keep all words as input states, instead of representing words from complex types as gates that modify

circuit inputs, remaining closer to the compositional spirit of the syntax and therefore being more easily extensible to larger language fragments. Further work includes finding an effective implementation of the measurement of the permutation operator for an arbitrary number of qubits, possibly making use of the Hadamard test [4], and understanding how to find a universal form of the inversion operator that does not depend on $|\alpha\rangle$ and $|\beta\rangle$ separately. An extension of the present formalism can furthermore account for a better understanding of the temporal evolution of the meanings of sentences.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for the helpful comments. This work is supported by the Utrecht University's Complex Systems Fund, with special thanks to Peter Koeze, and the D-ITP consortium, a program of the Netherlands Organization for Scientific Research (NWO) that is funded by the Dutch Ministry of Education, Culture and Science (OCW).

5.A MEASURING THE PERMUTATION OPERATOR ON TWO QUBITS

In this appendix we show that for the measurement of the permutation operator applied to two qubits it suffices to measure the input states in the Bell basis. The two input qubits have the four possible joint states in the standard basis, given by

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{5.52}$$

The permutation operator applied to two qubits is equivalent to the SWAP gate S. In this basis, this operator has the matrix representation

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.53}$$

The eigenstates of this operator are the well-known singlet and triplet states that represent the joint spin of two spin-1/2 particles. With eigenvalue -1, we have the singlet state

$$|0,0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \tag{5.54}$$

and with eigenvalue 1 we have the three triplet states

$$|1,-1\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |1,0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\\0\end{pmatrix}, \text{ and } |1,1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}, \tag{5.55}$$

expressed in the standard basis as

$$|0,0\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \tag{5.56}$$

$$|1,-1\rangle = |00\rangle, \tag{5.57}$$

$$|1,0\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{5.58}$$

$$|1,1\rangle = |11\rangle. \tag{5.59}$$

In its turn, the Bell basis can be expressed in terms of the standard basis in the following way

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{5.60}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{5.61}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{5.62}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{5.63}$$

The Bell states can thus be rewritten using the total-spin eigenstates of S, given in (5.56) to (5.59), as:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|1,-1\rangle + |1,1\rangle) \tag{5.64}$$

$$|\beta_{01}\rangle = |1,0\rangle \tag{5.65}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|1,-1\rangle - |1,1\rangle) \tag{5.66}$$

$$|\beta_{11}\rangle = |0,0\rangle\,. \tag{5.67}$$

Because any linear combination of degenerate eigenstates is also an eigenstate of that operator with the same eigenvalue [proof: $A\vec{v} = \lambda\vec{v}, A\vec{w} = \lambda\vec{w} \Rightarrow A(a\vec{v} + b\vec{w}) = \lambda(a\vec{v} + b\vec{w})$], we see that $|\beta_{00}\rangle$, $|\beta_{01}\rangle$ and $|\beta_{10}\rangle$ are eigenstates of S with eigenvalue 1, and $|\beta_{11}\rangle$ is an eigenstate with eigenvalue -1. Therefore, we can conclude that the Bell basis also diagonalizes the permutation operator, and as such repeated measurements of the qubits in this basis allows us to directly compute the expectation value of the operator in the input states. So, for a two-qubit input state

$$|\Phi\rangle = \sum_{ij} a_i b_j \,|ij\rangle\,, \tag{5.68}$$

with $i, j \in \{0,1\}$, the expectation value of the S operator is given by

$$\langle S\rangle_\Phi = |\langle\beta_{00}|\Phi\rangle|^2 + |\langle\beta_{01}|\Phi\rangle|^2 + |\langle\beta_{10}|\Phi\rangle|^2 - |\langle\beta_{11}|\Phi\rangle|^2. \tag{5.69}$$

If we are in the possession of a measuring device that can only measure in the standard basis, we must transform our input states with the inverse transformation that generates the Bell states. This serves to guarantee that an outcome $|ij\rangle$ is in fact as likely as the measurement of $|\beta_{ij}\rangle$ if the input states were measured directly in the Bell basis.

A

NOTES ON CATEGORY THEORY

Category theory is the branch of mathematics formalized by Eilenberg and Saunders Mac Lane in 1945 [42]. It emerged from the field of algebraic topology, where it became apparent that a more general abstract theory was necessary to describe the underlying relationships between different mathematical entities, which were until then understood independently.

The best analogy to motivate the role and importance of category theory comes from group theory. In the same way that the notion of group transformation allowed mathematicians to assign abstract algebraic properties to geometric objects, and thus freed them to focus on the underlying geometric space, so category theory offered a continuation of this idea, by boiling down mathematical objects to their bare minimum properties, rendering them comparable and interchangeable by turning on or off certain properties [81].

A *category* is defined as a mathematical structure comprised by *objects* and *morphisms* (also knows as arrows, or maps) between these objects. It extends the theory of functions, which is then just an instance of the larger theory. This can be easily understood by looking at the category of sets, **Set**, which has sets as objects and total functions as morphisms. Another example is the category of groups, **Grp**, which has groups as objects, and group homomorphisms as morphisms. To fully be defined as a category, it has to be furthermore endowed with a binary operation $\circ$, designated by *composition* of morphisms. If f and g are two category morphisms, then so must be $f \circ g$. The composition has to be associative and there must be, for

every object, an identity morphism that sends it to itself. It is easy to verify that all conditions are satisfied for the categories of sets and groups. As categories become more exotic, this becomes increasingly more difficult to prove.

The main advantage of this abstraction is that we can prove the equivalence of different frameworks in a rigorous way, and compare them according to the properties that they eventually share, thus classifying mathematical structures. For instance, we can inspect what it would take to turn the category of functions to the category of groups, and vice-versa. To this effect, a structure preserving map can be used between categories, called a *functor*. If the functors mapping the categories to each other possess a number of characteristics[a], then the categories can be said to be *equivalent*, or isomorphic. Equivalence of categories turns out to be a powerful analytical tool: if one knows how to prove statements in one category, and another category is equivalent to the first, then the proofs of the first extend to the second.

CURRY-HOWARD-LAMBEK CORRESPONDENCE

As the theory of categories is an abstract one, it can cover any kind of object and morphism, and this includes formal logics. Lambek showed that intuitionistic logics akin to his calculus could be understood in the language of categories, with type formulas as objects, and (the equivalence classes of) derivations as morphisms [70]. Lambek named the corresponding category a *residuated category*, and noted that, within the hierarchy of categories, it is related to a class of categories called *closed categories*. Later, Lambek showed that, not only is a intuitionistic calculus a *cartesian closed category*, but that also a λ-calculus generated on a category of the same kind, that is, where the semantic types have, as domains, objects and morphisms of a category of that type, would itself be a cartesian closed category, making the connection between both of these structures, already known as the Curry-Howard correspondence, more precise through category theory [71],

a The functor mapping a category to an isomorphic category must be full, faithful and dense, with its inverse functor sharing these properties.

forming what is now known as the Curry-Howard-Lambek correspondence. One essential step on the proof is that the functor that takes $A \times B$ to C is *naturally isomorphic* to the functor that takes B to the set of morphisms from A to C.

