Table of Contents

Table of Contents **i**

Abstract **v**

Introduction **1**

1 Previous work **5**

1.1 V. Sánchez: Studies on Natural Logic and Categorial Grammar (1991) . . 5

1.2 D. Dowty: The role of Negative Polarity and Concord Marking in Natural Language Reasoning (1994) 8

1.3 R. Bernardi: Reasoning with Polarity in Categorial Type Logic (2002) . . . 10

1.4 Fyodorov et al.: Order-Based Inference in Natural Language (2002) 13

2 Semantic types and order relations **16**

2.1 Basic semantic notions 16

2.2 Decoration of types 19

3 Type-Logical Categorial Grammar **21**

4 The $\mathcal{L}$-based Order Calculus **26**

4.1 $\mathcal{L}$-OC 26

4.2 The semantics of $\mathcal{L}$-OC 29

5 $\mathcal{L}$-OC as an inference system for natural language **31**

5.1 The lexicon 31

5.2 Natural Logic inferences 32

5.3 Non-logical axioms and the ‘every’ postulate 33

5.4 Examples of $\mathcal{L}$-OC derivations 34

6 Exploring properties of $\mathcal{L}$-OC **41**

6.1 Normalization in $\mathcal{L}$-OC 41

6.1.1 Examples of using the normalization axioms 41

6.1.2 Why is normalization needed? 44

6.1.3 Dynamic marking: an alternative? 50

6.2 Multiple derivations 57

7 The Proof Search Procedure **62**
7.1 Description of the algorithm 62
7.2 Correctness 66

8 Conclusions **68**

A Termination of the proof search procedure **70**

B Completeness of the proof search procedure **76**

Bibliography **90**

List of Figures

5.1 Deriving $S_1, ..., S_n \vdash_{NatLog} S$ in the system 33

5.2 $\mathcal{L}$ derivation of Every student whom Mary kissed smiled 36

5.3 Relative clauses:
Every student whom Mary touched smiled $\vdash_{NatLog}$Every student whom Mary kissed smiled 37

5.4 $\mathcal{L}$ derivation of Some boy the brother of whom Mary loves walked 38

5.5 Pied piping:
Some boy, the brother of whom Mary loves, walked $\vdash_{NatLog}$Some boy walked . 39

5.6 $\mathcal{L}$ derivation of Mary kissed every student 39

5.7 Using the extended non-logical postulate for 'every':Mary kissed every student, No student whom Mary kissed walked $\vdash_{NatLog}$ No student walked 40

6.1 Mary adores and passionately loves John $\vdash_{NatLog}$ Mary loves John 42

6.2 Some creative intelligent boy smiled $\vdash_{NatLog}$ Some smart boy smiled, using the non-logical axiom $(a5)$. 43

6.3 The problematic inference is derivable in $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$: John does and Mary doesn't move $\vdash_{NatLog}$ Mary doesn't walk 51

6.4 $\mathcal{L}$ derivation of No man loves some woman corresponding to the first semantic reading 59

6.5 $\mathcal{L}$ derivation of No man loves some woman corresponding to the second semantic reading 60

6.6 $\mathcal{L}$-OC derivation of No man loves some tall woman$\vdash_{NatLog}$No man loves some woman 60

6.7 $\mathcal{L}$-OC derivation of No man loves some woman$\vdash_{NatLog}$No man loves some tall woman 61

7.1 The call tree of the algorithm for the $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof from fig. 6.3 67

List of Tables

1.1 Encoding monotonicity and polarity 11

5.1 The lexicon 32

Abstract

Most of the literature in natural reasoning assumes a model-theoretic perspective and uses translation into some formal language, e.g. first order logic, as an intermediate step for computing inferences. The general goal of a 'Natural Logic' inference system is giving a systematic account of natural reasoning using linguistic structures as the vehicle of inference.

This thesis develops a 'Natural Logic' inference system based on a modification of the (associative) Lambek calculus ($\mathcal{L}$). The proposed inference system manipulates natural language syntactic representations with no intermediate translation to logical formulae. The main part of the system used for deriving inferences is the *$\mathcal{L}$-based Order Calculus* ($\mathcal{L}$-OC), which manipulates *order statements* between proof terms, representing $\mathcal{L}$ derivation trees of natural language expressions via the Curry-Howard isomorphism.

Our work extends the system of [4], which is based on Ajdukiewicz-Bar Hillel calculus. In our work we overcome some limitations of the system of [4] and enable it to deal with new kinds of inferences, such as inferences involving sentences with extraction, pied piping etc.

Basing the inference system on $\mathcal{L}$ brings about a certain complication - the emergence of proof terms which are not in normal form. We augment $\mathcal{L}$-OC with normalization axioms as a remedy for this complication and demonstrate that they enable the system to derive new kinds of inferences. Finally, we present a terminating proof search procedure.

Introduction

Model-theoretic semantic theories of natural language assume that most linguistic expressions denote objects in partially ordered domains, so that meanings of expressions (i.e., denotations in models) of the same semantic type are naturally comparable. Formal semantics treats order relations between denotations of complex expressions as compositionally derived from order relations between denotations of simpler expressions (subexpressions), based on a given grammar and certain semantic properties of denotations of words. For instance, under standard assumptions about the meaning of certain adjectives, like tall, the denotation of the nominal expression tall student is semantically "smaller" than the denotation of the noun student in every model. This simple ordering, together with the 'order reversing' meaning of the determiner no, is responsible for the fact that the denotation of the noun phrase no tall student is semantically "greater" than the denotation of the noun phrase no student in every model. At the top level, such order statements result in a semantic ordering of natural language (indicative) sentences. In an adequate semantic theory, this ordering between sentences corresponds to an intuitively valid *entailment* relation. For instance, the mentioned order relations, together with the other elements in the sentence, are responsible for the valid entailment John saw no student ⇒ John saw no tall student.

However, appealing to models to derive inferences is not computationally feasible. This thesis aims to develop an inference system based on basic insights on order relations from model-theoretic semantics, but using only proof-theoretic manipulations of natural language syntactic representations, with no direct appeal to models. Only some *abstractions* of the full denotations are used as a basis of the proposed calculus of inference. Furthermore, the close relationship between structure and meaning in model-theoretic semantics eliminates the need of translating the syntactic representations into intermediate logical levels of representation, such as first order logic.

The initial conception of *Natural Logic* was introduced in [11]. The general goal of a Natural Logic system is giving a systematic account of natural reasoning using grammatical structures of natural language. Different versions of Natural Logic were proposed by [9], [1], [4] and others. Sánchez ([9]) proposes a mechanism that decorates categorial grammar proofs of natural language expressions using signs that indicate the *monotonicity* properties of the denotations of these expressions. Bernardi ([1]) follows [9] and introduces a system for monotonicity reasoning that is based on a more complex categorial type logic than Sánchez' work. In addition, she concentrates on monotonicity reasoning in order to capture the *syntactic* distribution of negative polarity items, without fully accounting for monotonicity-based inferences in natural language (by contrast to this thesis, where we concentrate on a calculus for deriving inferences, without any attempt to treat negative polarity items). However, neither Sánchez nor Bernardi provide a formal calculus for computing inferences. The situation was partially amended in Fyodorov et al. ([4]), where a calculus based on similar annotations decorating syntactic derivations of the AB calculus is proposed.

The goal of this thesis is to present and study a more extensive inference system based on a slight modification of the (associative) Lambek calculus ($\mathcal{L}$). We define an $\mathcal{L}$-based Order Calculus ($\mathcal{L}$-OC) for deriving inferences in natural language. $\mathcal{L}$-OC manipulates *order statements* between proof terms representing $\mathcal{L}$ derivation trees of natural language expressions via the Curry-Howard isomorphism. Although $\mathcal{L}$-OC order statements reflect semantic order relations between elements of partially ordered domains, they are purely syntactic units and are derived with no direct appeal to models, only to the abstractions of denotations based on the above mentioned annotations.

As was mentioned above, we extend an existing system of [4] which is based on the AB calculus. Their system allows a rather straightforward derivation of inferences with monotone and non-monotone quantifiers and cross-categorial conjunctions and disjunctions. However, despite its value for demonstrating a novel technique of inference in natural language, it fails to derive many kinds of inferences due to the syntactic limitations of the AB calculus as the syntactic calculus underlying the categorial grammar. In this thesis we show that the situation can be partially amended by basing the system on $\mathcal{L}$. However, in view of the well-known limitations of $\mathcal{L}$ itself, it is clear that in order to define a more realistic system that can cope with a larger variety of inferences derivable in natural language, a yet more complex grammatical formalism is needed. Hence, we view $\mathcal{L}$-OC as an intermediate step towards a general system supporting various kinds of inferences in natural language, to be eventually based on some decidable fragment of the multi-modal type logical grammar ([8]).

The proposed $\mathcal{L}$-OC has the following characteristics:

1. Basing the system on $\mathcal{L}$ allows us to use a more elegant representation of derivations as Curry-Howard proof terms. This is opposed to the formulae of the Order Calculus of [4], which are order statements between (AB) derivation trees.

2. $\mathcal{L}$-OC is a modification of the Order Calculus of [4], which we augment with:

   (a) *Abstraction* inference rule that enables $\mathcal{L}$-OC to deal with inferences involving sentences with extraction, pied piping etc. Consider the following examples derivable in $\mathcal{L}$-OC (to be discussed in chapter 5):

      i. Every student whom Mary touched smiled $\Rightarrow$ Every student whom Mary kissed smiled given the order statement kissed $\leq$ touched.

      ii. Some boy, the brother of whom Mary loves, walked $\Rightarrow$ Some boy walked.

   (b) β/η-normalization axioms, based on β/η-reduction of proof terms. These axioms resolve a certain complication caused by basing the system on $\mathcal{L}$ – the emergence of proof terms that are not in normal form. The normalization axioms enable the system to derive more kinds of inferences (for reasons to be discussed in chapter 6), for example:

   John does and Mary doesn't move $\Rightarrow$ Mary doesn't walk

   given the order statement walk $\leq$ move.

3. By basing the system on $\mathcal{L}$, we also obtain the ability to use more complex non-logical axioms, that better reflect the denotations of natural language expressions.

> For example, we can define the following order statement as a non-logical axiom of $\mathcal{L}$-OC : passionately love $\leq$ adore. This is made possible due to the derivability of function composition in $\mathcal{L}$ (to be further explained).

Finally, we propose a terminating proof search procedure for $\mathcal{L}$-OC , based on the proposal by [4].

To sum up, the main contribution of this thesis is introducing a $\mathcal{L}$-based formal Order Calculus for computing Natural Logic inferences, for which a proof search procedure is defined. In extending the AB-based Order Calculus of [4] to be based on $\mathcal{L}$, we had to deal with a number of challenges, such as non-normalized $\mathcal{L}$ derivations (treated by using normalization axioms), decorations of types for dynamically created abstraction terms (one possible solution – *dynamic marking*, to be further explained).

This thesis has 8 chapters. Chapter 1 discusses some previous work on direct inference in natural language and its relation to the current work. Chapter 2 provides some necessary background on basic notions from model-theoretic semantics, such as semantic types, order relations etc. Chapter 3 describes $\mathcal{L}$ and the categorial grammar based on it. Chapter 4 defines $\mathcal{L}$-OC and its semantics. Chapter 5 demonstrates the applications of $\mathcal{L}$-OC for deriving natural language inferences and presents examples. Chapter 6 is devoted to exploring the properties of $\mathcal{L}$-OC . Its first section focuses on the problem of normalization, explaining how the non-normalized proof terms are created in $\mathcal{L}$-OC, why they pose a problem and how the problem is solved using normalization. It also presents examples of inferences using normalization and describes an alternative method for solving the mentioned problem – using *dynamic marking*. Its second section describes how multiple derivations in $\mathcal{L}$ can be used to account for scope ambiguity when deriving inferences. Chapter 7 defines the proof search procedure and discusses its correctness. The full correctness proofs are provided in appendixes A and B. Chapter 8 presents conclusions and directions for further research.

Chapter 1

Previous work

There is a number of proposals of systems using grammatical structures to account for natural language inferences. Some of the earlier ones are [10] and [7], the overview of which can be found in [4]. In this chapter we give an overview of the previous work that is most relevant to this thesis, including [9], [3], [1] and [4].

1.1 V. Sánchez: Studies on Natural Logic and Categorial Grammar (1991)

Sánchez ([9]) describes a Natural Logic system for calculating entailments between natural language sentences. The system manipulates marked undirected Lambek calculus derivation trees of English sentences. The marking of proof nodes allows to determine whether the derived expressions are in a *positive* or *negative* position. An expression E_1 is in *positive (negative) position* in an expression E_2 if its replacement by a *semantically greater* expression results in E_2', which is semantically greater (smaller) than E_2. An expression that is in positive/negative position is said to have *polarity*.

The semantic order relation is standardly defined as the numerical order relation for truth values and an inclusion relation for complex types, quantifying over all models. For sentences the order relation corresponds to the entailment relation. A sentence S' is entailed by a sentence S iff the denotation of S is less or equal to the denotation of S' in any model. The polarity of occurrences of expressions is used to calculate entailments in the following way: if (i) sentence S' is obtained from sentence S by replacing its subexpression R by R', (ii) R is semantically smaller (greater) than R', and (iii) R has positive (negative) polarity in S, then S entails S'.

Sánchez uses an external algorithm working on derivations of the associative and commutative Lambek calculus, which is able to mark the parsed strings determining the different polarity positions. Given a derivation, the algorithm starts from marked leaves. The categories of the lexical items are assigned monotonicity markings according to their standard semantics. If the denotation of a lexical item is an upward/downward monotone function, then its category is marked with '+'/'-' accordingly. Then the markings are propagated through the proof by labelling the rest of the nodes in the following way:

1. The major premise $(\alpha \rightarrow \beta)$ in an instance of arrow elimination is positive:

$$\frac{(\alpha \rightarrow \beta) \quad \alpha}{\beta} \quad \Rightarrow \quad \frac{\overset{(\alpha \rightarrow \beta)}{+} \quad \alpha}{\beta}$$

2. The minor premise in an arrow elimination is positive (negative) if the major's category is upward (downward) monotone:

$$\frac{(\alpha^{+(-)} \rightarrow \beta) \quad \alpha}{\beta} \quad \Rightarrow \quad \frac{\overset{(\alpha^{+(-)} \rightarrow \beta)}{+(-)} \qquad \overset{\alpha}{+(-)}}{\beta}$$

3. The withdrawal of a numerical index leaves the previous marking unchanged:

$$\frac{\begin{matrix}[\alpha^i] \\ D_1 \\ \beta\end{matrix}}{(\alpha \rightarrow \beta)}\, I \quad \Rightarrow \quad \frac{\begin{matrix}[\alpha^i] \\ D_1 \\ \beta \\ +\end{matrix}}{(\alpha^{+} \rightarrow \beta)}\, I$$

4. If each node from α^i to β is either positive or negative and the number of negative nodes is even (odd), then:

$$\dfrac{\begin{array}{c}[\alpha^i]\\ D_1\\ \beta\end{array}}{(\alpha \rightarrow \beta)}\,(I) \quad \Rightarrow \quad \dfrac{\begin{array}{c}[\alpha^i]\\ D_1\\ \beta\\ +(-)\end{array}}{(\alpha^{+(-)} \rightarrow \beta)}\,(I)$$

Finally, the polarity of nodes is computed in the following way. Let D be a derivation with conclusion α. A node γ has polarity iff all the nodes in the path from γ to α are marked, and γ is in a positive (negative) position in D iff (a) γ has polarity, and (b) the number of nodes marked by '-' in the path from γ to the root is even (odd).

The final result is a parsed output in which polarity positions are displayed, so that they can be used for computing entailments. For example, below is an annotated derivation of the sentence Heloise wanders:

$$\dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\dfrac{\begin{array}{cc}(e \rightarrow t) & \textsf{Heloise}\\ + & e\end{array}}{t}\,(E)\\ +\end{array}}{((e \rightarrow t)^{+} \rightarrow t)}\,(I)\\ +\end{array} \qquad \begin{array}{c}\textsf{wanders}\\ (e \rightarrow t)\\ +\end{array}}{t}\,(E) \qquad (1)$$

According to the rules for polarity of subderivations, presented above, (2a) occurs positively in (1). It is also given that the denotation of (2a) is semantically smaller than the denotation of (2b) (Sánchez does not provide any formal method for calculating this order relation).

$$\dfrac{\begin{array}{c}\dfrac{\begin{array}{cc}(e \rightarrow t) & \textsf{Heloise}\\ + & e\end{array}}{t}\,(E)\\ +\end{array}}{((e \rightarrow t)^{+} \rightarrow t)}\,(I) \qquad (2a) \qquad \dfrac{\begin{array}{cc}\textsf{Some} & \textsf{nun}\\ (p^{+} \rightarrow ((e \rightarrow t)^{+} \rightarrow t)) & p\\ + & +\end{array}}{((e \rightarrow t)^{+} \rightarrow t)}\,(E) \qquad (2b)$$

By replacing (2a) by (2b), we obtain (3):

$$\dfrac{\dfrac{\underset{+}{\overset{\mathsf{Some}}{(p^+ \to ((e \to t)^+ \to t))}} \qquad \underset{+}{\overset{\mathsf{nun}}{p}}}{\underset{+}{((e \to t)^+ \to t)}}\,(I) \qquad \underset{+}{\overset{\mathsf{wanders}}{(e \to t)}}}{t}\,(E) \qquad (3)$$

Thus, given that Heloise is semantically smaller than Some nun, we can prove that Heloise wanders entails Some nun wanders.