It turns out that **Set** is a cartesian closed category, and therefore so is the λ-calculus generated on it. This beautifully brings to a full circle the compositional set-theoretic approach to natural language semantics as already introduced by Montague, endowed with a robust recursive grammar in the form of type-driven formal deduction systems, such as is the Lambek calculus.

CATEGORICAL VECTOR SPACES

If we are interested in extending this reasoning to a semantics of vector spaces, we can use the categorical power of abstraction, knowing that vector spaces of the kind that we need to represent words are object of the category **FdVect**, which has finite vector spaces as objects and linear maps as morphisms. It is however not a cartesian closed category, but a compact closed one. This is because, while **Set** is furthermore endowed with a monoidal structure that allows for the generation of a product space, the cartesian product, **FdVect** also allows for the generation of product spaces, which is crucial to this account, but instead by the tensor product between vector spaces, which is not a proper categorical product[b]. It turns out that it can be shown, using dual vector spaces, that a linear map from $U^* \otimes V$ to X is equivalent to the map from V to the set of linear maps from U to X. This justifies the assignment of these tensor product spaces to the domains of our complex semantic types, by using that $D_{A \to B} = A^* \otimes B$.

b The cartesian product is a proper product in the category theory sense. It can also be defined between vector spaces, but it does not produce the desired result. This can be seen from the fact that the tensor product generates a new space with dimensions Dim(V)×Dim(W), while a Cartesian product generates a space with dimensions Dim(V)+Dim(W)[7].

QUANTUM CATEGORIES

Around 2004, Samson Abramsky and Bob Coecke started developing a categorical approach to quantum computing [3]. Their realization was that the spaces where quantum states are defined, namely Hilbert spaces, form a compact closed category. In 2010, Coecke et al. [33] proposed that a specific grammar for natural language, that Lambek developed later in his career, designated by *pregroups*[73], be used. This grammar is less restrictive than the multiplicative fragment of the Lambek calculus, leading to a less fine-grained control of language phenomena. For instance, while in pregroups the statement $(a \bullet b)\backslash c \vdash a/(b \bullet c)$ is derivable, this is not the case in the original Lambek Calculus. Pregroups, however, have the advantage that they too correspond to a compact closed category, and so using it as syntax to a vector space semantics is more direct, allowing us to map the syntax to a semantics of quantum states, where words are interpreted as quantum states, which are vectors in a Hilbert space. In an attempt to reconcile all approaches, the authors of Ref. [30] show that, if the product in the Lambek Calculus is a monoidal tensor, then it can also be shown to be equivalent to a compact closed category.

B

NOTES ON DENSITY MATRICES

In this appendix are some of the rules of density matrix manipulation that we use throughout this thesis. The *inner product* reduces the dimension of the operators. The inner product is a map $A \times A^* \to \mathbb{C}$. It works on the basis vectors as

$$ {}_A\langle i'|i\rangle_A = \delta_{i,i'}. \tag{B.1}$$

If there is a correlation between two or more quantum systems, the tensor product is used to create or represent density matrices and states that belongs to a multipartite quantum system. The basis elements of the composite vector space, which are kets, are given by

$$ |i\rangle_A \otimes |j\rangle_B \equiv |ij\rangle_{A\otimes B} \equiv |k\rangle_C \in C = A \otimes B, \tag{B.2}$$

while the basis elements of the composite dual vector space, bras, are given by

$$ {}_A\langle i'| \otimes {}_B\langle j'| \equiv {}_{A\otimes B}\langle i'j'| \equiv {}_C\langle k'| \in C^* = A^* \otimes B^*. \tag{B.3}$$

The density matrix of a multipartite system can then be defined:

$$ \rho_x^C = \rho_x^{A\otimes B} = \sum_{ii',jj'} X_{ii'jj'} \left(|i\rangle\, {}_A\langle i'|\right) \otimes \left(|j\rangle\, {}_B\langle j'|\right) = \sum_{ii',jj'} X_{ii'jj'} \, |ij\rangle\, {}_C\langle i'j'|, \tag{B.4}$$

with $C = A \otimes B$. While a density matrix can already be describing a composite system, the tensor product of two single-system density matrices also creates a density matrix in the basis of the composite system:

$$\rho_y^C = \rho_w^A \otimes \rho_z^B = \sum_{ii',jj'} W_{ii'} Z_{jj'} \left|ij\right\rangle_C \left\langle i'j'\right|, \tag{B.5}$$

again with $C = A \otimes B$. If the density matrices are described in the same space, matrix multiplication affects them in the following way:

$$\rho_x^A . \rho_y^A = \sum_{ii'} X_{ii'} \left|i\right\rangle_A \left\langle i'\right| \sum_{jj'} Y_{jj'} \left|j\right\rangle_A \left\langle j'\right| = \sum_{ii',jj'} X_{ii'} Y_{jj'} \left|i\right\rangle_A \left\langle j'\right| \delta_{i',j} \tag{B.6}$$

$$= \sum_{ii',j'} X_{ii'} Y_{i'j'} \left|i\right\rangle_A \left\langle j'\right|. \tag{B.7}$$

This product is not commutative in general. If the product is between composite density matrices that share a part in the same subsystem, the components that belong to overlapping spaces are the ones affected in the way described above.

An operation in what follows is the *trace* of a density matrix. In case the trace is not taken over all the spaces of a composite density matrix, it is called the *partial trace*, otherwise it is called the *total trace*. It acts by adding the diagonal elements of the matrix that belongs to that space.

$$\mathrm{Tr}_A \left(\rho_x^A\right) = \mathrm{Tr}_A \left(\sum_{ii'} X_{ii'} \left|i\right\rangle_A \left\langle i'\right| \right) = \sum_j {}_A\left\langle j\right| \sum_{ii'} X_{ii'} \left|i\right\rangle_A \left\langle i'\middle|j\right\rangle_A$$

$$= \sum_{ii',j} X_{ii'} \delta_{ji} \delta_{i'j} = \sum_j X_{jj}.$$

The trace is also a map $A \times A^* \to \mathbb{C}$, so it can be seen as the generalization for matrices of the inner product. The trace is a basis independent quantity and can also be defined as the sum of the eigenvalues of the linear map

represented by the matrix. If the argument of the trace has a product of matrices, the product is evaluated before the trace:

$$\mathrm{Tr}_A\left(\rho_x^A.\rho_y^A\right) = \mathrm{Tr}_A\left(\sum_{ii',j'} X_{ii'}Y_{i'j'}\left|i\right\rangle_A\left\langle j'\right|\right) = \sum_{ii',j'} X_{ii'}Y_{i'j'}\delta_{i,j'} = \sum_{ii'} X_{ii'}Y_{i'i}. \tag{B.8}$$

Another property of the trace is that is *cyclic*:

$$\mathrm{Tr}_A\left(\rho_x^A.\rho_y^A\right) = \mathrm{Tr}_A\left(\rho_y^A.\rho_x^A\right). \tag{B.9}$$

The partial trace for a matrix in a composite space is a trace taken over only some of the spaces it is composed of:

$$\mathrm{Tr}_A\left(\rho_y^C\right) = \mathrm{Tr}_A\left(\rho_y^{A\otimes B}\right) = \mathrm{Tr}_A\left(\sum_{ii',jj'} X_{ii'jj'}\left|ij\right\rangle_{A\otimes B}\left\langle i'j'\right|\right) \tag{B.10}$$

$$= \sum_l {}_A\langle l|\sum_{ii',jj'} X_{ii'jj'}\left(\left|i\right\rangle_A\left\langle i'\right|\right)\otimes\left(\left|j\right\rangle_B\left\langle j'\right|\right)\left|l\right\rangle_A = \sum_{l,jj'} X_{lljj'}\left|j\right\rangle_B\left\langle j'\right|. \tag{B.11}$$