While the annotation of derivation trees with monotonicity markings and calculation of polarity of subderivations are rigorously defined in the work of Sánchez, he provides no calculus that can derive the order relations using that annotation. As a result, the range of the inferences captured by his system is rather limited.

1.2 D. Dowty: The role of Negative Polarity and Concord Marking in Natural Language Reasoning (1994)

Dowty ([3]) explores the linking between Natural Logic and several linguistic phenomena related to monotonicity, such as the fact that negative polarity items (NPI's) in English and other languages are limited to downward monotone contexts. For this purpose, he proposes a reformulation of the system of Sánchez, while his goal is to "collapse" the independent steps of monotonicity marking and polarity determination into the syntactic derivation itself. He uses the markers '+' and '-' to indicate the final polarity.

Dowty proposes a Natural Logic system with the following characteristics:

1. Since one and the same word can appear with positive polarity in one derivation and with negative polarity in another, most lexical items (with the important exception of negative and positive polarity items) will have both a'+' and a '-' marked category, with the same interpretation.

2. Upward monotone functions are assigned categories of the forms A^+/B^+ and A^-/B^-, meaning that they *preserve* the polarity markers. The pattern A^x/B^x matches both of these cases.

3. Downward monotone functions are assigned categories of the forms A^+/B^- and A^-/B^+, meaning that they *reverse* the polarity markers. The pattern A^x/B^y matches both of these cases.

Functional application respects the polarity markers in the following way. Let $x \in \{+,-\}$, then

$$\frac{A^x/B^y \quad B^y}{A^x}$$

where x and y coincide when the major premise A^x/B^x is an upward monotone function, and differ when it is a downward monotone function.

The defined grammar generates sentences of category S^+ (the category of independent grammatical sentences) and S^- (the category of sentences embedded inside the scope of a downward monotone function).

Below are some examples of the lexical entries. Let $x \in \{+,-\}$ and y is the "opposite" of x. Let $VP^x = NP^x\backslash S^x$ and TV.

$$\textsf{walks} = VP^x \quad \textsf{book} = CN^x$$

$$\textsf{John} = S^x/VP^x \quad \textsf{doesn't} = VP^x/VP^y$$

$$\textsf{no} = \{(S^+/VP^-)/CN^-, (S^-/VP^+)/CN^+\}$$

$$\textsf{every} = \{(S^-/VP^-)/CN^+, (S^+/VP^+)/CN^-\}$$

$$\textsf{any} = (S^-/VP^-)/CN^-$$

Consider the following examples of Dowty's polarity marking:

$$\frac{\dfrac{\textsf{John}}{S+/VP^+} \quad \dfrac{\textsf{walks}}{VP^+}}{S^+} \qquad (1)$$

$$\frac{\dfrac{\textsf{John}}{S^+/VP^+} \quad \dfrac{\dfrac{\textsf{doesn't}}{VP^+/VP^-} \quad \dfrac{\textsf{walk}}{VP^-}}{VP^+}}{S^+} \qquad (2)$$

Note how the polarity of VP is changed by the presence of the downward monotone function doesn't.

To deal with more complex sentences in which a generalized quantifier occurs in object position, Dowty includes in the lexicon for each determiner $(S^x/VP^y)/CN^z$ an object counterpart, of category $(TV^y\backslash VP^x)/CN^x$, where $TV^y = (NP^y\backslash S^Y)/NP^y$.

Now let us observe how the system of Dowty can account for correct NPI distribution. Consider the sentences involving the NPI any:

(a) No boy reads any book

(b) *Every boy reads any book

The Dowty-style derivation of (a) is as follows:

$$\dfrac{\dfrac{\dfrac{\text{no}}{(S^+/VP^-)/CN^-)} \quad \dfrac{\text{boy}}{CN^-}}{S^+/VP^-} \quad \dfrac{\dfrac{\text{reads}}{TV^-} \quad \dfrac{\dfrac{\text{any}}{(TV^-\backslash VP^-)/CN^-} \quad \dfrac{\text{book}}{CN^-}}{TV^-\backslash VP^-}}{VP^-}}{S^+}$$

However, if we replace no with every, the derivation fails. In order to match the VP^- category of reads any book, every has to be considered of category $(S^-/VP^-)/CN^+$ and the whole expression every boy reads any book would be of category S^-.

By internalizing the monotonicity and polarity information into the logical types, Dowty enables his system to correctly account for some cases of negative polarity items distribution. However, the flow of the markers from the argument to the functional types expressed by the lexical items category assignments is too strong to model some linguistic information. Therefore, Dowty's system fails to deal with NPIs in embedded sentences and other more complex cases. For example, the system blocks the following correct sentences with embedded NPIs:

1. John doubts every boy reads any book.
2. If he doesn't know anything about logic, he will not know Modus Ponens.

1.3 R. Bernardi: Reasoning with Polarity in Categorial Type Logic (2002)

Bernardi ([1]) presents a Natural Logic system, which while marking linguistic structures with the information required to model inferences, also accounts for NPI distribution. She shows that a logical account of NPIs requires the use of internalized polarity markers, like

To express that:	Type
A structure Γ is an upward monotone function	$\Gamma \vdash B/A$ or $\Gamma \vdash A\backslash B$
A structure Γ is an upward monotone function	$\Gamma \vdash B/\ominus A$ or $\Gamma \vdash A\backslash \ominus B$
A structure Γ has negative polarity	$\langle \Gamma \rangle^-$
A structure Γ has positive polarity	$\Gamma \neq \langle \Gamma' \rangle^-$
A structure Γ must have negative polarity	$\Gamma \vdash \boxminus^{\downarrow} A$

Table 1.1: Encoding monotonicity and polarity

in the system of Dowty.

The Natural Logic of Bernardi is based on a multi-modal type-logical grammar with polarity structural rules, used simply as a tool for computing polarity.

The unary operator $\ominus$ marks downward monotone functions, while the upward monotone functions are left unmarked. The corresponding unary structural connective $\langle . \rangle^-$ marks a structure in a downward monotone argument position. The unary operator $\boxminus^{\downarrow}$ encodes negative polarity information.

The dynamic flow of information from the function to the argument is directly accounted for by the logical rules without the need of an external monotonicity marking algorithm (like the one of Sánchez). Similarly, structural rules allow to internalize the polarity algorithm producing marked structures instead of marking the corresponding nodes. An advantage of such an approach is that the marked structures are readily available for deriving monotone inference, as opposed to the Natural Logic of Sánchez, where the polarity markers were externally displayed on the structures once read off the nodes. Furthermore, NPI distribution can be controlled, since polarity information is encoded in the logical types. The logical and the structural languages are used to encode monotonicity and polarity information as shown in table 1.1.

Bernardi uses Sánchez's notion of polarity of occurrences. Also, similarly to the approaches of Sánchez and Dowty, to determine the polarity at the sentence level, she starts by assigning polarity markers to the lexical entries. The downward monotone functions are distinguished by prefixing their argument with $\boxminus^{\downarrow}$. For example, since doubt is downward monotone in its first argument, it is of type $(np\backslash s)/\boxminus^{\downarrow}s$.

The flow of information from the logical to the structural formulas is as follows:

- Application of a downward monotone function implies the propagation of the marker from the function to the argument:

$$\dfrac{\dfrac{\Delta \vdash B}{\langle\Delta\rangle^{-} \vdash \ominus B}\ \ominus I \qquad \Gamma \vdash \boxminus^{\downarrow} B\backslash A}{\langle\Delta\rangle^{-} \circ \Gamma \vdash A}\ \backslash E$$

- Functions are built by applying the introduction rules $(\backslash I)$ and $(/I)$. Upward and downward monotone functions abstract over positive and negative positions respectively:

$$\dfrac{\begin{matrix} D \\ \vdots \\ \Gamma \circ C \vdash A \end{matrix}}{\Gamma \vdash A/C}\ /I \qquad \dfrac{\begin{matrix} D \\ \vdots \\ \Gamma \circ \ominus C \vdash A \end{matrix}}{\Gamma \vdash A/\ominus C}\ /I$$

- The polarity is passed from the structure to the logical type and vice versa using the rules $(\boxminus^{\downarrow} I)$ and $(\boxminus^{\downarrow} E)$:

$$\dfrac{\langle\Delta\rangle^{-} \vdash A}{\Delta \vdash \boxminus^{\downarrow} A}\ \boxminus^{\downarrow} I \qquad \dfrac{\Delta \vdash \boxminus^{\downarrow} A}{\langle\Delta\rangle^{-} \vdash A}\ \boxminus^{\downarrow} E$$

For example, let us see how we can account for NPI distribution in the sentence Nobody left yet. The lexical assignments are as follows: yet – $\boxminus^{\downarrow} \ominus iv \backslash \boxminus^{\downarrow} \ominus iv$ (it must occur in a negative polarity position, which it passes to its argument), left – iv, nobody – $s/\ominus iv$ (downward monotone function).

$$\dfrac{nobody \vdash s/\ominus iv \qquad \dfrac{\dfrac{\dfrac{\dfrac{left \vdash iv}{\langle left\rangle^{-} \vdash \ominus iv}\ \ominus I}{left \vdash \boxminus^{\downarrow} \ominus iv}\ \boxminus^{\downarrow} I \qquad yet \vdash \boxminus^{\downarrow} \ominus iv \backslash \boxminus^{\downarrow} \ominus iv}{left \circ yet \vdash \boxminus^{\downarrow} \ominus iv}\ \backslash E}{\langle left \circ yet\rangle^{-} \vdash \ominus iv}\ \boxminus^{\downarrow} E}{nobody \circ \langle left \circ yet\rangle^{-} \vdash s}\ /E$$

However, if we replace nobody by everybody that is assigned the type s/iv, the derivation fails: the type of everybody does not provide the needed negative feature to match the type assigned to left yet.

The system of Bernardi can also deal with more complex NPI distribution, e.g. NPIs in embedded sentences or multiple negative polarity occurrences.

As opposed to both Dowty and Sánchez, Bernardi encodes monotonicity and polarity information into syntactic type assignments by means of unary operators. As opposed to Sánchez, internalizing this information allows the system to account for NPIs distribution. Due to basing her system on the multi-modal type-logical grammar, Bernardi achieves more flexibility than Dowty and is able to deal with more complex examples of NPI distribution.

1.4 Fyodorov et al.: Order-Based Inference in Natural Language (2002)

Fyodorov, Winter and Francez (henceforth FWF, see [4]) propose a Natural Logic system based on the AB calculus for deriving inferences in natural language. The main part of the system is the *Order Calculus*, which manipulates *order statements* between semantically annotated derivation trees of natural language expressions of the same (partially ordered) semantic type. An order statement between derivation trees of natural language sentences corresponds to an entailment relation between these sentences.

As opposed to the Natural Logic systems of Sánchez and Bernardi, in FWF's system there is no direct appeal to actual denotations of natural language expressions for computing inferences. Instead, the following *semantic features* that abstract the actual denotations of the expressions are used to mark the syntactic categories assigned to these expressions: '+'/'-' mark upward/downward monotonicity, 'R' marks restrictivity and 'C'/'D' mark conjunctive/disjunctive behavior. Thus, as opposed to Sánchez and Bernardi, FWF also use semantic properties other than monotonicity for deriving inferences.

The formulae of the Order Calculus are of form $\alpha_A \leq \beta_A$, where α_A, β_B are derivation trees of expressions of categories A and B resp. Below are some of the rules of the Order Calculus. Assume that B, B', B'' be syntactic categories similar up to their semantic decoration, and so are D, D'. Assume that the categories D and B are assigned the same semantic type, and so are A and C. Let $x \subseteq \{+, -, R, C, D\}$. If both $\gamma \leq \delta$ and $\delta \leq \gamma$ we write $\gamma \equiv \delta$.

Monotonicity:

$$\frac{\alpha_{B'} \leq \beta_{B''}}{\boxed{\frac{\gamma_{A/^{+}B} \quad \alpha_{B'}}{A}/E} \leq \boxed{\frac{\gamma_{A/^{+}B} \quad \beta_{B''}}{A}/E}}\ \text{MON+}$$

Function Replacement:

$$\frac{\alpha_{A/B} \leq \beta_{C/D} \quad \gamma_{B'} \equiv \delta_{D'}}{\boxed{\frac{\alpha_{A/^{x}B} \quad \gamma_{B'}}{A}/E} \leq \boxed{\frac{\beta_{C/^{x}D} \quad \delta_{D'}}{C}/E}}\ \text{FR}$$

Restrictive Modification:

$$\frac{\emptyset}{\boxed{\frac{\alpha_{B'/^{R}B} \quad \beta_{B''}}{B'}/E} \leq \beta_{B''}}$$

In addition to a calculus for deriving inferences, we need a lexicon that assigns semantically annotated syntactic categories to lexical items, according to their standard actual denotations. For example, the denotation of the determiner every is downward monotone in its first argument (given that student $\leq$ person, every person $\leq$ every student) and upward monotone in its second argument (given that walked $\leq$ moved, every person walked $\leq$ every person moved). In the lexicon every is assigned the category $(s/^{+}(s\backslash np))/^{-}n$.

Let us see an example of deriving the following inference: Every student kissed every teacher $\Rightarrow$ Every student kissed every tall teacher. For this purpose, we need to derive an order statement between the AB derivation trees of the above sentences. The derivation in the Order Calculus is as follows (we use a shortened presentation of the derivation trees):

$$\dfrac{\dfrac{\dfrac{\dfrac{\emptyset}{tall^{R}\ [teacher] \leq teacher}\ \text{RMOD}}{every^{-}[teacher] \leq every^{-}[tall - teacher]}\ \text{MON}}{[kissed]\ every - teacher \leq [kissed]\ every - tall - teacher}\ \text{FR}}{every - student^{+}\ [kissed\ every\ teacher] \leq every - student^{+}\ [kissed\ every\ tall\ teacher]}\ \text{MON+}$$

FWF's system is able to derive also inferences involving non-monotonic expressions which are reducible to conjunctions of monotonic ones, e.g. exactly three, which is reducible to at most three and at least three. In addition, FWF present a proof search procedure for the Order Calculus, which is sound and under certain restrictions on the grammar and the derivable order statements, complete.

However, despite this system's value for demonstrating a novel technique of inference in natural language, the range of the inferences the system is capable of dealing with is rather limited. An obvious reason for its limited ability is that the inference rules of the Order Calculus capture mainly inferences based on monotonicity, restrictivity and conjunction/disjunction properties of denotations of natural language expressions, while there are also other semantic properties of denotations, which cannot be expressed in the system (for example, there is no treatment of negation). Furthermore, there are other semantic factors affecting natural language inferences that have yet to be detected, and appropriate OC features - to be abstracted from them. Another important reason is related to the syntactic limitations of the AB calculus: it is not powerful enough to derive the grammaticality of many kinds of sentences, such as sentences with relative clauses, pied piping, non-constituent coordination etc. As a result, the system of FWF does not account for the following valid entailments involving such sentences:

1. The tall student, whom Mary kissed, smiled. $\Rightarrow$ The student, whom Mary kissed, smiled.

2. John, the brother of whom Mary loves, smiled. $\Rightarrow$ John smiled.

3. John does and Mary doesn't move. $\Rightarrow$ Mary doesn't walk., given walk $\leq$ move

4. Mary loves every student, No student whom Mary loves walked $\Rightarrow$ No student walked.

We conclude that the AB calculus is not powerful enough to serve as a categorial engine underlying a Natural Logic inference system. Therefore, the inference system should be based on a more complex categorial formalism, e.g. the Lambek calculus.

Chapter 2

Semantic types and order relations

2.1 Basic semantic notions

Model-theoretic semantic theories associate natural language expressions with *syntactic categories*, and their denotations with (closely related) *semantic types*. Furthermore, most expressions denote objects in partially ordered domains, so that meanings of equi-typed expressions are naturally comparable. Thus in the finite set of *primitive types* (denoted below by T^0), we distinguish its finite subset, which we call the set of *partially ordered primitive types* (denoted below by T^0_{po}), interpreted over partially ordered domains.

Definition 2.1.1 (Types) *Let T^0 be some finite set of primitive types. The set of types is the smallest set T so that $T^0 \subseteq T$ and if $\tau \in T$ and $\sigma \in T$ then also $(\tau\sigma) \in T$.*

Standardly, types e (for entities) and t (for 'truth values') are among the primitive types.

Definition 2.1.2 (PO types) *Let ${T^0}_{po} \subseteq T^0$ be some finite set of partially ordered primitive types. The set of PO types is the smallest set $T_{po} \subseteq T$ s.t. ${T^0}_{po} \subseteq T_{po}$ and if $\tau \in T$ and $\sigma \in T_{po}$ then also $(\tau\sigma) \in T_{po}$.*

Standardly, type t is among the primitive PO types.

Definition 2.1.3 (Modifer types) *A type $(\sigma\tau)$ is a modifier type iff $\sigma = \tau$.*

Definition 2.1.4 (Coordination types) *A type $(\sigma(\tau\rho))$ is a coordination type iff $\sigma = \tau = \rho$.*

Definition 2.1.5 (Domains) *For each primitive type* $\tau \in T^0$*, let* D_τ *be a non-empty domain. Let these primitive domains be mutually disjoint. For each non-primitive type* $(\tau\sigma) \in T \backslash T^0$*, let* $D_{(\tau\sigma)} = (D_\tau \to D_\sigma)$ *(the set of all functions from* D_τ *to* D_σ*).*

The domain D_σ of any primitive PO type σ is endowed with a given partial order relation $\leq_\sigma$. The partial order for complex PO types is defined as follows:

Definition 2.1.6 (Pointwise partial order) *If* σ *is a PO type with partial order* $\leq_\sigma$ *over the domain* D_σ*, then the partial order* $\leq_{(\tau\sigma)}$ *over the domain* $D_{(\tau\sigma)}$ *is defined pointwise:* $d_1 \leq_{(\tau\sigma)} d_2$ *iff for every* $d' \in D_\tau$*:* $d_1(d') \leq_\sigma d_2(d')$*.*

Next, we distinguish between several classes of functions that we will find useful in this work.