As it will be useful later, the *permutation operation* is now introduced. Permutations act on two quantum systems to swap the state associated to each system. On the states they act as

$$P_{AB}\left|ij\right\rangle_{A\otimes B} = \left|ij\right\rangle_{B\otimes A} \text{ and } {}_{A\otimes\tilde{B}}\left\langle i'j'\right|P_{A\tilde{B}} = {}_{A\otimes B}\left\langle i'j'\right|,$$

such that on the density matrix elements it acts as

$$P_{AB}\left|ij\right\rangle_{A\otimes B}\left\langle i'j'\right|P_{AB} = \left|ij\right\rangle_{B\otimes A}\left\langle i'j'\right| = \left|i\right\rangle_B\left\langle i'\right|\otimes\left|j\right\rangle_A\left\langle j'\right|.$$

BIBLIOGRAPHY

[1] In: van Benthem, Johan (Hrsg.) ; ter Meulen, Alice (Hrsg.): *Handbook of Logic and Language (2nd Edition)*. Elsevier, 2011

[2] Abbaszadeh, Mina ; Mousavi, S S. ; Salari, Vahid: Parametrized Quantum Circuits of Synonymous Sentences in Quantum Natural Language Processing. In: *arXiv:2102.02204* (2021)

[3] Abramsky, Samson ; Coecke, Bob: A categorical semantics of quantum protocols. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.* IEEE (Veranst.), 2004, S. 415–425

[4] Aharonov, Dorit ; Jones, Vaughan ; Landau, Zeph: A polynomial quantum algorithm for approximating the Jones polynomial. In: *Algorithmica* 55 (2009), Nr. 3, S. 395–421

[5] Ajdukiewicz, Kazimierz: Die syntaktische konnexitat. In: *Studia philosophica* (1935), S. 1–27

[6] Arute, Frank ; Arya, Kunal ; Babbush, Ryan ; Bacon, Dave ; Bardin, Joseph C. ; Barends, Rami ; Biswas, Rupak ; Boixo, Sergio ; Brandao, Fernando G. ; Buell, David A. u. a.: Quantum supremacy using a programmable superconducting processor. In: *Nature* 574 (2019), Nr. 7779, S. 505–510

[7] Axler, Sheldon J.: *Linear algebra done right*. Bd. 2. Springer, 1997

[8] Banjade, Rajendra ; Maharjan, Nabin ; Niraula, Nobal B. ; Rus, Vasile ; Gautam, Dipesh: Lemon and tea are not similar: Measuring word-to-word similarity by combining different methods. In: *International conference on intelligent text processing and computational linguistics*, 2015, S. 335–346

[9] Bankova, Dea ; Coecke, Bob ; Lewis, Martha ; Marsden, Dan: Graded hyponymy for compositional distributional semantics. In: *Journal of Language Modelling* 6 (2019), Nr. 2, S. 225–260

[10] Bar-Hillel, Yehoshua: A quasi-arithmetical notation for syntactic description. In: *Language* 29 (1953), Nr. 1, S. 47–58

[11] Baroni, Marco ; Bernardi, Raffaella ; Zamparelli, Roberto: Frege in Space: A Program for Composition Distributional Semantics. In: *Linguistic Issues in Language Technology, Volume 9, 2014-Perspectives on Semantic Representations for Textual Inference*, 2014

[12] Barsalou, Lawrence W.: Context-independent and context-dependent information in concepts. In: *Memory & cognition* 10 (1982), Nr. 1, S. 82–93

[13] Bender, Emily M. ; Gebru, Timnit ; McMillan-Major, Angelina ; Shmitchell, Shmargaret: On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, S. 610–623

[14] Bender, Emily M. ; Koller, Alexander: Climbing towards NLU: On meaning, form, and understanding in the age of data. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, S. 5185–5198

[15] Benthem, Johan van: *The semantics of variety in categorial grammar*. John Benjamins, 1988

[16] Biamonte, Jacob ; Wittek, Peter ; Pancotti, Nicola ; Rebentrost, Patrick ; Wiebe, Nathan ; Lloyd, Seth: Quantum machine learning. In: *Nature* 549 (2017), Nr. 7671, S. 195–202

[17] Bishop, Christopher M.: *Pattern recognition and machine learning*. Springer, 2006

[18] Blacoe, William: Semantic composition inspired by quantum measurement. In: *International Symposium on Quantum Interaction* Springer (Veranst.), 2014, S. 41–53

[19] Bohr, Niels: The spectra of helium and hydrogen. In: *Nature* 92 (1913), Nr. 2295, S. 231–232

[20] Boleda, Gemma: Distributional semantics and linguistic theory. In: *Annual Review of Linguistics* 6 (2020), S. 213–234

[21] Bruni, Elia ; Tran, Nam-Khanh ; Baroni, Marco: Multimodal distributional semantics. In: *Journal of Artificial Intelligence Research* 49 (2014), S. 1–47

[22] Bucaria, Chiara: Lexical and syntactic ambiguity as a source of humor: The case of newspaper headlines. In: *Humor* 17 (2004), Nr. 3, S. 279–309

[23] Cann, Ronnie ; Kempson, Ruth ; Gregoromichelaki, Eleni: *Semantics: An introduction to meaning in language*. Cambridge University Press, 2009

[24] Carnap, Rudolf: On the application of inductive logic. In: *Philosophy and phenomenological research* 8 (1947), Nr. 1, S. 133–148

[25] Chomsky, Noam: *Aspects of the Theory of Syntax*. Bd. 11. 2014

[26] Church, Alonzo: A formulation of the simple theory of types. In: *The journal of symbolic logic* 5 (1940), Nr. 2, S. 56–68

[27] Clark, Stephen: Vector space models of lexical meaning. In: *Handbook of Contemporary Semantics* 10 (2015), S. 9781118882139

[28] Coecke, Bob: The mathematics of text structure. In: *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*. Springer, 2021, S. 181–217

[29] Coecke, Bob ; Felice, Giovanni de ; Meichanetzidis, Konstantinos ; Toumi, Alexis: Foundations for Near-Term Quantum Natural Language Processing. In: *arXiv:2012.03755* (2020)

[30] Coecke, Bob ; Grefenstette, Edward ; Sadrzadeh, Mehrnoosh: Lambek vs. Lambek: Functorial vector space semantics and string

diagrams for Lambek calculus. In: *Annals of pure and applied logic* 164 (2013), Nr. 11, S. 1079–1100

[31] Coecke, Bob ; Meichanetzidis, Konstantinos: Meaning updating of density matrices. In: *Journal of Applied Logics* 7 (2020), Nr. 5

[32] Coecke, Bob ; Moore, David ; Wilce, Alexander: Operational quantum logic: An overview. In: *Current research in operational quantum logic*. Springer, 2000, S. 1–36