Definition 2.1.7 (Restrictive function) *For* τ *a PO type, a function of the modifier type* $\tau\tau$ $f \in D_{\tau\tau}$ *is* restrictive *iff for every* $d \in D_\tau$*:* $f(d) \leq_\tau d$*.*

For example, the denotations of adjectives like tall, pretty and adverbs like slowly, happily are analyzed as restrictive functions of type $((et)(et))$, as the denotation of the expression tall boy is less or equal to the denotation of the expression boy and the denotation of slowly move os less or equal to the denotation of move.
A special kind of restrictive functions is known as *greatest lower bound* (glb) functions. Consider the definition of the binary case, which is the most useful for our purposes.

Definition 2.1.8 (Greatest lower bound) *A function of the coordination type* $\tau(\tau\tau)$ $f \in D_{(\tau(\tau\tau))}$*, where* τ *is a PO type, is called* glb *iff for all* $d_1, d_2, d_3 \in D_\tau$ *the following two conditions hold:*

1. $(f(d_1))(d_2) \leq_\tau d_1$ *and* $(f(d_1))(d_2) \leq_\tau d_2$*;*

2. *if* $d_3 \leq_\tau d_1$ *and* $d_3 \leq_\tau d_2$ *then* $d_3 \leq_\tau (f(d_1))(d_2)$*.*

The first requirement states that f is restrictive, or returns a lower bound, on both of its arguments; the second requirement states that f returns a *greatest* lower bound on both of its arguments.
A symmetric notion is the least upper bound (lub) function.

Definition 2.1.9 (Least upper bound) *A function of the coordination type* $\tau(\tau\tau)$ $f \in D_{\tau(\tau\tau)}$*, where* τ *is a PO type, is called* lub *iff for all* $d_1, d_2, d_3 \in D_\tau$ *the following two conditions hold:*

1. $d_1 \leq_\tau (f(d_1))(d_2)$ *and* $d_2 \leq_\tau (f(d_1))(d_2)$*;*

2. *if* $d_1 \leq_\tau d_3$ *and* $d_2 \leq_\tau d_3$ *then* $(f(d_1))(d_2) \leq_\tau d_3$*.*

The first requirement states that f returns an upper bound on both of its arguments; the second requirement states that f returns a *least* upper bound on both of its arguments.

In natural language there are at least three kinds of glb functions:

1. Conjunctions: the standardly assumed meaning of conjunctions such as dance and smile, Mary danced and John smiled, and every teacher and some student is the glb of the meanings of the conjuncts.

2. Relative clauses: a 'subject oriented' relative clause such as child who sneezed is treated as a glb of the noun (child) denotation and the verb phrase (sneezed) denotation.

3. Intersective adjectives: adjectives such as blue and pregnant are often assumed to denote 'intersective functions': functions of type $((et)(et))$ that intersect their argument with an implicit argument of type (et). For instance, the nominal blue car is synonymous with the nominal car that is blue, which is formed using a glb relative.

A lub function in natural language is the disjunction or: the standardly assumed meaning of disjunctions such as dance or smile, Mary danced or John smiled, and every teacher or some student is the lub of the meanings of the disjuncts.

In addition to general ordering properties of functions like restrictiveness, glb or lub, we are also interested in more specific properties, which can be useful for describing order relations between natural language expressions. One of the most useful properties of functions in natural language is *monotonicity*.

Definition 2.1.10 (Monotonicity) *Let* σ_1 *and* σ_2 *be PO types. A function* $f \in D_{(\sigma_1\sigma_2)}$ *is:*

- upward monotone *iff for all* $d_1, d_2 \in D_{\sigma_1}$*:* $d_1 \leq_{\sigma_1} d_2 \Rightarrow f(d_1) \leq_{\sigma_2} f(d_2)$*;*
- downward monotone *iff for all* $d_1, d_2 \in D_{\sigma_1}$*:* $d_1 \leq_{\sigma_1} d_2 \Rightarrow f(d_1) \geq_{\sigma_2} f(d_2)$*.*

For example, the denotation of the determiner every is analyzed as a function of type $((et)((et)t))$ that is downward monotone w.r.t. its first argument and upward monotone w.r.t. its second argument. In this way we capture the following entailments:

- Every student ran $\Rightarrow$ Every tall student ran (assuming tall student $\leq$ student)
- Every student ran $\Rightarrow$ Every student moved (assuming moved $\leq$ ran)

2.2 Decoration of types

Now we replace the set of types defined in the previous section by a set of *decorated types*. Generally speaking, we pursue the goal of being able to manipulate order statements as purely syntactic units, with no direct appeal to models and denotations (in contrast with the Natural Logic of [9] and [1]). At the same time, we would still like to use the fact that many natural language expressions denote restrictive, monotone or other functions having special properties giving rise to entailments.

Thus we use the set of *semantic features* of [4] as an *abstraction* of the semantic properties of the actual denotations in question. These features describe semantic properties like monotonicity, restrictiveness and conjunctive/disjunctive behavior of functions. We *decorate* the types (and in the next chapter also the categories) of the natural language expressions denoting monotone etc. functions with the respective semantic features. This way we avoid dealing with models directly, but still keep track of the semantic information that is used to derive inference.

Definition 2.2.1 (The set of semantic features) *The set of semantic features is* $Feat = \{+, -, R, C, D\}$

The semantic features abstract the semantic properties of denotations of natural language expressions as follows:

- '+'/'-' marks upward/downward monotonicity
- 'R' marks restrictivity

- 'C'/'D' marks conjunction/disjunction

Definition 2.2.2 (Decorated types and decorated PO types) *The sets of decorated types and PO decorated types are the smallest sets* T_{dec}, T_{dec}^{PO} *so that:*

- $T^0 \subseteq T_{dec},\ T_{PO}^0 \subseteq T_{dec}^{PO}$
- *if* $\tau \in T_{dec}$, $\sigma \in T_{dec}$ *and* $\rho \in T_{dec}^{PO}$ *then* $(\tau^F\sigma) \in T_{dec}, (\tau^F\rho) \in T_{dec}^{PO}$, *where* $F \subseteq Feat$ *and the following conditions hold:*

  1. *If* $F \neq \emptyset$, *then* $\tau, \sigma \in T_{dec}^{PO}$.
  2. *If* $R \in F$ *then* $\tau = \sigma$.
  3. *If* C *or* $D \in F$ *then (i) If* $\tau = ({\tau_1}^{F'}\tau_2)$ *then* $F' = \emptyset$ *and (ii)* $\sigma = (\tau^{\emptyset}\tau)$.

Condition 1 guarantees that only functional types $(\tau^F\sigma)$, where both τ and σ are PO decorated types can be marked with $F \neq \emptyset$. Condition 2 guarantees that only modifier types are marked with 'R'. Condition 3 guarantees that an expression of a type marked with 'C' or 'D' is treated as denoting a binary function and all its markings are specified on the functor type.

Note that for $F = \emptyset$ we get the standard definition of *types*.

We use the pattern $(\tau^*\sigma)$ to match any of the types $(\tau^F\sigma)$ for $F \subseteq Feat$. A type τ such that all its subterms are marked with $F = \emptyset$, is denoted by τ°.

Definition 2.2.3 (Domains of decorated types) *Let* $F \subseteq Feat$. *For each non-primitive type* $(\tau^F\sigma) \in T_{dec}\backslash T^0$, *let* $D_{(\tau^F\sigma)}$ *be the set of all functions from* D_τ *to* D_σ *having the semantic properties marked by the semantic features in* F.

For example, $D_{(\sigma^+\tau)}$ is the domain of upward monotone functions from D_σ to D_τ.

For $F = \emptyset$, $D_{(\sigma^F\tau)} = D_{(\sigma\tau)}$. For σ a decorated PO type, $D_{(\tau^F\sigma)}$ inherits its partial ordering from $D_{(\tau\sigma)}$, where $\leq_{(\tau^F\sigma)}$ is the restriction of $\leq_{(\tau\sigma)}$ to $D_{(\tau^F\sigma)} \subseteq D_{(\tau\sigma)}$.

Note that *all* decorated functional types are marked, some with $F = \emptyset$.

Chapter 3

Type-Logical Categorial Grammar

In this chapter we describe the categorial formalism underlying the inference system. The type-logical categorial grammar that we use is based on $\mathcal{L}$, which is a slight modification of the (associative) Lambek calculus ([8]). There are two main differences between $\mathcal{L}$ and the standard (semantically augmented) Lambek calculus, namely:

1. Syntactic categories in $\mathcal{L}$ are *decorated* with subsets of $Feat = \{+, -, R, C, D\}$. Semantic types of proof terms (encoding $\mathcal{L}$ derivations of decorated syntactic categories) inherit the decoration from the categories.

2. In order to establish a one-to-one correspondence between $\mathcal{L}$ derivations and proof terms representing them, we use *directed lambda terms* ([12]) to encode $\mathcal{L}$ derivations. Thus each syntactic category A s.t. $\mathbf{type}(A) = \tau$ is assigned a directed lambda term φ_τ that encodes the $\mathcal{L}$-derivation of this category via the Curry-Howard isomorphism.

Note also that in the elimination rules of $\mathcal{L}$, an argument category B in a functor category (A/B) (or $A\backslash B$) is allowed to be decorated differently from the category B' that combines with the functor.

Let a mapping **type** assign (decorated) semantic types to (decorated) syntactic categories.

Definition 3.1 (Decorated syntactic categories) *Let* $\boldsymbol{CAT}^0$ *be a finite set of primitive categories s.t.* $s \in \boldsymbol{CAT}^0$. *Let* $\boldsymbol{type}^0 : \boldsymbol{CAT}^0 \rightarrow \boldsymbol{type}$ *be a* <u>*typing function*</u> *for this set s.t.* $\boldsymbol{type}^0(s) = t$. *The set of categories is the smallest set* $\boldsymbol{CAT}$ *that satisfies:*

1. $\boldsymbol{CAT}^0 \subseteq \boldsymbol{CAT}$. *For any* $A \in \boldsymbol{CAT}^0$: $\boldsymbol{type}(A) = \boldsymbol{type}^0(A)$.
2. *If* $A \in \boldsymbol{CAT}$ *and* $B \in \boldsymbol{CAT}$ *then for any* $F \subseteq Feat$ *s.t.* $(\ \boldsymbol{type}(B)^F\ \boldsymbol{type}(A)) \in T_{dec}$, $(A/^F B) \in \boldsymbol{CAT}$ *and* $(A\backslash^F B) \in \boldsymbol{CAT}$, *and* $\boldsymbol{type}((A/B)) = \boldsymbol{type}((A\backslash B)) = (\ \boldsymbol{type}(B)^F\ \boldsymbol{type}(A))$.

We assume that the set of primitive categories includes at least a designated category s for sentences. Meta variables A, B range over categories.

We use the patterns $(A/^*B), (A\backslash^*B)$ to match the categories $(A/^FB), (A\backslash^FB)$ for any $F \subseteq Feat$ resp.

Now let $\mathbf{LexVAR}_\tau$, $\mathbf{VAR}_\tau$ be some mutually disjoint countable sets of variables of type τ. Let $\mathbf{LexVAR} = \bigcup_\tau \mathbf{LexVAR}_\tau$ and $\mathbf{VAR} = \bigcup_\tau \mathbf{VAR}_\tau$ for $\tau \in T_{dec}$. Informally, the members of **LexVAR** in a proof term originate from the lexical entries and cannot be discharged by introduction rules. Members of **VAR** in a proof term correspond to undischarged assumptions in an $\mathcal{L}$ derivation.

Definition 3.2 (Directed Lambda terms) *Let Let* ψ, φ *be meta variables that range over terms and* x *a meta variable that ranges over variables from* $\boldsymbol{VAR}$. *The set* ***Terms*** *is the smallest set s.t.:*

- ***LexVAR***, ***VAR***$\subseteq$***Terms***
- *If* $\varphi_{(\sigma\tau)}, \psi_\sigma \in$***Terms***, *then* $(\varphi(\psi))_\tau, ([\psi]\varphi)_\tau \in$***Terms*** *(right/left application)*
- *If* $x_\sigma \in \boldsymbol{VAR}, \varphi_\tau \in$***Terms***, *then* $(\overrightarrow{\lambda} x.\varphi)_{(\sigma\tau)}, (\overleftarrow{\lambda} x.\varphi)_{(\sigma\tau)} \in$***Terms*** *(right/left abstraction)*

See definition 4.2.1 for the semantics of directed lambda terms.

We now define *sets of free variables* and *sequences of free variables* in directed lambda terms. Sequences of free variables will be used to classify an important subset of all lambda terms - the *peripherally-linear* (PL) terms, which correspond exactly to $\mathcal{L}$ derivations and which will become especially useful in the sequel.

For two sequences $\overline{\psi}$ and $\overline{\phi}$, we take $\overline{\psi} \cdot \overline{\varphi}$ to denote their concatenation.

Definition 3.3 (Free variables) *For every* $\psi \in$***Terms***, *the set* $Free(\psi)$ *and the sequence* $\overline{Free}(\psi)$ *of free variables are defined as follows:*

- *For* $x \in \boldsymbol{VAR} \cup \boldsymbol{LexVAR}$, $\overline{Free}(x) =< x >,\ Free(x) = \{x\}$
- $Free(\psi(\varphi)) = Free(\psi) \cup Free(\varphi)$, $\overline{Free}(\psi(\varphi)) = \overline{Free}(\psi) \cdot \overline{Free}(\varphi)$
- $Free([\varphi]\psi) = Free(\psi) \cup Free(\varphi)$, $\overline{Free}([\varphi]\psi) = \overline{Free}(\varphi) \cdot \overline{Free}(\psi)$
- $Free(\overrightarrow{\lambda} x.\psi) = Free(\psi) - \{x\}$, $\overline{Free}(\overrightarrow{\lambda} x.\psi) = \overline{Free}(\overrightarrow{\lambda} x.\psi) - < x >$ *(removing* all *occurrences of* x *from the sequence)*
- $Free(\overleftarrow{\lambda} x.\psi) = Free(\psi) - \{x\}$, $\overline{Free}(\overleftarrow{\lambda} x.\psi) = \overline{Free}(\overleftarrow{\lambda} x.\psi) - < x >$ *(removing* all *occurrences of* x *from the sequence)*

In the last two clauses, the abstraction operators $\overrightarrow{\lambda}$ and $\overleftarrow{\lambda}$ *bind* all free occurrences of x in ψ by removing (all occurrences of) x from the set (sequence) of the free variables of ψ. Every such occurrence of x is *bound*. If an occurrence of x is not bound, then we say that it is *free*.

Now we define PL terms, a subset of all directed lambda terms that represent Lambek derivations via the Curry-Howard correspondence. This set of terms plays the same role as the set of linear terms plays with respect to the non-directional implicational fragment.

Definition 3.4 (PL terms) *Let* ***PLTerms*** *be the smallest subset of* ***Terms*** *s.t.:*

- *Each subterm of* ψ $\in$***PLTerms*** *contains a free occurrence of a variable from* ***VAR***$\cup$***LexVAR***.
- *No subterm of* ψ $\in$***PLTerms*** *contains more than one free occurrence of a variable from* ***VAR***.
- *Each occurrence of the* $\overrightarrow{\lambda}$ $(\overleftarrow{\lambda})$*-abstractor in* ψ $\in$***PLTerms*** *binds a variable within its scope, and this variable is right (left)-peripheral in* $\overline{Free}(\psi)$.

Definition 3.5 (The set of subterms) *Let* $\psi \in$ ***PLTerms***. *The set of subterms of* ψ *(*$ST(\psi)$*) is defined recursively: for* $\psi \in$ ***VAR*** $\cup$ ***LexVAR*** $ST(\psi) = \{\psi\}$, *for* $\psi = \varphi\langle\phi\rangle$ $ST(\psi) = \{\psi\} \cup ST(\varphi) \cup ST(\phi)$, *for* $\psi = \overline{\lambda}x.\phi$ $ST(\psi) = \{\psi\} \cup \{\phi\} \cup ST(\phi)$.

Definition 3.6 (Substitution) *Let* $\alpha \in$ ***PLTerms*** *s.t.* $x_\tau \in$ ***VAR*** *is in* $Free(\alpha)$. *Let* γ_τ *be a term s.t. no free variables of* γ_τ *occur bound in* α. *Then the term* $\alpha[x/\gamma]$ *is obtained from* α *by substituting all occurrences of* x *by* γ.

Henceforth we use the following notation:

- A variable $w_\tau \in \mathbf{LexVAR}$ is assigned to a lexical item **w** of (decorated) type τ.
- A variable $x_\tau \in \mathbf{VAR}$ is assigned to a dischargeable assumption x of type τ.
- We use the pattern $\psi\langle\phi\rangle$ for $\psi(\phi)$ or $[\phi]\psi$ and the pattern $\overline{\lambda}x.\psi$ for $\overrightarrow{\lambda}x.\psi$ or $\overleftarrow{\lambda}x.\psi$.
- A term φ_τ s.t. $x_\sigma \in \mathbf{VAR}$ is the rightmost (leftmost) variable in $\overline{Free}(\varphi)$ is denoted by $\varphi_\tau^{\overrightarrow{x}_\sigma}$ ($\varphi_\tau^{\overleftarrow{x}_\sigma}$), both abbreviated to $\varphi_\tau^{x_\sigma}$.

Definition 3.7 (Formally equivalent categories and types):

- *For any two decorated categories A, B, A is formally equivalent to B, denoted by $A \equiv_f B$, iff (i) A and B are primitive and $A = B$ or (ii) $A = (C/^*D)$ (or $A = (C\backslash^*D)$), $B = (C'/^*D')$ (or $B = (C'\backslash^*D')$), $C \equiv_f C'$, $D \equiv_f D'$.*
- *For any two decorated types τ, σ, τ is formally equivalent to σ, denoted by $\tau \equiv_f \sigma$, iff (i) τ and σ are primitive and $\tau = \sigma$ or (ii) $\tau = (\tau_1{}^*\tau_2)$, $\sigma = (\sigma_1{}^*\sigma_2)$, $\tau_1 \equiv_f \sigma_1$, $\tau_2 \equiv_f \sigma_2$.*

In words, formally equivalent categories (types) are equal up to their decoration. Note that the types of formally equivalent categories are also formally equivalent.

Definition 3.8 *($\mathcal{L}$) Let $\Gamma, \Gamma_1, \Gamma_2$ range over finite non-empty sequences of pairs $A : \psi_\tau$, where A is a (decorated) syntactic category and ψ_i a directed lambda term of a (decorated) type τ. Let $\tau, \tau_1, \tau_2, ...$ range over decorated types. The notation $\vdash_{\mathcal{L}-OC} \Gamma \triangleright A : \psi_\tau$ means that the sequence Γ is $\mathcal{L}$-reducible to $A : \psi_\tau$. The rules of $\mathcal{L}$ are as follows :*

$$(axiom_1) A : x_\tau \triangleright A : x_\tau \quad for\ x_\tau \in \boldsymbol{VAR}$$

$$(axiom_2) B : w_\tau \triangleright B : w_\tau \quad for\ w_\tau \in \boldsymbol{LexVAR}$$

Elimination rules:

$$for\ B \equiv_f B', \tau_1 \equiv_f \tau_1' :$$

$$(/E)\frac{\Gamma_1 \triangleright (A/^*B) : \psi_{(\tau_1{}^*\tau_2)} \quad \Gamma_2 \triangleright B' : \varphi_{\tau_1'}}{\Gamma_1\Gamma_2 \triangleright A : (\psi_{(\tau_1{}^*\tau_2)}(\varphi_{\tau_1'}))_{\tau_2}}, \quad (\backslash E)\frac{\Gamma_2 \triangleright B' : \varphi_{\tau_1'} \quad \Gamma_1 \triangleright (A\backslash^*B) : \psi_{(\tau_1{}^*\tau_2)}}{\Gamma_2\Gamma_1 \triangleright A : ([\varphi_{\tau_1'}]\psi_{(\tau_1{}^*\tau_2)})_{\tau_2}}$$

Introduction rules:

$$for\ \Gamma_1\ not\ empty,\ \ x_{\tau_1} \in \boldsymbol{VAR}$$

$$(/I)\frac{\Gamma_1,\ B : x_{\tau_1} \triangleright A : \psi_{\tau_2}^{\overrightarrow{x_{\tau_1}}}}{\Gamma_1 \triangleright (A/B) : (\overrightarrow{\lambda} x_{\tau_1}.\psi_{\tau_2}^{\overrightarrow{x_{\tau_1}}})_{(\tau_1\tau_2)}} \quad (\backslash I)\frac{B : x_{\tau_1},\ \Gamma_1 \triangleright A : \psi_{\tau_2}^{\overleftarrow{x_{\tau_1}}}}{\Gamma_1 \triangleright (A\backslash B) : (\overleftarrow{\lambda} x_{\tau_1}.\psi_{\tau_2}^{\overleftarrow{x_{\tau_1}}})_{(\tau_1\tau_2)}}$$

It can be shown by induction on $\mathcal{L}$ derivations, that the class of PL terms corresponds exactly to proof terms representing $\mathcal{L}$ derivations (see [2] for the case of undirected PL terms).

Definition 3.9 (Type-Logical Categorial Grammar) *A type-logical categorial grammar is a tuple* $\langle \Sigma, \boldsymbol{CAT}^0, A_0, \alpha \rangle$*, where* Σ *is the alphabet,* $\boldsymbol{CAT}^0$ *is the set of primitive categories,* A_0 *is the target category and* $\alpha : \Sigma \rightarrow 2^{\boldsymbol{CAT}}$ *is an assignment of sets of categories to lexical items.*

Standardly, A_0 is taken to be s, a category designated for NL sentences. We will refer to an assignment α as a *lexicon*.

Definition 3.10 (Language generated by G) *Let* $G = \langle \Sigma, \boldsymbol{CAT}^0, B, A_0, \alpha \rangle$ *be a type-logical categorial grammar s.t.* ***type***(A_0)=τ*. Then the language generated by* G *is defined as follows:*

$$L[G] = \{w = w_1...w_n \in \Sigma^* \mid \exists A_1...A_n :\ A_i \in \alpha(w_i)\ for\ i = 1...n,\ and\ \vdash_{\mathcal{L}} A_1...A_n \triangleright A_0 : \psi_\tau\}$$

where ψ_τ *is the lambda term encoding the derivation.*

Chapter 4

The $\mathcal{L}$-based Order Calculus

In this chapter we introduce the main part of the system – the $\mathcal{L}$-based Order Calculus ($\mathcal{L}$-OC). $\mathcal{L}$-OC manipulates *order statements* between proof terms (of formally equivalent types) representing $\mathcal{L}$ derivations of natural language expressions. Thus the formulae of $\mathcal{L}$-OC are order statements of form $\varphi_\tau \leq_{\tau^\circ} \psi_{\tau'}$ for $\tau \equiv_f \tau'$. It should be stressed that the $\mathcal{L}$-OC order statements are purely syntactic units and although lexical semantic markings are abstracted from standard denotations, there is no direct appeal to models (in contrast to the works of [9] and [1]). Also, '$\leq_\tau$' is treated as a *syntactic* relation between proof terms, and not as a partial order relation between their denotations.

4.1 $\mathcal{L}$-OC

If an order statement $\varphi_\tau \leq_{\tau^\circ} \psi_{\tau'}$ is provable in $\mathcal{L}$-OC we denote this by $\vdash_{\mathcal{L}-OC} \varphi_\tau \leq_{\tau^\circ} \psi_{\tau'}$. When both $\vdash_{\mathcal{L}-OC} \varphi_\tau \leq_{\tau^\circ} \psi_{\tau'}$ and $\vdash_{\mathcal{L}-OC} \psi_\tau \leq_{\tau^\circ} \varphi_\tau$ for $\tau \equiv_f \tau'$, we denote this by $\vdash_{\mathcal{L}-OC} \varphi_\tau \equiv_{\tau^\circ} \psi_{\tau'}$.

Definition 4.1.1 ***($\mathcal{L}$-OC :)***

$$\text{For } \tau \equiv_f \tau' \equiv_f \hat{\tau} \equiv_f \tilde{\tau},\ \rho \equiv_f \rho' \equiv_f \hat{\rho}\ ^1:$$

$$\textbf{(REFL)}\frac{\emptyset}{\alpha_\tau \leq_{\tau^\circ} \alpha_{\tau'}} \quad \textbf{(TRANS)}\frac{\alpha_\tau \leq_{\tau^\circ} \delta_{\tau'} \quad \delta_{\tau'} \leq_{\tau^\circ} \gamma_{\hat{\tau}}}{\alpha_\tau \leq_{\tau^\circ} \gamma_{\hat{\tau}}}$$

[1]Note that τ° is equal to τ without any semantic decorations, thus $\tau^\circ \equiv_f \tau \equiv_f \tau' \equiv_f \hat{\tau} \equiv_f \tilde{\tau}$ and $\leq_{\tau^\circ}$ is compatible with $\leq_\tau$, $\leq_{\tau'}$, $\leq_{\hat{\tau}}$ and $\leq_{\tilde{\tau}}$. The case for ρ° is similar.

$$\textbf{(MON+)}\frac{\alpha_\tau \;\; \leq_{\tau^\circ} \;\; \delta_{\tau'}}{\gamma_{(\hat{\tau}^+\rho)}\langle\alpha_\tau\rangle \;\; \leq_{\rho^\circ} \;\; \gamma_{(\hat{\tau}^+\rho)}\;\langle\delta_{\tau'}\rangle} \quad \textbf{(MON-)}\frac{\delta_{\tau'} \;\; \leq_{\tau^\circ} \;\; \alpha_\tau}{\gamma_{(\hat{\tau}^-\rho)}\langle\alpha_\tau\rangle \;\; \leq_{\rho^\circ} \;\; \gamma_{(\hat{\tau}^-\rho)}\;\langle\delta_{\tau'}\rangle}$$

$$\textbf{(FR)}\frac{\alpha_{(\tau^*\rho)} \;\; \leq_{(\tau\rho)^\circ} \;\; \psi_{(\tau'^*\rho')} \;\;\; \gamma_{\hat{\tau}} \;\; \equiv_{\tau^\circ} \;\; \delta_{\tilde{\tau}}}{\alpha_{(\tau^*\rho)}\;\langle\gamma_{\hat{\tau}}\rangle \;\;\; \leq_{\rho^\circ} \;\;\; \psi_{(\tau'^*\rho')}\;\langle\delta_{\tilde{\tau}}\rangle} \quad \textbf{(RMOD)}\frac{\emptyset}{\alpha_{(\tau^R\tau')}\;\langle\gamma_{\hat{\tau}}\rangle \;\; \leq_{\tau^\circ} \;\; \gamma_{\hat{\tau}}}$$

$$\textbf{(Ab)}\frac{\alpha_\rho^{x_\tau} \;\; \leq_{\rho^\circ} \;\; \gamma_{\rho'}^{x_{\tau'}}}{\overline{\lambda} x_\tau.\alpha_\rho{}^{x_\tau} \;\; \leq_{(\tau\rho)^\circ} \;\; \overline{\lambda} x_{\tau'}.\gamma_\rho{}^{x_{\tau'}}}$$

$\overline{\lambda}x.\alpha^x, \overline{\lambda}x.\gamma^x$ *contain at least one free variable from* $\mathbf{VAR} \cup \mathbf{LexVAR}$

$$(\mathbf{C}_1)\frac{\emptyset}{([\gamma_{\tau'}]\delta_{(\tau^C(\tau\tau))}) \;\; (\psi_{\hat{\tau}}) \;\; \leq_{\tau^\circ} \;\; \Omega} \qquad (\mathbf{C}_2)\frac{\alpha_{\tilde{\tau}} \;\; \leq_{\tau^\circ} \;\; \psi_{\tau'} \quad \alpha_{\tilde{\tau}} \;\; \leq_{\tau^\circ} \;\; \gamma_{\hat{\tau}}}{\alpha_{\tilde{\tau}} \;\; \leq_{\tau^\circ} \;\; ([\gamma_{\hat{\tau}}]\delta_{(\tau^C(\tau\tau))})\;(\psi_{\tau'})}$$

$\Omega = \psi_{\hat{\tau}}$ *or* $\Omega = \gamma_{\tau'}$, α, ψ, γ *do not contain free variables from* $\mathbf{VAR}$

$$(\mathbf{D}_1)\frac{\emptyset}{\Omega \;\; \leq_{\tau^\circ} \;\; ([\gamma_{\tau'}]\delta_{(\tau^D(\tau\tau))})\;(\psi_{\hat{\tau}})} \qquad (\mathbf{D}_2)\frac{\psi_{\tau'} \;\; \leq_{\tau^\circ} \;\; \alpha_{\hat{\tau}} \quad \gamma_{\tilde{\tau}} \;\; \leq_{\tau^\circ} \;\; \alpha_{\hat{\tau}}}{([\gamma_{\tilde{\tau}}]\delta_{(\tau^D(\tau\tau))})\;(\psi_\tau) \;\;\; \leq_{\tau^\circ} \;\; \alpha_{\hat{\tau}}}$$

$\Omega = \psi_{\tau'}$ *or* $\Omega = \gamma_{\tilde{\tau}}$, α, ψ, γ *do not contain free variables from* $\mathbf{VAR}$

Normalization axioms:

$$(\beta)\frac{\emptyset}{(\phi_\tau^{y_\rho}[y_\rho/\gamma_{\rho'}])_\tau \equiv_{\tau^\circ} (\overline{\lambda} y_\rho.\phi_\tau^{y_\rho})_{(\rho\tau)}\langle\gamma_{\rho'}\rangle} \quad (\eta)\frac{\emptyset}{\psi_{(\tau^*\rho)} \equiv_{(\tau\rho)^\circ} (\overline{\lambda} x_\tau.\psi_{(\tau^*\rho)}\langle x_\tau\rangle)_{(\tau\rho)}}$$

$$x_\tau \in \mathbf{VAR}, x_\tau \notin Free(\psi)$$

Henceforth, when types are clear from context or are implicitly universally quantified over, $\leq_{\tau^\circ}$ is abbreviated to $\leq$.

The Reflexivity and Transitivity rules are general properties of the partial order relation. The RMOD rule (restrictive modification) is for deriving order statements involving terms with types marked for restrictivity. For example:

$$tall_{((et)^R(et))}(student_{(et)}) \leq student_{(et)}$$

The Upward Monotonicity rule (MON+) derives an order statement between proof terms $\gamma_{(\tau^+\sigma)}\langle\alpha_\tau\rangle$ and $\gamma_{(\tau^+\sigma)}\langle\delta\rangle$, given an order statement $\alpha \leq \delta$ between proof terms to which $\gamma_{(\tau^+\sigma)}$, the type of which is marked for upward monotonicity, is applied. The Downward Monotonicity rule (MON-) is symmetric.

The FR rule handles a situation where we have an order statement between functional terms $\alpha_{(\tau\sigma)}$ and $\delta_{(\tau\sigma)}$, and both $\gamma \leq \gamma'$ and $\gamma' \leq \gamma$ are provable. It captures the definition of the pointwise partial order.

C1 and C2 specify Conjunctive items (e.g. and) as *greatest lower bounds* with respect to the partial order relation. Similarly, D1 and D2 specify Disjunctive items (e.g. or) as *least upper bounds*. Note that these rules and axioms are not allowed to have terms with free variables from **VAR** in their conclusion. This limitation prevents the creation of non-PL terms in $\mathcal{L}$-OC proofs [2]. Without it, the following would be a valid $\mathcal{L}$-OC proof ($\varphi, \psi, \gamma \in$ **LexVAR** and $x \in$ **VAR**):

$$\frac{\gamma(x) \leq \varphi(x) \quad \gamma(x) \leq \psi(x)}{\gamma(x) \leq \underline{([\varphi(x)]\alpha_{(\tau^{C}(\tau\tau))})(\psi(x))}}\ \text{C2}$$

The underlined term is not PL, as it contains more than one occurrence of the free variable $x \in$ **VAR**.

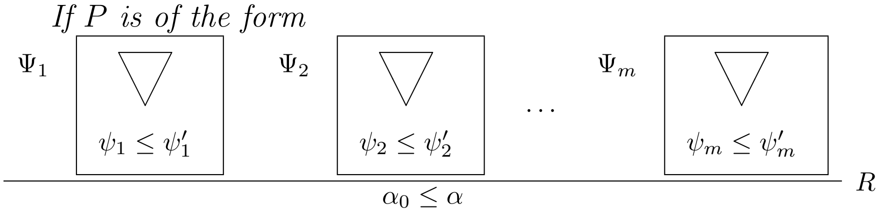
To handle abstraction proof terms in $\mathcal{L}$, we define the rule Ab, which captures discharging an assumption in a $\mathcal{L}$-derivation. Given a premise $\varphi_1^x \leq \varphi_2^x$, where both φ_1 and φ_1 represent derivation trees with a rightmost (leftmost) free variable x, the order statement $\overline{\lambda} x.\varphi_1 \leq \overline{\lambda} x.\varphi_2$ is derived. Note that if x is the *only* free variable in φ_1^x or φ_2^x, the application of Ab is not allowed. It prevents the creation of non-PL terms like $\overline{\lambda} x.x$.

The normalization axioms (β) and (η) capture β/η reduction of proof terms. The application of these axioms is discussed in detail in chapter 6.

Definition 4.1.2 (Size of $\mathcal{L}$-OC proof) *The* size n *of a $\mathcal{L}$-OC proof P of $\alpha_0 \leq \alpha$ is calculated as follows:*

If P is of the form $\dfrac{\emptyset}{\delta \leq \delta'}\ R$*, where R is an axiom, then $n = 1$.*

If P is of the form



where $m > 0$ and the sizes of $\Psi_1, \ldots, \Psi_m$ are $n_1, \ldots, n_m$ respectively, then $n = 1 + \sum_{i=1}^{m} n_i$.

[2] We chose to apply this limitation in order to prevent the system from dealing with order statements between lambda terms that do not correspond to $\mathcal{L}$ derivations. The assumption that $\mathcal{L}$-OC manipulates PL terms only makes it easier to define a proof search procedure for $\mathcal{L}$-OC as will be shown in chapter 7.