[33] Coecke, M Sadrzadeh B. ; Clark, S: Mathematical Foundations for a Compositional Distributed Model of Meaning. In: *Lambek Festschirft ,Linguistic Analysis ,vol. 36* 36 (2010)

[34] Cohen-Tannoudji, Claude ; Diu, Bernard ; Laloë, Franck: *Quantum Mechanics, Volume 1: Basic Concepts, Tools, and Applications*. John Wiley & Sons, 2019

[35] Correia, A. D. ; Moortgat, M. ; Stoof, H.T.C.: Density matrices with metric for derivational ambiguity. In: *Journal of Applied Logics* 7 (2020), Nr. 5, S. 795–822

[36] Correia, AD ; Stoof, HTC ; Moortgat, M: Putting a Spin on Language: A Quantum Interpretation of Unary Connectives for Linguistic Applications, 2021, S. 114–140

[37] Deerwester, Scott ; Dumais, Susan T. ; Furnas, George W. ; Landauer, Thomas K. ; Harshman, Richard: Indexing by latent semantic analysis. In: *Journal of the American society for information science* 41 (1990), Nr. 6, S. 391–407

[38] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '19)*, 2019, S. 4171–4186

[39] Dobó, András ; Csirik, János: A comprehensive study of the parameters in the creation and comparison of feature vectors in distributional

semantic models. In: *Journal of Quantitative Linguistics* 27 (2020), Nr. 3, S. 244–271

[40] DOWTY, David R. ; WALL, Robert E. ; PETERS, Stanley: Introduction to Montague Semantics. (1981)

[41] DULLEMOND, Kees ; PEETERS, Kasper: Introduction to tensor calculus. In: *Kees Dullemond and Kasper Peeters* (1991)

[42] EILENBERG, Samuel ; MACLANE, Saunders: General theory of natural equivalences. In: *Transactions of the American Mathematical Society* 58 (1945), Nr. 2, S. 231–294

[43] FAN, Jung-wei ; YANG, Elly W. ; JIANG, Min ; PRASAD, Rashmi ; LOOMIS, Richard M. ; ZISOOK, Daniel S. ; DENNY, Josh C. ; XU, Hua ; HUANG, Yang: Syntactic parsing of clinical text: guideline and corpus development with handling ill-formed sentences. In: *Journal of the American Medical Informatics Association* 20 (2013), Nr. 6, S. 1168–1177

[44] FINKELSTEIN, Lev ; GABRILOVICH, Evgeniy ; MATIAS, Yossi ; RIVLIN, Ehud ; SOLAN, Zach ; WOLFMAN, Gadi ; RUPPIN, Eytan: Placing search in context: The concept revisited. In: *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, 2001, S. 406–414

[45] FRAUNHOFER, Joseph von: *Kurzer Bericht von den Resultaten neuerer Versuche, über die Gesetze des Lichtes, und die Theorie derselben*. éditeur inconnu, 1823

[46] FURNAS, George W. ; LANDAUER, Thomas K. ; GOMEZ, Louis M. ; DUMAIS, Susan T.: Human factors and behavioral science: Statistical semantics: Analysis of the potential performance of key-word information systems. In: *The Bell System Technical Journal* 62 (1983), Nr. 6, S. 1753–1806

[47] GEORGI, Howard: *Lie algebras in particle physics: from isospin to unified theories*. Taylor & Francis, 2000

[48] Giovannetti, Vittorio ; Lloyd, Seth ; Maccone, Lorenzo: Quantum random access memory. In: *Physical Review Letters* 100 (2008), Nr. 16, S. 160501

[49] Goldstone, Robert L. ; Medin, Douglas L. ; Halberstadt, Jamin: Similarity in context. In: *Memory & Cognition* 25 (1997), Nr. 2, S. 237–255

[50] Greco, Giuseppe ; Liang, Fei ; Moortgat, Michael ; Palmigiano, Alessandra: Vector spaces as Kripke frames. In: *Journal of Applied Logics* 7 (2020), Nr. 5

[51] Grefenstette, Edward ; Sadrzadeh, Mehrnoosh: Experimental support for a categorical compositional distributional model of meaning. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, 2011, S. 1394–1404

[52] Grefenstette, Edward ; Sadrzadeh, Mehrnoosh: Experimenting with transitive verbs in a DisCoCat. In: *Proceedings of the 2011 Workshop on GEometrical Models of Natural Language Semantics (GEMS '11)*, 2011, S. 62–66

[53] Grover, Lov K.: A fast quantum mechanical algorithm for database search. In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 1996, S. 212–219

[54] Harris, Zellig S.: Distributional structure. In: *Word* 10 (1954), Nr. 2-3, S. 146–162

[55] Heijmans, A.: *Wetenschap tussen universiteit en industrie. De experimentele natuurkunde in Utrecht onder W. H. Julius en L. S. Ornstein 1896 1940*, Utrecht U., PhD Thesis, 1994

[56] Heunen, Chris ; Sadrzadeh, Mehrnoosh ; Grefenstette, Edward: *Quantum physics and linguistics: a compositional, diagrammatic discourse*. Oxford University Press, 2013

[57] Hewitt, John ; Manning, Christopher D.: A structural probe for finding syntax in word representations. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '19)*, 2019, S. 4129–4138

[58] Hill, Felix ; Reichart, Roi ; Korhonen, Anna: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. In: *Computational Linguistics* 41 (2015), Nr. 4, S. 665–695

[59] Hu, Jennifer ; Gauthier, Jon ; Qian, Peng ; Wilcox, Ethan ; Levy, Roger: A Systematic Assessment of Syntactic Generalization in Neural Language Models. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, S. 1725–1744

[60] Jacobson, Pauline I.: *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford University Press, 2014

[61] Janssen, Theo M. u. a.: Compositionality: Its historic context. In: *The Oxford handbook of compositionality* (2012), S. 19–46

[62] Janssen, Theo M. ; Partee, Barbara H.: Compositionality. In: Benthem, Johan van (Hrsg.) ; Meulen, Alice ter (Hrsg.): *Handbook of Logic and Language*. 1997, S. 417 – 473

[63] Karamcheti, Siddharth ; Krishna, Ranjay ; Fei-Fei, Li ; Manning, Christopher: Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021, S. 7265–7281

[64] Kogkalidis, Konstantinos ; Moortgat, Michael ; Deoskar, Tejaswini: Constructive Type-Logical Supertagging With Self-Attention Networks. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP '19)*, 2019, S. 113–123

[65] Kulis, Brian u. a.: Metric learning: A survey. In: *Foundations and Trends® in Machine Learning* 5 (2013), Nr. 4, S. 287–364

[66] Kurtonina, N. ; Moortgat, M.: Structural Control. In: Blackburn, P. (Hrsg.) ; Rijke, M. de (Hrsg.): *Specifying Syntactic Structures*. 1997, S. 75–113

[67] Lakoff, George: *Women, fire, and dangerous things: What categories reveal about the mind*. University of Chicago press, 1987

[68] Lambek, Joachim: The mathematics of sentence structure. In: *The American Mathematical Monthly* 65 (1958), Nr. 3, S. 154–170

[69] Lambek, Joachim: On the calculus of syntactic types. In: Jakobson, Roman (Hrsg.): *Structure of Language and its Mathematical Aspects* Bd. XII. American Mathematical Society, 1961, S. 166–178