4.2 The semantics of $\mathcal{L}$-OC

The *semantics* of $\mathcal{L}$-OC is naturally defined using standard models for the extensional fragment of Montague's IL [5] and a pointwise definition for a semantic order relation $\leq_\tau$. A model M is some (non-empty) domain D_e. Any proof term φ_τ is associated with a *denotation* $[\![\varphi_\tau]\!]_{M,g}$ relative to a model M and an assignment function g, which assigns to a variable (of decorated type τ) $\alpha_\tau \in \mathbf{VAR} \cup \mathbf{LexVAR}$ some element of D_τ.

Definition 4.2.1 (Denotations of proof terms) *Let M be a model and g an assignment function. For a given proof term ψ_τ, the denotation $[\![\psi_\tau]\!]_{M,g}$ is defined as follows:*

- *If $\psi_\tau \in$ **VAR** $\cup$ **LexVAR**, then $[\![\psi_\tau]\!]_{M,g} = g(\psi_\tau)$.*
- *If $\psi_\tau = \varphi_{(\sigma^*\tau)}\langle\phi_\sigma\rangle$, then $[\![\psi_\tau]\!]_{M,g} = [\![\varphi_{(\sigma^*\tau)}]\!]_{M,g}([\![\phi_\sigma]\!]_{M,g})$.*
- *If $\psi_\tau = \overline{\lambda} x_\sigma.\varphi_\rho$, then $[\![\psi_\tau]\!]_{M,g}$ is that function $h \in D_\tau$ s.t. for all $d \in D_\sigma$: $h(d) = [\![\psi_\tau]\!]_{M,g[x:=d]}$, where $g[x := d]$ is an assignment function similar to g, except that it assigns d_σ to x_σ.*

Definition 4.2.2 (Semantics of order statements) *Let φ_1, φ_2 be terms of (decorated) type τ and g an assignment function. Then: (i)$M, g \models \varphi_1 \leq_\tau \varphi_2$ iff $[\![\varphi_1]\!]_{M,g} \leq_\tau [\![\varphi_2]\!]_{M,g}$ (ii) $M \models \varphi_1 \leq_\tau \varphi_2$ iff $\forall g$: $M, g \models \varphi_1 \leq_\tau \varphi_2$.*

Lemma 4.2.1 ***(Strong soundness of $\mathcal{L}$-OC)*** *Let α_τ, γ_τ be terms.*

$$\vdash_{\mathcal{L}-OC} \alpha_\tau \leq_\tau \gamma_\tau \;\Rightarrow\; \forall M, g : [\![\alpha_\tau]\!]_{M,g} \leq_\tau [\![\gamma_\tau]\!]_{M,g}$$

Proof: by showing that $\mathcal{L}$-OC rules and axioms are strongly sound, that is if the premise order statements of $\mathcal{L}$-OC rules are satisfied for every model, then so is the conclusion order statement. Let us show the proof for the rules MON+ and Ab and for the axiom β:

- **MON+**: Assume that for every model M

$$(*)\; M \models \alpha_\tau \leq_\tau \gamma_\tau$$

We have to show that every model M satisfies the order statement $\psi_{(\tau^+\sigma)}(\alpha_\tau) \leq_\sigma \psi_{(\tau^+\sigma)}(\gamma_\tau)$.

$\psi_{(\tau^+\sigma)}$ denotes an upward monotone function, so that

$$\forall d_1, d_2 \; s.t. \; d_1 \leq_\tau d_2 \in D_\tau, \; \forall g : \; [\![\psi]\!]_{M,g}(d_1) \leq_\sigma [\![\psi]\!]_{M,g}(d_2)$$

From (*) follows

$$\forall g : \; [\![\alpha]\!]_{M,g} \leq [\![\gamma]\!]_{M,g}$$

Then

$$\forall g : \; [\![\psi(\alpha)]\!]_{M,g} = [\![\psi]\!]_{M,g}([\![\alpha]\!]_{M,g}) \leq_\sigma [\![\psi]\!]_{M,g}([\![\gamma]\!]_{M,g}) = [\![\psi(\gamma)]\!]_{M,g}$$

It follows that every model M satisfies the order statement $\psi_{(\tau^+\sigma)}(\alpha_\tau) \leq_\sigma \psi_{(\tau^+\sigma)}(\gamma_\tau)$.

- **Ab**: Assume that for every model M: $M \models \alpha_\tau^{x_\sigma} \leq \delta_{\tau'}^{x_\sigma}$.

  Then, by definition 4.2.2: $(**)\forall g : \; [\![\alpha^{x_\sigma}]\!]_{M,g} \leq_\tau [\![\delta^{x_\sigma}]\!]_{M,g}$. The denotation of $\overline{\lambda} x_\sigma.\alpha_\tau^{x_\sigma}$ is defined as follows:

$$\forall g, \; \forall d \in D_\sigma : \; [\![\overline{\lambda} x.\alpha]\!]_{M,g}(d) = [\![\alpha]\!]_{M,g[x:=d]}$$

  By $(**)$, we have:

$$\forall d \in D_\sigma, \; \forall g : \; [\![\alpha]\!]_{M,g[x:=d]} \leq_\tau [\![\delta]\!]_{M,g[x:=d]} \Rightarrow [\![\overline{\lambda} x.\alpha]\!]_{M,g}(d) \leq_\tau [\![\overline{\lambda} x.\delta]\!]_{M,g}(d)$$

  Then by definition 2.1.6:

$$\forall g : \; [\![\overline{\lambda} x.\alpha]\!]_{M,g} \leq_{(\sigma\tau)} [\![\overline{\lambda} x.\delta]\!]_{M,g}$$

  It follows that every model M satisfies the order statement $\overline{\lambda} x_\sigma.\alpha_\tau \leq_{(\sigma\tau)} \overline{\lambda} x_\sigma.\delta_\tau$.

- β: If a term α_τ β-reduces to α'_τ, then

$$\forall M, g : \; [\![\alpha_\tau]\!]_{M,g} = [\![\alpha'_\tau]\!]_{M,g}$$

- Similarly for η.

For the other rules, the soundness proof of [4] can be easily adapted to their soundness in our formulation.

Chapter 5

$\mathcal{L}$-OC as an inference system for natural language

We now illustrate how $\mathcal{L}$-OC can be used for deriving inferences in natural language. First of all, we introduce a toy lexicon which is used for defining a small fragment of English. Then we define a way to represent natural language assertions as $\mathcal{L}$-OC order statements. In addition, we extend the every postulate introduced by [4] in order to expand the range of inferences derived by the system. We also introduce some complex non-logical axioms. Finally, we present examples of deriving inferences with sentences involving relative clauses and pied piping, as well as inferences using the extended every postulate.

5.1 The lexicon

An important part of a type-logical categorial grammar used to define the language generated by it is the lexicon, or an assignment $\alpha : \Sigma \rightarrow 2^{\mathbf{CAT}}$. In table 5.1 we introduce a toy lexicon for a fragment of English, including the decorated types assigned to the syntactic categories.

We use W^T – a fictitious word, which is assigned a proof term $w_t^T \in \mathbf{LexVAR}$, to represent a natural language assertion S (indicative sentence) as the $\mathcal{L}$-OC order statement $w_t^T \leq_t \psi_t^S$, where ψ_t^S is a proof term representing $\mathcal{L}$-derivation of S and w_t^T is the proof term, the denotation of which is defined to be the truth value **true**.

Word	Category	Type
W^T	s	t
every	$((s/^+(s\backslash np))/^-n),((s\backslash^+(s/np))/^-n)$	$((et)^-((et)^+t))$
no	$((s/^-(s\backslash np))/^-n),((s\backslash^-(s/np))/^-n)$	$((et)^-((et)^-t))$
some	$((s/^+(s\backslash np))/^+n),((s\backslash^+(s/np))/^+n)$	$((et)^+((et)^+t))$
student,boy	n	(et)
walk,walked,smile, smiled, move, moved	$(s\backslash np)$	(et)
touched, loved	$((s\backslash np)/np)$	$(e(et))$
tall, nice, smart, intelligent,creative	$(n/^Rn)$	$(et)^R(et)$
Mary,John	$(s/^+(s\backslash np)),(s\backslash^+(s/np))$	$((et)^+t)$
does	$((s\backslash np)/^+(s\backslash np))$	$(et)^+(et)$
doesn't	$((s\backslash np)/^-(s\backslash np))$	$(et)^-(et)$
whom	$((n\backslash n)/^C(s/np))$	$(et)^C((et)(et))$
and	$((s\backslash s)/s),(((s\backslash np)\backslash^C(s\backslash np))/(s\backslash np))$	$(t^C(tt)),((et)^C((et)(et))),$ $((et)t)^C(((et)t)((et)t))$

Table 5.1: The lexicon

5.2 Natural Logic inferences

In general, we represent Natural Logic inferences in $\mathcal{L}$-OC as follows.

Definition 5.2.1 *($\vdash_{NatLog}$) Let G be some categorial grammar and $S, S_1, ..., S_n \in L[G]$. Let $\alpha_t^S, \alpha_t^{S_1}, ..., \alpha_t^{S_n}$ be proof terms representing $\mathcal{L}$-derivation trees of $S, S_1, ..., S_n$ resp. Then $S_1, ..., S_n \vdash_{NatLog} S$ iff $\vdash_{\mathcal{L}-OC} w_t^T \leq \alpha_t^{S_1}, ..., \vdash_{\mathcal{L}-OC} w_t^T \leq \alpha_t^{S_n}$ implies $\vdash_{\mathcal{L}-OC} w_t^T \leq \alpha_t^S$.*

Note that in order to prove $S_1 \vdash_{NatLog} S_2$, it is enough to show $\vdash_{\mathcal{L}-OC} \alpha_1 \leq \alpha_2$, where α_1, α_2 are proof terms representing derivations of S_1, S_2 resp., and the rest follows from transitivity. We do so in all the following examples to shorten the presentation.

It is also important to note that one of the sentences S_1, S_2 may have more than one semantic reading. As a result there may be more then one normal $\mathcal{L}$ derivation of the ambiguous sentence. Furthermore, suppose that α^{S_1} and α'^{S_1} are proof terms representing different normal derivations of S_1 and α^{S_2} and α'_{S_2} - proof terms representing different normal derivations of S_2. It is possible that $\alpha^{S_1} \vdash_{\mathcal{L}-OC} \alpha^{S_2}$, but $\alpha'^{S_1} \nvdash_{\mathcal{L}-OC} \alpha'^{S_2}$. In this case $S_1 \vdash_{NatLog} S_2$ means that there *exist* such proof terms α''^{S_1} and α''^{S_2} representing derivations of S_1 and S_2 resp., that $\alpha''^{S_1} \vdash_{\mathcal{L}-OC} \alpha''^{S_2}$. We resort to this existentially quantified interpretation of $' \vdash'_{NatLog}$ in order not to complicate further the notation by introducing notations for *readings* of natural language sentences. The latter, though more complicated, would yield a more accurate interpretation. This subject is further discussed in section 6.2.

$$\boxed{\textbf{Input: Sentences } S, S_1, ..., S_n} \Rightarrow \boxed{\textbf{Find } \mathcal{L} \textbf{ proof terms } \psi^S, \psi_t^{S_1}, ..., \psi_t^{S_n} \textbf{ for } S, S_1, ..., S_n} \Rightarrow$$

$$\boxed{\textbf{Prove } \vdash_{\mathcal{L}-OC} w^T \leq \psi^S \textbf{ from } \vdash_{\mathcal{L}-OC} w^T \leq \psi^{S_1}, ..., \vdash_{\mathcal{L}-OC} w^T \leq \psi^{S_n}}$$

Figure 5.1: Deriving $S_1, ..., S_n \vdash_{NatLog} S$ in the system

As a summary, we present the process of deriving $S_1, ..., S_n \vdash_{NatLog} S$ in fig. 5.1 (assuming for simplicity of presentation that $S_1, ..., S_n$ are not ambiguous).

5.3 Non-logical axioms and the 'every' postulate

Non-logical axioms are order statements that reflect our assumptions on the denotations of natural language expressions. For example, in models that we consider, a student is also a person, and a walking object is a moving object. We can go further and claim that in the relevant models a creative intelligent X is a smart X.

Here we postulate some non-logical axioms of $\mathcal{L}$-OC useful in examples of $\mathcal{L}$-OC proofs introduced below.

$$\frac{\emptyset}{walked_{(et)} \leq moved_{(et)}}\, a_1 \quad \frac{\emptyset}{walk_{(et)} \leq move_{(et)}}\, a_2 \quad \frac{\emptyset}{kissed_{(et)} \leq touched_{(et)}}\, a_3$$

$$\frac{\emptyset}{student_{(et)} \leq person_{(et)}}\, a_4 \quad \frac{\emptyset}{\overrightarrow{\lambda} x_{(et)}.creative_{((et)^R(et))}(intelligent_{((et)^R(et))}(x_{(et)})) \leq smart_{((et)^R(et))}}\, a_5$$

$$\frac{\emptyset}{\overrightarrow{\lambda} x_e.passionately_{((et)^R(et))}(loves_{(e(et))}(x_e)) \leq adores_{(e(et))}}\, a_6$$

Note that one of the important advantages of our $\mathcal{L}$-based system (as opposed to [4]'s AB-based system) is the ability to define such complex non-logical axioms as a_5 and a_6. For example, one of the terms involved in a_5 is a composition of two functional terms $creative_{((et)^R(et))}$ and $intelligent_{((et)^R(et))}$, which could not be derived in the less powerful AB calculus. Hence, moving to $\mathcal{L}$ enriches the variety of non-logical axioms of the inference system.

In addition, it allows us to extend the ad hoc postulate that [4] defines for the determiner 'every'. According to it, the determiner 'every' induces an order statement between its two arguments. For example, from the order statement

$$w^T \leq (every_{(et)^-((et)^+t)}\ (student))\ (smiled)$$

the order statement $student_{(et)} \leq smiled_{(et)}$ is induced. However, [4] cannot handle a similar case when 'every' is in an object position. For example, the fact that the order statement

$$(*)\ student_{(et)}\ \leq\ \overrightarrow{\lambda} x_e.(Mary_{((et)^+t)}(kissed_{(e(et))}(x_e)))$$

should be induced from the proof term

$$[\overrightarrow{\lambda} x_e.Mary_{((et)^+t)}(kissed_{(e(et))}(x_e))](every_{(et)^-((et)^+t)}(student_{(et)}))$$

cannot be accounted for by [4]. We define the generalized postulate for 'every' as follows:

$$\boxed{\frac{w_t^T\ \leq\ (every_{(et)^-((et)^+t)}\ (\alpha_{(et)}))\ (\gamma_{(et)})}{\alpha_{(et)}\ \leq\ \gamma_{(et)}}\ (\mathbf{ev})}$$

An example of using the **ev** postulate for deriving inferences is shown in the following section.

5.4 Examples of $\mathcal{L}$-OC derivations

In this section we show examples of $\mathcal{L}$-OC derivations of inferences involving sentences with relative clauses and pied piping, as well as inferences using the extended 'every' postulate. Before each $\mathcal{L}$-OC proof, full $\mathcal{L}$ derivations of some of the sentences are shown. In fig. 5.2 we show a full $\mathcal{L}$ derivation of the sentence Every student whom Mary kissed smiled. In fig. 5.3 we see a $\mathcal{L}$-OC derivation of Every student whom Mary touched smiled $\vdash_{NatLog}$Every student whom Mary kissed smiled. In this derivation an instance of Ab (note that the Order Calculus of [4] does not have this inference rule) is used to discharge the assumption x_e.

In fig. 5.4 a full $\mathcal{L}$ derivation of the sentence Some boy the brother of whom Mary loves walked (involving pied piping) is given. In fig. 5.5 we show a $\mathcal{L}$-OC derivation of the inference Some boy, the brother of whom Mary loves, walked $\vdash_{NatLog}$Some boy walked. Note that since there does not exist an AB derivation for Mary loves, the system of [4] cannot derive inferences of this kind.

In fig. 5.6. we give a full $\mathcal{L}$ derivation of the sentence Mary kissed every student. In fig. 5.7 the extended 'every' postulate is used to derive the inference Mary kissed every student, No student whom Mary kissed walked $\vdash_{NatLog}$ No student walked. Recall that the original 'every' postulate defined by [4] could not deal with sentences where 'every' is in object position.