[70] Lambek, Joachim: Deductive systems and categories. In: *Mathematical Systems Theory* 2 (1968), Nr. 4, S. 287–318

[71] Lambek, Joachim: Cartesian closed categories and typed λ-calculi. In: *LITP Spring School on Theoretical Computer Science* Springer (Veranst.), 1985, S. 136–175

[72] Lambek, Joachim: Type Grammar Revisited. In: Lecomte, Alain (Hrsg.) ; Lamarche, François (Hrsg.) ; Perrier, Guy (Hrsg.): *Logical Aspects of Computational Linguistics, Second International Conference (LACL '97)* Bd. 1582, 1997, S. 1–27

[73] Lambek, Joachim: Type grammar revisited. In: *International conference on logical aspects of computational linguistics*, 1997, S. 1–27

[74] Levart, Borut: *Triangles on a Sphere*. 2011. – URL `http://demonstrations.wolfram.com/TrianglesOnASphere/`. – Zugriffsdatum: 2016-07-26. – Wolfram Demonstrations Project

[75] Lin, Dekang: Automatic retrieval and clustering of similar words. In: *36th Annual Meeting of the Association for Computational Linguistics* Bd. 2, 1998, S. 768–774

[76] Liu, Xiaoyong ; Croft, W B.: Cluster-based retrieval using language models. In: *Proceedings of the 27th annual international ACM SIGIR*

conference on Research and development in information retrieval, 2004, S. 186–193

[77] LORENZ, Robin ; PEARSON, Anna ; MEICHANETZIDIS, Konstantinos ; KARTSAKLIS, Dimitri ; COECKE, Bob: QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. In: *arXiv:2102.12846* (2021)

[78] LUND, Kevin ; BURGESS, Curt: Producing high-dimensional semantic spaces from lexical co-occurrence. In: *Behavior research methods, instruments, & computers* 28 (1996), Nr. 2, S. 203–208

[79] MANNING, Christopher ; SCHUTZE, Hinrich: *Foundations of statistical natural language processing*. MIT press, 1999

[80] MARKOV, Andrei A.: Primer statisticheskogo issledovanija nad tekstomEvgenija Onegina'illjustrirujuschij svjaz'ispytanij v tsep (An example of statistical study on the text of Eugene Onegin' illustrating the linking of events to a chain). In: *Izvestija Imp. Akad. nauk* (1913)

[81] MARQUIS, Jean-Pierre: *From a geometrical point of view: A study of the history and philosophy of category theory*. Bd. 14. Springer Science & Business Media, 2008

[82] MEDIN, Douglas L. ; GOLDSTONE, Robert L. ; GENTNER, Dedre: Respects for similarity. In: *Psychological review* 100 (1993), Nr. 2, S. 254

[83] MEICHANETZIDIS, Konstantinos ; GOGIOSO, Stefano ; DE FELICE, Giovanni ; CHIAPPORI, Nicolò ; TOUMI, Alexis ; COECKE, Bob: Quantum natural language processing on near-term quantum computers. In: *arXiv:2005.04147* (2020)

[84] MEICHANETZIDIS, Konstantinos ; TOUMI, Alexis ; FELICE, Giovanni de ; COECKE, Bob: Grammar-Aware Question-Answering on Quantum Computers. In: *arXiv:2012.03756* (2020)

[85] MEYER, Francois ; LEWIS, Martha: Modelling Lexical Ambiguity with Density Matrices. In: *Proceedings of the 24th Conference on Computational*

Natural Language Learning, Association for Computational Linguistics, 2020, S. 276–290

[86] Meyer, Francois ; Lewis, Martha: Modelling Lexical Ambiguity with Density Matrices. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '20)*, 2020, S. 276–290

[87] Mikolov, Tomas ; Chen, Kai ; Corrado, Greg ; Dean, Jeffrey: Efficient estimation of word representations in vector space. In: *arXiv preprint arXiv:1301.3781* (2013)

[88] Mikolov, Tomáš ; Deoras, Anoop ; Povey, Daniel ; Burget, Lukáš ; Černockỳ, Jan: Strategies for training large scale neural network language models. In: *2011 IEEE Workshop on Automatic Speech Recognition & Understanding* IEEE (Veranst.), 2011, S. 196–201

[89] Mikolov, Tomas ; Sutskever, Ilya ; Chen, Kai ; Corrado, Greg S. ; Dean, Jeff: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013, S. 3111–3119

[90] Milajevs, Dmitrijs ; Kartsaklis, Dimitri ; Sadrzadeh, Mehrnoosh ; Purver, Matthew: Evaluating Neural Word Representations in Tensor-Based Compositional Settings. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*, 2014, S. 708–719

[91] Mitchell, Jeff ; Lapata, Mirella: Composition in distributional models of semantics. In: *Cognitive science* 34 (2010), Nr. 8, S. 1388–1429

[92] Mitchell, Tom M.: Does machine learning really work? In: *AI magazine* 18 (1997), Nr. 3, S. 11–11

[93] Montague, Richard: Universal grammar. In: *Theoria* 36 (1970), Nr. 3, S. 373–398

[94] Moortgat, Michael: Multimodal Linguistic Inference. In: *Journal of Logic, Language and Information* 5 (1996), Nr. 3/4, S. 349–385

[95] Moortgat, Michael: Categorial type logics. In: Benthem, Johan van (Hrsg.) ; Meulen, Alice ter (Hrsg.): *Handbook of Logic and Language*. Amsterdam : Elsevier, 1997, S. 93–177

[96] Moortgat, Michael ; Sadrzadeh, Mehrnoosh ; Wijnholds, Gijs: A Frobenius Algebraic Analysis for Parasitic Gaps. In: *Journal of Applied Logics* 7 (2020), Nr. 5, S. 823–852

[97] Moortgat, MJ ; Wijnholds, Gijs ; Cremers, Alexandre ; Gessel, Thom van ; Roelofsen, Floris u. a.: Lexical and Derivational Meaning in Vector-Based Models of Relativisation. In: *Proceedings of the 21st Amsterdam Colloquium* ILLC, University of Amsterdam (Veranst.), 2017, S. 55

[98] Moot, Richard ; Retoré, Christian: *The logic of categorial grammars: a deductive account of natural language syntax and semantics*. Bd. 6850. Springer, 2012

[99] Morrill, Glyn: *Categorial grammar: Logical syntax, semantics, and processing*. Oxford University Press, 2011

[100] Nasiruddin, Mohammad: A State of the Art of Word Sense Induction: A Way Towards Word Sense Disambiguation for Under-Resourced Languages. In: *Proceedings of RECITAL 2013*, 2013, S. 192–205

[101] Navigli, Roberto ; Crisafulli, Giuseppe: Inducing word senses to improve web search result clustering. In: *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*, 2010, S. 116–126

[102] Nielsen, Michael A. ; Chuang, Isaac: *Quantum computation and quantum information*. 2002

[103] Paperno, Denis ; Baroni, Marco u. a.: A practical and linguistically-motivated approach to compositional distributional semantics. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14)*, 2014, S. 90–99

[104] Partee, Barbara u. a.: Compositionality. In: *Varieties of formal semantics* 3 (1984), S. 281–311

[105] Pelletier, Francis J.: Did Frege believe Frege's principle? In: *Journal of Logic, Language and information* 10 (2001), Nr. 1, S. 87–114