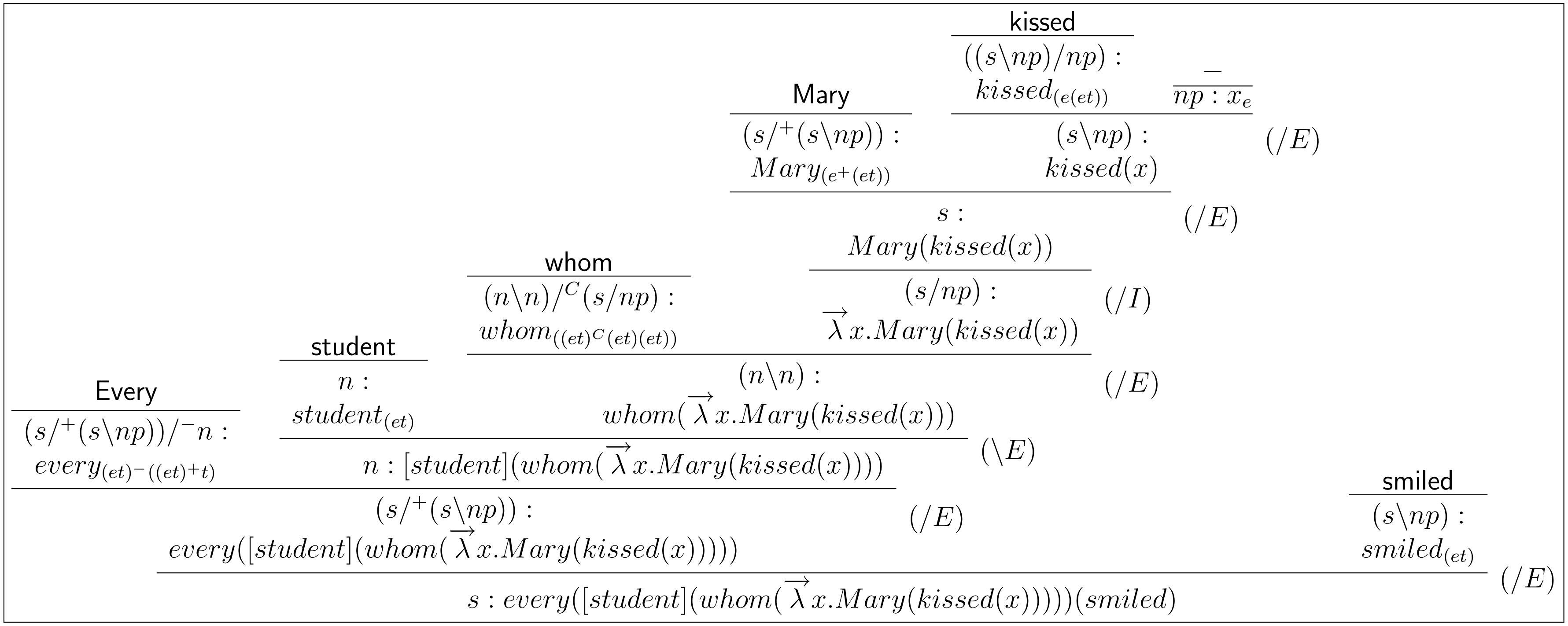


Figure 5.2: $\mathcal{L}$ derivation of Every student whom Mary kissed smiled

$$\dfrac{\emptyset}{kissed_{(e(et))} \;\leq\; touched_{(e(et))}}\; a_3$$

$$\dfrac{kissed_{(e(et))} \;\leq\; touched_{(e(et))}}{(kissed\ (x_e))_{\mathbf{(et)}} \;\leq\; (touched\ (x_e))_{\mathbf{(et)}}}\ \text{FR}$$

$$\dfrac{(kissed\ (x_e))_{\mathbf{(et)}} \;\leq\; (touched\ (x_e))_{\mathbf{(et)}}}{(Mary_{((et)^{+}t)}\ (kissed\ (x)))_{\mathbf{t}} \;\leq\; (Mary_{((et)^{+}t)}\ (touched\ (x)))_{\mathbf{t}}}\ \text{MON+}$$

$$\dfrac{(Mary_{((et)^{+}t)}\ (kissed\ (x)))_{\mathbf{t}} \;\leq\; (Mary_{((et)^{+}t)}\ (touched\ (x)))_{\mathbf{t}}}{(\overrightarrow{\lambda} x.Mary(kissed\ (x)))_{\mathbf{(et)}} \;\leq\; (\overrightarrow{\lambda} x.Mary\ (touched\ (x)))_{\mathbf{(et)}}}\ \text{Ab}$$

$$\dfrac{(\overrightarrow{\lambda} x.Mary(kissed\ (x)))_{\mathbf{(et)}} \;\leq\; (\overrightarrow{\lambda} x.Mary\ (touched\ (x)))_{\mathbf{(et)}}}{(whom_{(et)^{C}((et)(et))}\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x))))_{\mathbf{(et)(et)}} \leq (whom_{(et)^{C}((et)(et))}\ (\overrightarrow{\lambda} x.Mary(touched\ (x))))_{\mathbf{(et)(et)}}}\ \text{MON+}$$

$$\dfrac{(whom_{(et)^{C}((et)(et))}\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x))))_{\mathbf{(et)(et)}} \leq (whom_{(et)^{C}((et)(et))}\ (\overrightarrow{\lambda} x.Mary(touched\ (x))))_{\mathbf{(et)(et)}}}{([student_{(et)}]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x)))))_{\mathbf{(et)}} \leq ([student_{(et)}]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (touched\ (x)))))_{\mathbf{(et)}}}\ \text{FR}$$

$$\dfrac{([student_{(et)}]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x)))))_{\mathbf{(et)}} \leq ([student_{(et)}]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (touched\ (x)))))_{\mathbf{(et)}}}{(every_{(et^{-})((et)^{+}t)}\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (touched\ (x))))))_{(\mathbf{(et)}^{+}\mathbf{t})} \leq ((every_{(et^{-})((et)^{+}t)}\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x))))))_{(\mathbf{(et)}^{+}\mathbf{t})}}\ \text{MON-}$$

$$\dfrac{(every_{(et^{-})((et)^{+}t)}\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (touched\ (x))))))_{(\mathbf{(et)}^{+}\mathbf{t})} \leq ((every_{(et^{-})((et)^{+}t)}\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x))))))_{(\mathbf{(et)}^{+}\mathbf{t})}}{((every\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (touched\ (x))))))(smiled_{(et)}))_{\mathbf{t}} \leq ((every\ ([student]\ (whom\ (\overrightarrow{\lambda} x.Mary\ (kissed\ (x))))))\ (smiled_{(et)}))_{\mathbf{t}}}\ \text{FR}$$

Figure 5.3: Relative clauses:
Every student whom Mary touched smiled $\vdash_{NatLog}$Every student whom Mary kissed smiled

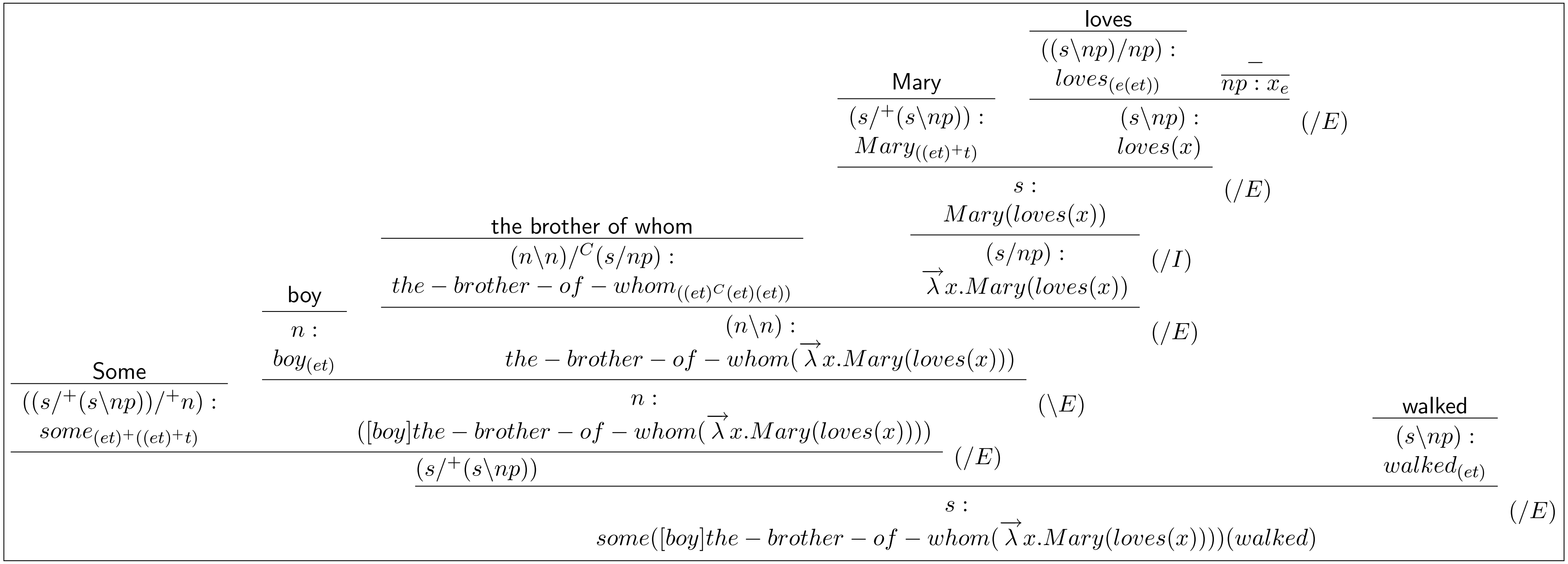


Figure 5.4: $\mathcal{L}$ derivation of Some boy the brother of whom Mary loves walked

$$\dfrac{\dfrac{\dfrac{\emptyset}{[boy_{(et)}](the-brother-of-whom_{((et)^C((et)(et)}(\overrightarrow{\lambda}x.Mary(loves(x))_{(et)})) \leq boy_{(et)}}\ \text{C1}}{some_{(et)^+((et)^+t)}([boy_{(et)}](the-brother-of-whom_{((et)^C((et)(et)}(\overrightarrow{\lambda}x.Mary(loves(x))_{(et)}))) \leq some_{(et)^+((et)^+t)}(boy)}\ \text{MON+}}{some_{(et)^+((et)^+t)}([boy_{(et)}](the-brother-of-whom_{((et)^C((et)(et)}((\overrightarrow{\lambda}x.Mary(loves(x)))_{(et)})))(walked_{(et)}) \leq (some_{(et)^+((et)^+t)}(boy))(walked_{et})}\ \text{FR}$$

Figure 5.5: Pied piping:
Some boy, the brother of whom Mary loves, walked $\vdash_{NatLog}$Some boy walked

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Mary}}{(s/^+(s\backslash np)) : Mary_{((et)^+t)}} \quad \dfrac{\dfrac{\text{kissed}}{((s\backslash np)/np) : kissed_{(e(et))}} \quad \dfrac{-}{np : x_e}}{(s\backslash np) : kissed(x)}\ (/E)}{s : Mary(kissed(x))}\ (/E)}{(s/np) : \overrightarrow{\lambda}x.Mary(kissed(x))}\ (/I) \quad \dfrac{\dfrac{\text{every}}{((s\backslash^+(s/np))/^-n) : every_{(et)^-((et)^+t)}} \quad \dfrac{\text{student}}{n : student_{(et)}}}{(s\backslash^+(s/np)) : every(student)}\ (/E)}{s : [\overrightarrow{\lambda}x.Mary(kissed(x))](every(student))}\ (\backslash E)$$

Figure 5.6: $\mathcal{L}$ derivation of Mary kissed every student

$$\dfrac{\dfrac{\dfrac{\dfrac{student_{(et)} \ \leq \ student_{(et)}, \quad w_t^T \leq ([\overrightarrow{\lambda} x_e.Mary_{((et)^+t)} \ (kissed_{(e(et))} \ (x_e))] \ (every_{(et)^-((et)^+t)} \ (student_{(et)})))_{\mathbf{t}}}{student_{(et)} \ \leq \ (\overrightarrow{\lambda} x.Mary \ (kissed \ (x)))_{\mathbf{(et)}}}\ \mathbf{ev}}{student_{(et)} \ \leq \ ([student_{(et)}] \ whom_{(et)^C(et)(et)} \ (\overrightarrow{\lambda} x.Mary \ (kissed \ (x))))_{\mathbf{(et)}}}\ \text{C2}}{(No_{(et)^-((et)^-t)} \ ([student] \ (whom \ (\overrightarrow{\lambda} x.Mary \ (kissed \ (x))))))_{(\mathbf{(et)}^-\mathbf{t})} \leq (No_{(et)^-((et)^-t)} \ (student))_{(\mathbf{(et)}^-\mathbf{t})}, \quad walked_{(et)} \ \equiv \ walked_{(et)}}\ \text{MON-}}{w_t^T \ \leq \ ((No \ ([student] \ (whom \ (\overrightarrow{\lambda} x.Mary \ (kissed \ (x)))))) \ (walked_{(et)}))_{\mathbf{t}} \ \leq ((No \ (student))(walked_{(et)}))_{\mathbf{t}}}\ \text{FR}$$

$$\dfrac{w_t^T \ \leq \ ((No \ ([student] \ (whom \ (\overrightarrow{\lambda} x.Mary \ (kissed \ (x)))))) \ (walked_{(et)}))_{\mathbf{t}} \ \leq ((No \ (student))(walked_{(et)}))_{\mathbf{t}}}{w_t^T \ \leq \ ((No \ (student))(walked))_{\mathbf{t}}}\ \text{TRANS} \times 2$$

Figure 5.7: Using the extended non-logical postulate for 'every':Mary kissed every student, No student whom Mary kissed walked $\vdash_{NatLog}$ No student walked

Chapter 6

Exploring properties of $\mathcal{L}$-OC

6.1 Normalization in $\mathcal{L}$-OC

In this section we focus on normalization[1] in $\mathcal{L}$-OC . In the first subsection we present examples of $\mathcal{L}$-OC derivations of inferences using normalization. In the second subsection we investigate the problematic aspects of non-normalized proof terms and show how normalization deals with them. In the third subsection we propose an alternative for normalization – the method of *dynamic marking* and discuss the relation between these methods.

6.1.1 Examples of using the normalization axioms

In fig. 6.1 and 6.2 we show examples of inferences that could not be proven in $\mathcal{L}$-OC without the normalization axioms. Instances of the Reflexivity rule are omitted.

In fig. 6.1 η-normalization is used to prove the order statement

$[adores]and(\overrightarrow{\lambda}x.passionately(loves(x)) \leq loves.$

In fig. 6.2 β-normalization is used to normalize the non-NF proof term

$\overrightarrow{\lambda}x.creative(intelligent(x))(boy)$ (for the goal order statement).

[1]Normalization in $\mathcal{L}$-OC was initially proposed in [13].

$$\dfrac{\dfrac{\emptyset}{([adores]and(\overrightarrow{\lambda} x.passionately(loves(x)))) \leq \overrightarrow{\lambda} x.passionately(loves(x))}\ \textsc{c1} \qquad \dfrac{\dfrac{\emptyset}{passionately_{(et)^R(et)}(loves(x)) \leq loves(x)}\ \textsc{rmod}}{\overrightarrow{\lambda} x.passionately(loves(x)) \leq \overrightarrow{\lambda} x.loves(x)}\ \text{Ab} \qquad \dfrac{\emptyset}{\overrightarrow{\lambda} x.loves(x) \equiv loves}\ \eta}{([adores]and(\overrightarrow{\lambda} x.passionately(loves(x)))) \leq loves}\ \textsc{trans}$$

$$\dfrac{([adores]and(\overrightarrow{\lambda} x.passionately(loves(x)))) \leq loves}{([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y)) \leq loves(y)}\ \textsc{fr}$$

$$\dfrac{([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y)) \leq loves(y)}{Mary([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y)) \leq Mary(loves(y))}\ \textsc{mon}+$$

$$\dfrac{Mary([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y)) \leq Mary(loves(y))}{\overrightarrow{\lambda} y.(Mary([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y))) \leq \overrightarrow{\lambda} y.Mary(loves(y))}\ \text{Ab}$$

$$\dfrac{\overrightarrow{\lambda} y.(Mary([adores]and(\overrightarrow{\lambda} x.passionately(loves(x))))(y))) \leq \overrightarrow{\lambda} y.Mary(loves(y))}{[\overrightarrow{\lambda} y.([Mary([adores]and(\overrightarrow{\lambda} x.passionately(loves(x)))(y))])]John \leq [\overrightarrow{\lambda} y.Mary(loves(y))]John_{(et^+t)}}\ \textsc{mon}+$$

Figure 6.1: Mary adores and passionately loves John $\vdash_{NatLog}$ Mary loves John

$$\dfrac{\dfrac{\dfrac{\dfrac{\emptyset}{\begin{array}{c} creative(intelligent(boy)) \equiv \\ \overrightarrow{\lambda} x.creative(intelligent(x))(boy) \end{array}}\,\beta \qquad \dfrac{\dfrac{\emptyset}{\overrightarrow{\lambda} x_{(et)}.creative_{((et)^R(et))}(intelligent_{((et)^R(et))}(x_{(et)})) \leq smart_{(et)}}\,a_5}{\overrightarrow{\lambda} x.creative(intelligent(x))(boy_{(et)}) \leq smart(boy_{(et)})}\,\textsc{fr}}{creative(intelligent(boy)) \leq smart(boy)}\,\textsc{trans}}{Some_{((et)^+((et)^+t))}(creative(intelligent(boy))) \leq Some_{((et)^+((et)^+t))}(smart(boy))}\,\textsc{mon}}{Some(creative(intelligent(boy)))(smiled) \leq Some(smart(boy))(smiled)}\,\textsc{fr}$$

Figure 6.2: Some creative intelligent boy smiled $\vdash_{NatLog}$ Some smart boy smiled, using the non-logical axiom ($a5$).