[106] Pennington, Jeffrey ; Socher, Richard ; Manning, Christopher D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP '14)*, 2014, S. 1532–1543

[107] Peters, Matthew ; Neumann, Mark ; Iyyer, Mohit ; Gardner, Matt ; Clark, Christopher ; Lee, Kenton ; Zettlemoyer, Luke: Deep Contextualized Word Representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '18)*, 2018, S. 2227–2237

[108] Piedeleu, Robin: *Ambiguity in categorical models of meaning*, University of Oxford Master's thesis, Dissertation, 2014

[109] Piedeleu, Robin ; Kartsaklis, Dimitri ; Coecke, Bob ; Sadrzadeh, Mehrnoosh: Open System Categorical Quantum Semantics in Natural Language Processing. In: Moss, Larry (Hrsg.) ; Sobociński, Paweł (Hrsg.): *6th International Conference on Algebra and Coalgebra in Computer Science (CALCO'15)*, 2015, S. 267–286

[110] Piwowarski, Benjamin ; Frommholz, Ingo ; Lalmas, Mounia ; Van Rijsbergen, Keith: What can quantum theory bring to information retrieval. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, S. 59–68

[111] Radford, Alec ; Wu, Jeffrey ; Child, Rewon ; Luan, David ; Amodei, Dario ; Sutskever, Ilya u. a.: Language models are unsupervised multitask learners. In: *OpenAI blog* (2019)

[112] Rapp, Reinhard: Word sense discovery based on sense descriptor dissimilarity. In: *Proceedings of the ninth machine translation summit* Citeseer (Veranst.), 2003, S. 315–322

[113] Raschka, Sebastian: *Python machine learning*. Packt publishing, 2015

[114] Recski, Gábor ; Iklódi, Eszter ; Pajkossy, Katalin ; Kornai, András: Measuring Semantic Similarity of Words Using Concept Networks. In: *Proceedings of the 1st Workshop on Representation Learning for NLP (RepL4NLP '16)*, 2016, S. 193–200

[115] Richie, Russell ; Bhatia, Sudeep: Similarity judgment within and across categories: A comprehensive model comparison. In: *Cognitive Science* 45 (2021), Nr. 8, S. e13030

[116] Richie, Russell ; White, Bryan ; Bhatia, Sudeep ; Hout, Michael C.: The spatial arrangement method of measuring similarity can capture high-dimensional semantic structures. In: *Behavior research methods* 52 (2020), Nr. 5, S. 1906–1928

[117] Rieger, Burghard B.: *On distributed representation in word semantics*. International Computer Science Institute, 1991

[118] Rosenfeld, Ronald: Two decades of statistical language modeling: Where do we go from here? In: *Proceedings of the IEEE* 88 (2000), Nr. 8, S. 1270–1278

[119] Sadrzadeh, Mehrnoosh ; Kartsaklis, Dimitri ; Balkir, Esma: Sentence entailment in compositional distributional semantics. In: *Annals of Mathematics and Artificial Intelligence* 82 (2018), Nr. 4, S. 189–218

[120] Sadun, Lorenzo A.: *Applied linear algebra: The decoupling principle*. American Mathematical Soc., 2007

[121] Salton, G: The SMART system. In: *Retrieval Results and Future Plans* (1971)

[122] Salton, Gerard ; McGill, Michael J.: *Introduction to modern information retrieval*. McGraw Hill, 1983

[123] Salton, Gerard ; Wong, Anita ; Yang, Chung-Shu: A vector space model for automatic indexing. In: *Communications of the ACM* 18 (1975), Nr. 11, S. 613–620

[124] Schane, Sanford: Ambiguity and Misunderstanding in the Law. In: *T. Jefferson Law Review* 25 (2002), S. 167

[125] Schwarz, Mykhaylo ; Lobur, Mykhaylo ; Stekh, Yuriy: Analysis of the effectiveness of similarity measures for recommender systems. In: *Porceedings of the 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM '17)*, 2017, S. 275–277

[126] Selinger, Peter: Dagger compact closed categories and completely positive maps. In: *Electronic Notes in Theoretical computer science* 170 (2007), S. 139–163

[127] Shannon, Claude E.: Prediction and entropy of printed English. In: *Bell system technical journal* 30 (1951), Nr. 1, S. 50–64

[128] Shiebler, Dan ; Toumi, Alexis ; Sadrzadeh, Mehrnoosh: Incremental Monoidal Grammars. In: *arXiv:2001.02296* (2020)

[129] Sidorov, Grigori ; Gelbukh, Alexander ; Gómez-Adorno, Helena ; Pinto, David: Soft similarity and soft cosine measure: Similarity of features in vector space model. In: *Computación y Sistemas* 18 (2014), Nr. 3, S. 491–504

[130] Smith, Edward E. ; Osherson, Daniel N. ; Rips, Lance J. ; Keane, Margaret: Combining prototypes: A selective modification model. In: *Cognitive science* 12 (1988), Nr. 4, S. 485–527

[131] Soares, Marco Antonio C. ; Parreiras, Fernando S.: A literature review on question answering techniques, paradigms and systems. In: *Journal of King Saud University-Computer and Information Sciences* 32 (2020), Nr. 6, S. 635–646

[132] Speer, Robyn ; Chin, Joshua ; Havasi, Catherine: Conceptnet 5.5: An open multilingual graph of general knowledge. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI '17)*, 2017, S. 4444–4451

[133] Toshevska, Martina ; Stojanovska, Frosina ; Kalajdjieski, Jovan: Comparative analysis of word embeddings for capturing word similarities. In: *arXiv preprint arXiv:2005.03812* (2020)

[134] Turney, Peter D.: Similarity of semantic relations. In: *Computational Linguistics* 32 (2006), Nr. 3, S. 379–416

[135] Turney, Peter D. ; Pantel, Patrick: From frequency to meaning: Vector space models of semantics. In: *Journal of artificial intelligence research* 37 (2010), S. 141–188

[136] Tversky, Amos: Features of similarity. In: *Psychological review* 84 (1977), Nr. 4, S. 327

[137] Van Benthem, Johan: Language in action. In: *Journal of philosophical logic* 20 (1991), Nr. 3, S. 225–263

[138] Van Rijsbergen, Cornelis J.: *The geometry of information retrieval*. Cambridge University Press, 2004

[139] Wald, Robert M.: General relativity. In: *University of Chicago Press* (1984)

[140] Wald, Robert M.: *General relativity*. University of Chicago Press, 2010

[141] Wansing, Heinrich: Formulas-as-types for a hierarchy of sublogics of intuitionistic propositional logic. In: Pearce, David (Hrsg.) ; Wansing, Heinrich (Hrsg.): *Nonclassical Logics and Information Processing*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1992, S. 125–145

[142] Weaver, Warren u. a.: Translation. In: *Machine translation of languages* 14 (1955), Nr. 15-23, S. 10

[143] Widdowson, Henry: Jr firth, 1957, papers in linguistics 1934–51. In: *International Journal of Applied Linguistics* 17 (2007), Nr. 3, S. 402–413

[144] Wijnholds, Gijs ; Sadrzadeh, Mehrnoosh: Evaluating Composition Models for Verb Phrase Elliptical Sentence Embeddings. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, S. 261–271

[145] Wijnholds, Gijs ; Sadrzadeh, Mehrnoosh ; Clark, Stephen: Representation Learning for Type-Driven Composition. In: *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '20)*, 2020, S. 313–324

[146] Witten, Ian H. ; Frank, Eibe ; Hall, Mark A. ; Pal, CJ: *Data Mining: Practical machine learning tools and techniques*. Elsevier, 2005

[147] Wittgenstein, Ludwig ; Anscombe, Gertrude Elizabeth M. ; Rhees, Rush: *Philosophische Untersuchungen.(Philosophical investigations*. 1953

[148] Wouters, J. J.: Ornstein en de ontwikkeling van het Utrechtse Instituut voor Theoretische Fysica. In: *Nederlandsch tijdschrift voor natuurkunde* 87 (2021), September, Nr. 9, S. 42–46

SUMMARY IN ENGLISH

Oftentimes we say things that can have more than one meaning at the same time:

“Look at the dog with one eye”.

Are you using just one eye to look at the dog, or is it the dog that only has one eye?

We would really like a computer to understand that both of these meanings are possible, but this is very hard to achieve for normal computers. However, there is a special type of computer, called a quantum computer, for which this task is particularly suited. This computer uses a property of nature called “quantum superposition”, which allows for two apparently opposite things to exist at the same time. For instance, a cat can be dead and alive at the same time, as in the famous Schrodinger’s cat. By using that property, both meanings of a phrase can too be true at the same time, until more information is given to infer the intended meaning. In this way, we hope to have computers that are smarter and can understand us better. In this thesis we try to develop algorithms that make this goal possible, advancing the frontier of applications of quantum computation in the field of natural language processing.

To start with, we look at one of the central obstacles of human-computer interaction: the fact that computers understand very little about the meanings of words. Historically, this communication has happened using specialized computer languages, which are particularly fit to encode meanings related to logical statements. For instance, `x+3` can be a way to write "The sum of any number with the number three". If properly programmed, one can obtain the actual result of this sum for a given `x`, which we would consider its *meaning*. The fundamental question that this thesis tries to contribute to is thus: is it possible to code and compute the meaning of phrases, such as "fluffy dog", in a way that encodes not only logical reasoning, but provides

a way to return our (possibly multiple) common understanding(s) of such a phrase?

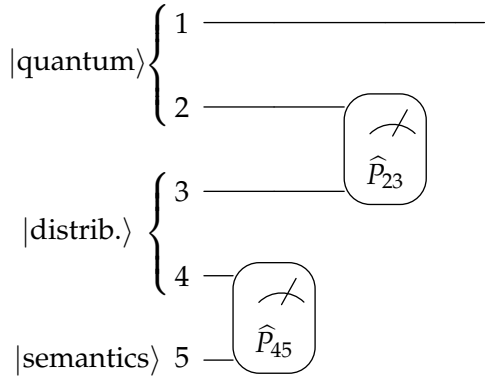
To this end, we rely on a compositional approach: the grammar rules act as the "computational rules", and the meanings of the individual words compose with each other following those same rules, in a way that is consistent with the type of meaning that we wish to assign to each word. To use the grammar rules in this way, we rely on Lambek types. In this framework, a common noun is represented by a Lambek type n and an adjective is represented by a type n/n. Following the appropriate type composition rules, we can show that these two types can be mapped to the type n. This means that an adjective applied on a noun gives something that can be used in the same place on a sentence. This is very intuitive: "fluffy dog" could always be used in the place of "dog".

To obtain the meaning of this phrase in a compositional way, the meanings must compose in a way that is homomorphic to the grammar composition. Additionally, it is convenient that these individual meanings have a format that a computer can understand. The choice made here comes from an approach to NLP called "distributional" semantics, that is able to extract vector representations of words by looking at the other words that usually appear next to it: "You shall know a word by the company it keeps".

Each Lambek type is assigned a vector space, and so the composition takes place using operations in vector spaces. Then, each word of a certain type is assigned an element of the correct vector space, a vector. The vector space associated with the Lambek type n is N, which means that a word like "dog" is represented by a vector in the noun space. In its turn, the vector space associated with the Lambek type n/n is $N \otimes N$, which describe maps between vectors. The elements of this space can be represented as matrices, and so an adjective such as "fluffy" is also represented by a matrix. The composition of the adjective and the noun, thus, is automatically computed by a computer as the application of the respective matrix to the respective vector, which in turn results in a vector in the space N, consistent with the resulting Lambek type n of grammar composition.

This framework can be extend to more complex types and grammar formulations, which can get rather computationally heavy. One possibility

to make these calculations more computationally efficient is to resort to quantum computation, by having each vector component as a quantum state and by performing the contractions using a quantum circuit. In this thesis we explore those avenues, with special focus on ambiguities that stem from different meanings for each word (for which density matrices are particularly suited) and ambiguities that appear due to different possible readings of the same phrase, as is the case of the scope in "old men and women", or relative clauses in Dutch, where "man die de hond bijt" can translate either to "man that bites the dog" or "man that the dog bites". For these, the different possible contractions represent different readings, and they can exist simultaneously in quantum superposition. In the process, we explore ways of improving cosine similarity calculations using a metric tensor, and we develop a quantum computation algorithm that is capable to speed up the process of finding an answer to an open multiple choice question.



SAMMENVATTING IN HET NEDERLANDS

Vaak zeggen we dingen die meer dan één betekenis tegelijkertijd kunnen hebben:

"Kijk naar de hond met één oog".

Gebruik je maar één oog om naar de hond te kijken, , of is het de hond die maar één oog heeft?

We zouden graag willen dat een computer begrijpt dat beide betekenissen mogelijk zijn, , maar dit is erg moeilijk te bereiken voor normale computers. Er is echter een speciaal type computer, een de kwantumcomputer genaamd, waarvoor deze taak bijzonder geschikt is. Deze computer gebruikt een eigenschap van de natuur die "kwantumsuperpositie" wordt genoemd, waardoor twee schijnbaar tegengestelde dingen tegelijkertijd kunnen bestaan. Een kat kan bijvoorbeeld tegelijkertijd dood en levend zijn, zoals in het de beroemde voorbeeld van Schrödingers kat. Door die eigenschap te gebruiken, kunnen beide betekenissen van een zinsdeel kunnen ook tegelijkertijd waar zijn, totdat er meer informatie is gegeven om de beoogde betekenis af te leiden. Op deze manier hopen we computers te hebben die slimmer zijn en ons beter kunnen begrijpen. In dit proefschrift proberen we algoritmen te ontwikkelen die dit doel mogelijk maken, waardoor de grens van toepassingen van quantum computation kwantumberekening op het gebied van natuurlijke taalverwerking wordt vervroegdverlegd.

Om te beginnen, kijken we naar een van de centrale obstakels van mens-computerinteractie: het feit dat computers heel weinig begrijpen van de betekenis van woorden. Historisch gezien gebeurde deze communicatie met behulp van gespecialiseerde computertalen, die bijzonder geschikt zijn om betekenissen te coderen die verband houden met logische uitspraken. Zo kan `x+3` een manier zijn om te schrijven "De som van een willekeurig getal met het getal drie". Indien correct geprogrammeerd, kan men het

werkelijke resultaat van deze som verkrijgen voor een gegeven x, waarvan hetgeen we als de *betekenis* zouden beschouwen. De fundamentele vraag waar dit proefschrift een bijdrage aan probeert te leveren is dus als volgt: is het mogelijk om de betekenis van zinsdelen te coderen en te berekenen, zoals "pluizige hond", op een manier die niet alleen logisch redeneren codeert, maar ook voorziet in een manier om ons (mogelijk meerdere) gemeenschappelijke begrip(pen) van zo'n zinsdeel terug te geven?