6.1.2 Why is normalization needed?

In this subsection we would like to show why the emergence of proof terms which are not in normal form (NF) in $\mathcal{L}$-OC poses a problem. First of all, let us demonstrate how non-NF proof terms emerge in $\mathcal{L}$-OC . Consider the following examples.

$$\dfrac{\dfrac{\emptyset}{[\overrightarrow{\lambda} x_{(et)}.John_{((et)^+t)}(does_{((et)^+(et))}(x_{(et)}))]and_{((et)t)^C(((et)t)((et)t))}(\overrightarrow{\lambda} y_{(et)}.Mary(doesn't_{((et)^-(et))}(y))) \leq \overrightarrow{\lambda} y.Mary(doesn't(y_{(et)}))}\text{C1}}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(walk_{(et)}) \leq \overrightarrow{\lambda} y.Mary(doesn't(y))(walk_{(et)})}\text{FR}$$

The term $\overrightarrow{\lambda} y.Mary(doesn't(y))(walk)$ is not in NF and it β-reduces to $Mary(doesn't(walk))$.

$$\dfrac{\dfrac{\emptyset}{\overrightarrow{\lambda} x_{(et)}.creative_{((et)^R(et))}(intelligent_{((et)^R(et))}(x_{(et)})) \leq smart_{((et)^{(}et))}}a_5}{\overrightarrow{\lambda} x.creative(intelligent(x))(boy_{(et)}) \leq smart(boy_{(et)})}\text{FR}$$

The term $\overrightarrow{\lambda} x.creative(intelligent(x))(boy)$ is not in NF and it β-reduces to $creative(intelligent(boy))$.

$$\dfrac{\dfrac{\emptyset}{happy_{(et)^R(et)}(tall_{(et)^R(et)}(x_{(et)})) \leq tall_{(et)^R(et)}(x_{(et)})}\text{RMOD}}{\overrightarrow{\lambda} x.happy_{(et)^R(et)}(tall_{(et)^R(et)}(x_{(et)})) \leq \overrightarrow{\lambda} x.tall_{(et)^R(et)}(x_{(et)})}\text{Ab}$$

The term $\overrightarrow{\lambda} x.tall(x)$ is not in NF and it η-reduces to $tall$.

We see two main problematic aspects in the emergence of non-normalized proof terms in $\mathcal{L}$-OC . The first problem is *abstraction terms* with unmarked semantic types. Basing the system on $\mathcal{L}$ allows us to derive order statements that involve complex functional terms that do not originate from the lexicon, e.g. composition of terms. In AB, in contrast to $\mathcal{L}$, the creation of functional terms (that do not originate from the lexicon) is impossible due to the lack of introduction rules. In $\mathcal{L}$-OC new functional terms, that are created via abstraction during parsing, can be applied as functions to other terms, creating non-NF terms. Some of the abstraction terms may denote monotone (restrictive, etc.) functions, but their types are not respectively marked. However, it is desirable to derive inferences based on the semantic properties of the denotations of the abstraction terms. For instance, consider the abstraction term $\mu = \overrightarrow{\lambda} x_\tau.\psi_{(\sigma^+\rho)}(\phi_{(\tau^+\sigma)}(x_\tau))$, which is a composition of the terms ϕ and ψ. Since their types are marked for upward monotonicity, the denotation of their composition also is an upward monotone function. Thus, given $\vdash_{\mathcal{L}-OC} \gamma_\tau \leq_\tau \delta_\tau$, we expect $\mathcal{L}$-OC to derive $(\overrightarrow{\lambda} x.\psi(\phi(x)))(\gamma) \leq_\rho (\overrightarrow{\lambda} x.\psi(\phi(x)))(\delta)$. But the type of $\overrightarrow{\lambda} x.\psi(\phi(x))$ is not marked for monotonicity, thus we cannot use the MON+ rule (or any other $\mathcal{L}$-OC rules) directly.

A more concrete example is the following valid inference: John does and Mary doesn't move $\vdash_{NatLog}$ Mary doesn't walk, using the non-logical axiom (a_2) $walk \leq move$. Note that the type of $doesn't_{((et)^-(et))}$ is marked for downward monotonicity. Also, by using C1 and FR:

$$\vdash_{\mathcal{L}-OC} [\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move) \leq$$

$$\overrightarrow{\lambda} y.Mary(doesn't(y))(move)$$

However, since the type of $\overrightarrow{\lambda} y.Mary(doesn't(y))$ is not marked for downward monotonicity, without normalizing we cannot use the non-logical axiom a_2 in any way. On the other hand, by using MON and FR: $\vdash_{\mathcal{L}-OC} Mary(doesn't(move)) \leq Mary(doesn't(walk))$. Thus we conclude that establishing a connection between two $\beta\eta$-equivalent terms is needed in $\mathcal{L}$-OC [2].

Effectiveness considerations are another problematic aspect of non-NF terms in $\mathcal{L}$-OC . Recall the general structure of our system (summarized in fig. 5.1). One of its integral parts is finding $\mathcal{L}$-derivations for the goal sentences. However, finding a non-normalized derivation of some NL expression is problematic due to the lack of the sub-formula property in non-normalized derivations, which in its turn creates an infinite proof search space. Therefore, any realistic $\mathcal{L}$ parser searches for normal form derivations only. Thus for the purpose of implementation, we are led to the need to express the relation between non-NF terms representing $\mathcal{L}$ derivations of the goal sentences and their normal form equivalents. Before demonstrating the effect of normalization, we would like to propose $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ – a simplified reformulation of $\mathcal{L}$-OC , in which normalization is applied to the conclusions of FR and Ab rules only. The advantage of such a reformulation is simpler proofs: we will show that a $\mathcal{L}$-OC proof using normalization axioms (and thus necessarily containing non-NF proof terms) can be rewritten in $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ using NF proof terms only. We will

[2] In subsection 6.1.3 an alternative solution of the mentioned problem, using Dynamic marking of abstraction terms is presented.

also discuss the equivalence between $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ and $\mathcal{L}$-OC .

$\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ is similar to $\mathcal{L}$-OC , except that (β) and (η) are not explicitly among its axioms. Instead additional inference rules – $\hat{\mathrm{FR}}$ and $\hat{\mathrm{Ab}}$ implicitly encapsulate $\beta-$ and $\eta-$normalization:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\psi \leq \phi}\ R} \qquad \boxed{\dfrac{\bigtriangledown}{\delta \equiv \delta'}}}{\overline{norm}(\psi(\delta)) \leq \overline{norm}(\phi(\delta'))}\ \hat{\mathrm{FR}} \qquad \dfrac{\boxed{\dfrac{\bigtriangledown}{\gamma^x \leq \delta^x}}}{\overline{ab}(\gamma^x) \leq \overline{ab}(\delta^x)}\ \hat{\mathrm{Ab}}$$

$$\psi\ or\ \phi\ \ is\ an\ abstraction\ term,\ \gamma\ or\ \delta\ \ is\ of\ form\ \gamma\langle x\rangle.$$

$$\overline{norm}(\varphi) = \{ \begin{array}{l} \delta^x[x/\gamma]\ for\ \varphi = \overline{\lambda} x.\delta^x\langle\gamma\rangle \\ \varphi\ Otherwise \end{array} \qquad \overline{ab}(\varphi^x) = \{ \begin{array}{l} \delta\ for\ \varphi^x = \delta\langle x\rangle,\ x \in \mathbf{VAR} \\ \overline{\lambda} x.\varphi^x\ Otherwise \end{array}$$

In order to recover the terms ψ, δ from $\overline{norm}(\psi(\delta))$, we define *rightmost (leftmost) terms* and *anti-substitution*.

Definition 6.1.1 (RM (rightmost) subterms) *Let ψ be a term and $\alpha \in$ **LexVAR** $\cup$ **VAR** s.t. α is right-peripheral in $\overline{Free}(\psi)$. The set $RM(\psi) = \{\varphi \mid \varphi \in ST(\psi),\ \alpha \in ST(\varphi)\}$.*

The leftmost subterms are defined symmetrically.

Definition 6.1.2 (Anti-substitution) *For terms ψ_τ, φ_ρ s.t (i) $\varphi_\rho \in ST(\psi_\tau)$ and (ii) no variable $z \in Free(\varphi_\rho) \cap$ **VAR** is bound in ψ_τ, $(\psi_\tau \ll x_\rho/\varphi_\rho \gg)$ is the term obtained from ψ_τ by replacing (an occurrence of) its subterm φ_ρ by some variable $x_\rho \in$ **VAR** s.t. $x_\rho \notin Free(\psi_\tau)$.*

Note that for any[3] $\delta \in RM(\psi)$, $\overline{norm}((\overrightarrow{\overline{\lambda}} x.\psi \ll x/\delta \gg)(\delta)) = \psi$.

The following statements proven below show in what sense $\mathcal{L}$-OC and $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ are equivalent: (i) $\vdash_{\mathcal{L}-OC} \alpha' \leq \gamma' \ \Rightarrow \vdash_{\mathcal{L}-OC^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}} \alpha \leq \gamma$ for α, γ the NF of α', γ' resp., (ii) $\vdash_{\mathcal{L}-OC^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}} \alpha \leq \gamma \ \Rightarrow \vdash_{\mathcal{L}-OC} \alpha \leq \gamma$.

[3]There can be more than one such δ.

Lemma 6.1.1 *Let the order statement $\psi' \leq \phi'$ have a $\mathcal{L}$-OC proof Ω. Then there exists a $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof Ω' of $\psi \leq \phi$ s.t. ψ, ϕ are the NF of ψ', ϕ' resp. and Ω' contains NF terms only.*

Proof: By induction on the structure of Ω. Let $\mathbf{coor}^{C/D} = \delta_{(\tau^{C/D}(\tau\tau))}$ for some term δ. If Ω is one of the axioms:

- $\Omega = \boxed{\dfrac{\emptyset}{\psi' = \alpha' \leq \alpha' = \phi'}\ \text{REFL}}$. Then $\Omega' = \boxed{\dfrac{\emptyset}{\alpha \leq \alpha}\ \text{REFL}}$ for α NF of α'.
- $\Omega = \boxed{\dfrac{\emptyset}{\alpha_{(\tau^{R}\tau)}\langle\mu'\rangle \leq \mu'}\ \text{RMOD}}$. Since the type of α is marked for restrictivity, $\alpha \in$ **LexVAR**. Thus $\Omega' = \boxed{\dfrac{\emptyset}{\alpha_{(\tau^{R}\tau)}\langle\mu\rangle \leq \mu}\ \text{RMOD}}$ for μ NF of μ'.
- $\boxed{\dfrac{\emptyset}{[\alpha']\ \mathbf{coor}^{C/D}\ (\gamma') \leq \alpha'/\gamma'}\ \text{C1}}$. Then $\Omega' = \boxed{\dfrac{\emptyset}{[\alpha]\ \mathbf{coor}^{C/D}\ (\gamma) \leq \alpha/\gamma}\ \text{C1}}$ for α, γ NF of α', γ' resp.
- Similarly for D1.
- $\boxed{\dfrac{\emptyset}{\overline{\lambda x}.\psi'^{x}\langle\gamma'\rangle \equiv \psi'^{x}[x/\gamma']}\ \beta}$ Then $\Omega' = \boxed{\dfrac{\emptyset}{\psi^{x}[x/\gamma] \equiv \psi^{x}[x/\gamma]}\ \text{REFL}}$ for ψ^{x}, γ NF of ψ'^{x}, γ' resp.
- Similarly for η.

If Ω is of form:

$$\boxed{\dfrac{\Psi'_1\ \boxed{\dfrac{\nabla}{\psi_1 \leq \psi'_1}\ R_1}\quad \Psi'_2\ \boxed{\dfrac{\nabla}{\psi_2 \leq \psi'_2}\ R_2}\quad \ldots\quad \Psi'_n\ \boxed{\dfrac{\nabla}{\psi_n \leq \psi'_n}\ R_n}}{\psi_1 \leq \psi'_n}\ \text{R}}$$

then R is one of the following rules:

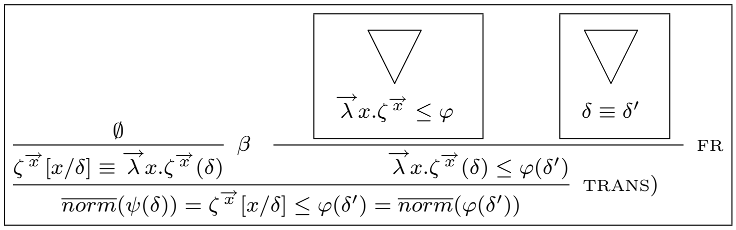
- R =TRANS. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proofs Ψ_i of $\phi_i \leq \phi'_i$ for $1 \leq i \leq n$, (where ϕ_i, ϕ'_i are NF of ψ_i, ψ'_i resp.), which contain NF terms only. Thus $\Omega' = \boxed{\dfrac{\Psi_1\ \ldots\ \Psi_n}{\phi_1 \leq \phi_n}\ \text{TRANS}}$.
- $R =$ C2. Then Ω is of form $\boxed{\dfrac{\Psi'_1\ \boxed{\dfrac{\nabla}{\psi' \leq \gamma'}}\quad \Psi'_2\ \boxed{\dfrac{\nabla}{\psi' \leq \delta'}}}{\psi' \leq ([\gamma']\ \mathbf{coor}^{C}\ )(\delta')}\ \text{C2}}$. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proofs Ψ_1, Ψ_2 of the order statements $\psi \leq \gamma$ and $\psi \leq \delta$ resp., s.t. ψ, γ, δ are NF of ψ', γ', δ' resp. and Ψ_1, Ψ_2 contain NF terms only. Then $\Omega' = \boxed{\dfrac{\Psi_1\ \boxed{\dfrac{\nabla}{\psi \leq \gamma}}\quad \Psi_2\ \boxed{\dfrac{\nabla}{\psi \leq \delta}}}{\alpha \leq ([\gamma]\ \mathbf{coor}^{C}\ )(\delta)}\ \text{C2}}$

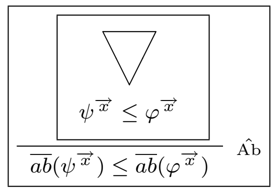
- $R =$ D2/MON. The proof is similar to the previous case.
- $R =$Ab. Then Ω is of form $\dfrac{\overset{\Psi'}{\dfrac{\nabla}{\mu'^x \leq \varphi'^x}}}{\overline{\lambda}x.\mu'^x \leq \overline{\lambda}x.\varphi'^x}\,\text{Ab}$. By the induction hypothesis, there exists a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof Ψ of $\mu^x \leq \varphi^x$ s.t. μ^x, φ^x are NF of μ'^x, φ'^x resp. which contains NF terms only. If none of the terms μ^x, φ^x is of form $\zeta\langle x\rangle$, then $\Omega' = \dfrac{\overset{\Psi}{\dfrac{\nabla}{\mu^x \leq \varphi^x}}}{\overline{\lambda}x.\mu^x \leq \overline{\lambda}x.\varphi^x}\,\text{Ab}$. Otherwise $\Omega' = \dfrac{\overset{\Psi}{\dfrac{\nabla}{\mu^x \leq \varphi^x}}}{ab(\mu^x) \leq ab(\varphi^x)}\,\hat{\text{Ab}}$.
- $R =$ FR. Then Ω is of form $\dfrac{\overset{\Psi'_1}{\dfrac{\nabla}{\mu' \leq \varphi'}} \quad \overset{\Psi'_2}{\dfrac{\nabla}{\delta' \equiv \gamma'}}}{\mu'\langle\delta'\rangle \leq \varphi'\langle\gamma'\rangle}\,\text{FR}$. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proofs Ψ_1, Ψ_2 of $\mu \leq \varphi$ and $\delta \equiv \gamma$ s.t. $\mu, \varphi, \delta, \gamma$ are NF of $\mu', \varphi', \delta', \gamma'$ resp. which contains NF terms only. If none of the terms μ, φ is an abstraction term, then $\Omega' = \dfrac{\overset{\Psi_1}{\dfrac{\nabla}{\mu \leq \varphi}} \quad \overset{\Psi_2}{\dfrac{\nabla}{\delta \equiv \gamma}}}{\mu\langle\delta\rangle \leq \varphi\langle\gamma\rangle}\,\text{FR}$. Otherwise Ω' $= \dfrac{\overset{\Psi_1}{\dfrac{\nabla}{\mu \leq \varphi}} \quad \overset{\Psi_2}{\dfrac{\nabla}{\delta \equiv \gamma}}}{\overline{norm}(\mu\langle\delta\rangle) \leq \overline{norm}(\varphi\langle\gamma\rangle)}\,\hat{\text{FR}}$.

Lemma 6.1.2 *Let the order statement $\alpha \leq \gamma$ have a $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof Ω. Then there exists a $\mathcal{L}$-OC proof Ω' of $\alpha \leq \gamma$.*

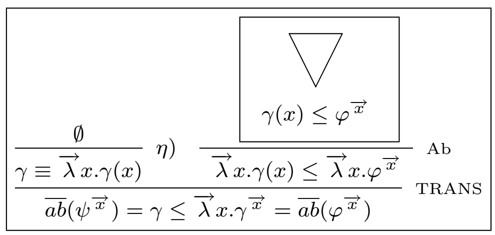
Proof sketch: We show the construction of a $\mathcal{L}$-OC proof Ω' of $\alpha \leq \gamma$ from Ω by elimination of $\hat{\text{FR}}$ and $\hat{\text{Ab}}$. If Ω does not contain any instances of $\hat{\text{FR}}$ and $\hat{\text{Ab}}$ then it is already a $\mathcal{L}$-OC proof of $\alpha \leq \gamma$. Otherwise suppose Ω contains an instance of $\hat{\text{FR}}$ $\dfrac{\dfrac{\nabla}{\psi \leq \varphi} \quad \dfrac{\nabla}{\delta \equiv \delta'}}{\overline{norm}(\psi(\delta)) \leq \overline{norm}(\varphi(\delta'))}\,\hat{\text{FR}}$ where w.l.o.g. $\psi = \overrightarrow{\lambda} x.\zeta^{\overrightarrow{x}}$, $\overline{norm}(\psi(\delta)) = \zeta^{\overrightarrow{x}}[x/\delta]$ and

$\overline{norm}(\varphi(\delta')) = \varphi(\delta')$. Then we can replace it by the following $\mathcal{L}$-OC proof:



Now suppose Ω contains an instance of $\hat{\text{Ab}}$: 

where w.l.o.g. $\psi^{\overrightarrow{x}} = \gamma(x)$. Then we can replace it by the following $\mathcal{L}$-OC proof:



In this way we can eliminate all instances of $\hat{\text{FR}}$ and $\hat{\text{Ab}}$ in Ω to obtain a valid $\mathcal{L}$-OC proof Ω'.