Hiervoor vertrouwen we op een compositionele benadering: de grammaticaregels fungeren als de "rekenregels", en de betekenissen van de afzonderlijke woorden vormen met elkaar samen worden samengevoegd volgens dezelfde regels, op een manier die consistent is met het soort betekenis dat we aan elk woord willen toekennen. Om de grammaticaregels op deze manier te gebruiken, vertrouwen we op Lambek-typen. In dit raamwerk wordt een zelfstandig naamwoord weergegeven door een Lambek type n, en een bijvoeglijk naamwoord wordt weergegeven door een type n/n. Volgens de toepasselijke regels voor de samenstelling van het type, kunnen we laten zien dat deze twee typen kunnen worden toegewezen aan afgebeeld op het type n. Dit is heel intuïtief: "pluizige hond" kan altijd worden gebruikt in plaats van "hond".

Om de betekenis van deze zin frase op een compositionele manier te verkrijgen, moeten de betekenissen zijn samengesteld op een manier die homomorf is aan ten opzichte vande grammaticale compositie. Bovendien is het handig dat deze individuele betekenissen een formaat hebben dat een computer kan begrijpen. De keuze die hier wordt gemaakt, komt van een benadering van NLP die "distributionele" semantiek heet, die vectorrepresentaties van woorden kan extraheren door te kijken naar de andere woorden die er meestal naast staan: "Je zult een woord kennen van het bedrijf dat het houdtgezelschap waarin het verkeert".

Elk Lambek-type krijgt een vectorruimte toegewezen, en dus vindt de compositie plaats met behulp van bewerkingen in vectorruimten. Vervolgens wordt aan elk woord van een bepaald type een element van de juiste vectorruimte toegewezen, een vector. De vectorruimte die hoort bij het Lambek-type n is N, wat betekent dat een woord als "hond" wordt weergegeven door een vector in de ruimte van zelfstandige naamwoor-

denruimte. Op zijn haar beurt is de vectorruimte geassocieerd met het Lambek-type n/n $N \otimes N$, die kaarten lineaire afbeeldingen tussen vectoren beschrijftven. De elementen van deze ruimte kunnen worden weergegeven als matrices, en dus wordt een bijvoeglijk naamwoord zoals "pluizig" ook weergegeven door een matrix. De compositie van het bijvoeglijk naamwoord en het zelfstandig naamwoord wordt dus automatisch berekend door een computer als de toepassing van de respectievelijke matrix op de respectievelijke vector, wat op zijn beurt resulteert in een vector in de ruimte N, consistent met het resulterende Lambek-type n van grammaticale compositie.

Dit raamwerk kan worden uitgebreid tot complexere typen en grammaticale formuleringen, , die rekenkundig nogal rekenkundig zwaar kan worden. Een mogelijkheid om deze berekeningen rekenkundig efficiënter te maken, is door gebruik te maken van kwantumberekening , door elke vectorcomponent als een kwantumtoestand te hebben beschouwen en door de samentrekkingen uit te voeren met behulp van een kwantumcircuit. In dit proefschrift verkennen we die wegenrichtingen, met speciale aandacht voor dubbelzinnigheden die voortkomen uit verschillende betekenissen voor elk woord (waarvoor dichtheidsmatrices bijzonder geschikt zijn) en dubbelzinnigheden die verschijnen als gevolg van verschillende mogelijke lezingen van dezelfde zinsdeel, , zoals het geval is met oude de reikwijdte in "oude mannen en vrouwen", of bijvoeglijke bijzinnen in het Nederlands, waar "man die de hondt bijt" twee betekenissen kan hebben. Hiervoor vertegenwoordigen de verschillende mogelijke contracties verschillende lezingen, en ze kunnen gelijktijdig bestaan in kwantumsuperpositie. In het proces onderzoeken we manieren om cosinus-gelijkenisovereenkomstberekeningen te verbeteren met behulp van een metrische tensor, en we ontwikkelen we een kwantumberekeningsalgoritme dat in staat is om het proces van het vinden van een antwoord op een open meerkeuzevraag te versnellen.

ACKNOWLEDGMENTS

"A viagem não acaba nunca. Só os viajantes acabam. E mesmos estes podem prolongar-se em memória, em lembrança, em narrativa."

– José Saramago, "Viagem a Portugal"

This thesis would not have been possible without the support of a great many people that contributed to push me forward, one way or another. Some of you I will inevitably forget to mention, and so it is to you that my first acknowledgement goes.

Thanks to my thesis supervisors, for embarking with me on this daring project. Henk, thank you for supporting me since the beginning in pursuing my interests, and for always being there to encourage me and share your unparalleled physical intuition. Michael, thank you for your generosity in sharing with me your knowledge and passion for logics and linguistics, and for constantly pushing me to explore new territory. A special thanks to Peter Koeze for supporting this project.

Thanks to my colleagues, my travel buddies. Gijs, thank you for being there, especially on the last steps of the this journey. You've been my connection to the field in ways that have empowered and inspired me. Kokos, thank you for being my sounding board on so many occasions, notably in the earlier days. You helped me become a better researcher and human being.

Thanks to my office mates. Jette, Carlene and Deniz, we overcame a lot together. Anarchofeminism forever!

Thanks to my ITF and CCSS colleagues, for their curious and supporting spirit.

Thanks to my students. Teaching you kept me fresh and challenged, and sent me in directions that ended up making their way into this thesis.

Thanks to my friends, that were so important during this time. Camila, Camilla, João, Joana, Maaike, Margherita, Pedro, Rita, Valentim, Vasco, thank you for travelling some of this journey by my side. Your sweetness and strength has lifted my spirit.

Finally, my biggest thanks goes to my family, my brother Alexandre and, especially, my parents, Fátima e António. Obrigada por terem estado ao meu lado sempre que precisei de apoio, e por acreditarem em mim em todos os momentos. Esta tese é para vocês.

ABOUT THE AUTHOR

Adriana Duarte Correia was born in Loulé, Portugal, in 1994, where she lived with her brother and parents until moving away for college. She attended Loulé High School, where she followed the Science and Technology track. She completed a Bachelor's degree in Physics from University of Coimbra in 2015, the same year where she enrolled in the Theoretical Physics Master's degree at Utrecht University. Here she chose the Complex Systems profile, writing her thesis on the modelling of games played between many players as an Ising system, under the supervision of H.T.C. Stoof, completing the programme *cum laude*. During this time she also took part in several committees from Erasmus Student Network Utrecht. Following her broad interests, in 2018 she started a doctoral programme on the connections between physics and linguistics, in association with the Center for Complex Systems Studies.

Adriana currently lives in Utrecht, but likes to travel and go back to Portugal as often as possible.

V[A • B] = {X · Y | X ∈ V(A) & Y∈V(B)}
1
1
1
0