To sum up, the relation between $\mathcal{L}$-OC and $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ is the following:

- $\mathcal{L}$-OC and $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ are equivalent w.r.t. order statements with NF proof terms. That is, for all NF proof terms ψ and φ, $\vdash_{\mathcal{L}-OC} \psi \leq \varphi$ iff $\vdash_{\mathcal{L}-OC^{\hat{\text{FR}},\hat{\text{Ab}}}} \psi \leq \varphi$.
- Proof-theoretically, $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ is weaker than $\mathcal{L}$-OC . For example, for any proof term $\psi_{(\tau\sigma)}$:

$$\vdash_{\mathcal{L}-OC} \psi \equiv \lambda x_\tau.\psi_{(\tau\sigma)}\langle x_\tau \rangle$$

$$(*)\ \nvdash_{\mathcal{L}-OC^{\hat{\text{FR}},\hat{\text{Ab}}}} \psi \equiv \lambda x_\tau.\psi_{(\tau\sigma)}\langle x_\tau \rangle$$

In practice, however, we do not lose any semantic information by weakening $\mathcal{L}$-OC , since for any order statement $\alpha' \leq \gamma'$ provable in $\mathcal{L}$-OC , we can prove a semantically equivalent order statement $\alpha \leq \gamma$ in $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ (where α, γ are NF of α', γ' resp.)

Thus we will use the simpler $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ instead of $\mathcal{L}$-OC both for demonstrating the effect of normalization and for defining a proof search procedure in the following chapter.

Now let us demonstrate how normalization in $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ (encapsulated in $\hat{\text{FR}}$ and $\hat{\text{Ab}}$)

solves the above mentioned problems of non-NF terms. Consider again the problematic inference we mentioned above – we show in fig. 6.3 that it is derivable in $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$. As for the *effectiveness* consideration, the goal we specified above is also achieved by normalization. Recall that we aimed at being able to use NF terms to represent $\mathcal{L}$ derivations of the goal sentences. We have shown that (i) for any order statement s.t. $\vdash_{\mathcal{L}-OC} \alpha' \leq \gamma'$, a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of the (semantically equivalent) order statement $\alpha \leq \gamma$ s.t. α, γ are NF of α', γ' resp. can be constructed, and (ii) for each such $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof, an equivalent $\mathcal{L}$-OC proof exists. Thus for any α', γ' so that $\vdash_{\mathcal{L}-OC} \alpha' \leq \gamma'$ there exists a $\mathcal{L}$-OC proof of $\alpha \leq \gamma$, s.t. α, γ are NF of α', γ' resp.

6.1.3 Dynamic marking: an alternative?

An alternative remedy to the problem of unmarked abstraction terms mentioned in the previous section would naturally be marking their types. We propose a method of marking the types of abstraction terms for monotonicity (it can also be extended to other semantic features as well), henceforth referred to as *dynamic marking* [4]. To implement dynamic marking we use the notion of *polarity* introduced by [11] and used by [9] and [1].

Definition 6.1.3 (Polarity of occurrences) *Given a term ψ and a subterm ϕ of ψ, a specified occurrence of ϕ in ψ is called* positive *(*negative*) according to the following clauses:*

1. *ϕ is positive in ϕ.*
2. *If $\psi = \alpha\langle\gamma\rangle$ then:*
   - *ϕ is positive (negative) in ψ if ϕ is positive (negative) in α.*
   - *ϕ is positive (negative) in ψ if ϕ is positive (negative) in γ and α denotes an upward monotone function.*
   - *ϕ is negative (positive) in ψ if ϕ is positive (negative) in γ and α denotes a downward monotone function.*
3. *If $\psi = \overline{\lambda} x.\mu$ then ϕ is positive (negative) in ψ if ϕ is positive (negative) in μ.*

[4]The method of dynamic marking was initially proposed in [6]

$$
\dfrac{
\boxed{\dfrac{\dfrac{\emptyset}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y))) \atop \leq\ \overrightarrow{\lambda} y.Mary(doesn't(y))}\ \textsc{c1} \qquad \dfrac{\emptyset}{move \equiv move}\ \textsc{refl}}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move) \atop \leq Mary(doesn't(move))}\ \hat{\textsc{fr}}}
\quad
\boxed{\dfrac{\dfrac{\dfrac{\emptyset}{walk_{(et)} \leq move_{(et)}}\ (a_2)}{doesn't_{(et)^-(et)}(move) \atop \leq doesn't(walk)}\ \textsc{mon-}}{Mary_{((et)^+t)}(doesn't(move)) \atop \leq Mary_{((et)^+t)}(doesn't(walk))}\ \textsc{mon+}}
}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move) \leq Mary(doesn't(walk))}\ \textsc{trans}
$$

Figure 6.3: The problematic inference is derivable in $\mathcal{L}$-$\mathrm{OC}^{\hat{\mathrm{FR}},\hat{\mathrm{A}}\mathrm{b}}$: John does and Mary doesn't move $\vdash_{NatLog}$ Mary doesn't walk

Fact 6.1.1 *([11]) If x is positive (negative) in ϕ then $\overline{\lambda}x.\phi$ denotes an upward (downward) monotone function.*

The dynamic marking is performed by $\mathcal{DDL}$ – an extended version of $\mathcal{L}$. Instead of PL lambda terms, $\mathcal{DDL}$ uses *extended* PL (EPL) terms, where each free variable occurrence is assigned a polarity marking $\Pi \in Pol = \{\oplus, \ominus, null\}$, which is an *abstraction* of its actual polarity: $\oplus, \ominus, null$ mark positive, negative and undefined polarity resp.

Definition 6.1.4 (Extended PL terms) *Let Π be a meta variable that ranges over the values from $Pol = \{\oplus, \ominus, null\}$. Let $\boldsymbol{VAR}^{\Pi} = \{\Pi \in Pol\}$ and $\boldsymbol{LexVAR}^{\Pi} = \{w^{\Pi}_{\tau} \mid w_{\tau} \in \boldsymbol{LexVAR},\ \Pi \in Pol\}$. Let Ψ, Φ be meta variables that range over members of the set **EPLTerms** and x a meta variable that ranges over members of the set $\boldsymbol{VAR}^{\Pi}$. The set **EPLTerms** is the smallest set s.t.:*

- $\boldsymbol{LexVAR}^{\Pi}$, $\boldsymbol{VAR}^{\Pi} \subseteq$ ***EPLTerms***
- *If* $\Phi_{(\sigma^{*}\tau)}, \Psi_{\sigma} \in$ ***EPLTerms***, *then* $(\Phi(\Psi))_{\tau}, ([\Psi]\Phi)_{\tau} \in$ ***EPLTerms***
- *If* $x_{\rho} \in \boldsymbol{VAR}$, $\Phi_{\tau} \in$ ***EPLTerms***, *then* $(\overrightarrow{\lambda}x.\Phi)_{(\rho\tau)}, (\overleftarrow{\lambda}x.\Phi)_{(\rho\tau)} \in$ ***EPLTerms***

The set of free variables (with an assigned polarity marking) $Free(\Psi)$ for $\Psi \in$ **EPLTerms** is defined standardly.

Definition 6.1.5 (Π-strip) *For $\Psi \in$ **EPLTerms**, its Π-strip PL term $\Pi strip(\Psi)$ is defined as follows:*

$$\Pi strip(\alpha^{\Pi}) = \alpha,\ \ for\ \alpha \in \boldsymbol{LexVAR} \cup \boldsymbol{VAR}$$

$$\Pi strip(\Phi\langle\Delta\rangle) = \Pi strip(\Phi)\langle \Pi strip(\Delta)\rangle$$

$$\Pi strip(\overline{\lambda}x.\Phi) = \overline{\lambda}x.\Pi strip(\Phi)$$

In words, $\Pi strip(\Psi)$ is the term obtained from Ψ by deleting the polarity marking of all of its free variables.

Next, we define the functions $Flip$: **EPLTerms** → **EPLTerms** and $Anull$: **EPLTerms** → **EPLTerms**. $Flip(\Psi)$ is the (EPL) term obtained from Ψ by swapping the polarity marking of the free variables in Ψ as follows: '$\ominus$' to '$\oplus$', '$\oplus$' to '$\ominus$', 'null' to 'null'. $Anull(\Psi)$ is the (EPL) term obtained from Ψ by setting the polarity marking of all the free variables in Ψ to 'null'.

We also define the function $Pol2Feat : Pol \rightarrow 2^{Feat}$ that decorates the type of abstraction terms according to the polarity marking of the discharged assumption:

$$Pol2Feat(\oplus) = \{+\},\ Pol2Feat(\ominus) = \{-\},\ Pol2Feat(null) = \emptyset$$

Definition 6.1.6 **($\mathcal{DDL}$)** *Let $\Gamma, \Gamma_1, \Gamma_2$ range over finite non-empty sequences of pairs $A : \Psi_\tau$, where A_i is a syntactic category and $\Psi \in EPLTerms$. Let τ, τ_1, τ_2 range over decorated types. The notation $\Gamma \rhd A : \Psi_\tau$ means that the sequence Γ is $\mathcal{DDL}$-reducible to $A : \Psi_\tau$. The rules of $\mathcal{DDL}$ are as follows :*

$$(axiom_1) A : x_\tau^\oplus \rhd A : x_\tau^\oplus \quad for\ x \in \boldsymbol{VAR}^\Pi$$

$$(axiom_2) B : w_\tau^{null} \rhd B : w_\tau^{null} \quad for\ w \in \boldsymbol{LexVAR}^\Pi$$

Elimination rules:

$$for\ \tau_1 \equiv_f \tau_1' :$$

$$(/E_\ominus)\frac{\Gamma_1 \rhd (A/B) : \Psi_{(\tau_1 - \tau_2)} \quad \Gamma_2 \rhd B : \Phi_{\tau_1'}}{\Gamma_1\Gamma_2 \rhd A : (\Psi_{(\tau_1 - \tau_2)}(Flip(\Phi_{\tau_1'})))_{\tau_2}}, \quad (\backslash E_\ominus)\frac{\Gamma_2 \rhd B : \varphi_{\tau_1'} \quad \Gamma_1 \rhd (A\backslash B) : \Psi_{(\tau_1 - \tau_2)}}{\Gamma_2\Gamma_1 \rhd A : ([Flip(\Phi_{\tau_1'})]\Psi_{(\tau_1 - \tau_2)})_{\tau_2}}$$

$$(/E_\oplus)\frac{\Gamma_1 \rhd (A/B) : \Psi_{(\tau_1 + \tau_2)} \quad \Gamma_2 \rhd B : \Phi_{\tau_1'}}{\Gamma_1\Gamma_2 \rhd A : (\Psi_{(\tau_1 + \tau_2)}(\Phi_{\tau_1'}))_{\tau_2}}, \quad (\backslash E_\oplus)\frac{\Gamma_2 \rhd B : \Phi_{\tau_1'} \quad \Gamma_1 \rhd (A\backslash B) : \Psi_{(\tau_1 + \tau_2)}}{\Gamma_2\Gamma_1 \rhd A : ([\Phi_{\tau_1'}]\Psi_{(\tau_1 + \tau_2)})_{\tau_2}}$$

$$(/E)\frac{\Gamma_1 \rhd (A/B) : \Psi_{(\tau_1{}^F \tau_2)} \quad \Gamma_2 \rhd B : \Phi_{\tau_1'}}{\Gamma_1\Gamma_2 \rhd A : (\Psi_{(\tau_1{}^F \tau_2)}(Anull(\Phi_{\tau_1'})))_{\tau_2}}, \quad (\backslash E)\frac{\Gamma_2 \rhd B : \Phi_{\tau_1'} \quad \Gamma_1 \rhd (A\backslash B) : \Psi_{(\tau_1{}^F \tau_2)}}{\Gamma_2\Gamma_1 \rhd A : ([Anull(\Phi_{\tau_1'})]\Psi_{(\tau_1 - \tau_2)})_{\tau_2}}$$

$$For F \neq \{+\}\ and\ F \neq \{-\} :$$

Introduction rules:

$$(/I)\frac{\Gamma_1,\ B : x_{\tau_1}^\Pi \rhd A : \Psi_{\tau_2}}{\Gamma_1 \rhd (A/B) : (\overrightarrow{\lambda} x_{\tau_1}.\Psi_{\tau_2})_{(\tau_1^{Pol2Feat(\Pi)}\tau_2)}} \quad (\backslash I)\frac{B : x_{\tau_1}^\Pi,\ \Gamma_1 \rhd A : \Psi_{\tau_2}}{\Gamma_1 \rhd (A\backslash B) : (\overleftarrow{\lambda} x_{\tau_1}.\Psi_{\tau_2})_{(\tau_1^{Pol2Feat(\Pi)}\tau_2)}}$$

$$for\ \Gamma_1\ not\ empty,\ x_\tau \in VAR^\Pi$$

Since variables from **LexVAR** are never abstracted, their polarity marking is not needed and is explicitly set to *null*. The Elimination rules update the polarity marking of the free variables of the argument term. The Introduction rules mark the type of the dynamically created functional term according to the polarity marking of the abstracted variable corresponding to the discharged assumption.

It is important to note the difference between the actual polarity of an occurrence of $x \in \mathbf{VAR}$ and its abstraction that is marked by $\mathcal{DDL}$. The actual polarity depends on the *denotations* of the applied functional terms, while the (abstracted) polarity marked by $\mathcal{DDL}$ depends on the *marking* of the types of the terms, which is an *abstraction* of these denotations.

The rules of the Order Calculus that is based on $\mathcal{DDL}$ ($\mathcal{DDL}$-OC) are similar to the rules of $\mathcal{L}$-OC except for the fact that $\mathcal{L}$-OC manipulates order statements between PL terms, while the order statements manipulated by $\mathcal{DDL}$-OC are between EPL terms.

For example, consider the derivation of the following inference in $\mathcal{DDL}$-OC :

$$\textsf{Mary does move and John doesn't move} \vdash_{NatLog} \textsf{John doesn't walk}$$

First, let us show how the type of the term $\overrightarrow{\lambda} y.John(doesn't(y))$ is dynamically marked for downward monotonicity during the parsing of John doesn't in $\mathcal{DDL}$ (to shorten the presentation, we use only the proof terms and specify only the polarity of the assumption y):

$$\boxed{y^{\oplus}_{(et)} \triangleright y^{\oplus}_{(et)}} \Rightarrow \boxed{doesn't_{(et)^{-}(et)}(y^{\ominus}_{(et)})} \Rightarrow \boxed{John_{((et)^{+}t)}(doesn't_{(et)^{-}(et)}(y^{\ominus}_{(et)}))}$$

$$\Rightarrow \boxed{(\overrightarrow{\lambda} y.John_{((et)^{+}t)}(doesn't_{(et)^{-}(et)}(y^{\ominus}_{(et)})))_{((et)^{-}t)}}$$

Next, using the non-logical axiom $walk \leq move$, the following order statement can be proven in $\mathcal{DDL}$-OC:

$$\vdash_{\mathcal{DDL}-OC} [\overrightarrow{\lambda} x.Mary(does(x))]and(\overrightarrow{\lambda} y.John(doesn't(y)))_{(et)^{-}(et)}(move) \leq$$

$$\overrightarrow{\lambda} y.John(doesn't(y))_{(et)^{-}(et)}(walk)$$

Now we investigate the relation between $\mathcal{L}$-OC and $\mathcal{DDL}$-OC and show that $\mathcal{DDL}$-OC is weaker than $\mathcal{L}$-OC , that is:

1. $\vdash_{\mathcal{DDL-OC}} \Psi \leq \Phi \ \Rightarrow \vdash_{\mathcal{L-OC}} \Pi strip(\Psi) \leq \Pi strip(\Phi)$

2. $\vdash_{\mathcal{L-OC}} \Pi strip(\Psi) \leq \Pi strip(\Phi) \ \not\Rightarrow \vdash_{\mathcal{DDL-OC}} \Psi \leq \Phi$

Lemma 6.1.3 *Let $\Psi \in$ **EPLTerms** s.t. $x_\sigma \in Free(\Pi strip(\Psi))$ and x is marked for positive (negative) polarity in Ψ. Then $\vdash_{\mathcal{L-OC}} \gamma \leq_\sigma \delta$ ($\vdash_{\mathcal{L-OC}} \delta \leq_\sigma \gamma$) implies $\vdash_{\mathcal{L-OC}}$ $\Pi strip(\Psi)[x/\gamma] \leq_\tau \Pi strip(\Psi)[x/\delta]$.*

Proof: by induction on the complexity of Ψ. We prove only for the positive polarity case; the proof for negative polarity is symmetric.

- $\Psi = x^\oplus$. Then $\Pi strip(\Psi) = x$ and $\vdash_{\mathcal{L-OC}} \gamma \leq_\sigma \delta$ implies $\vdash_{\mathcal{L-OC}} x[x/\gamma] \leq_\tau x[x/\delta]$.
- $\Psi = \Theta_{(\zeta^F \rho)}(\Lambda_\zeta)$. Then $x_\sigma \in Free(\Pi strip(\Theta))$ or $x \in Free(\Pi strip(\Lambda))$.
    - $x \in ST(\Pi strip(\Theta))$. If x is assigned positive polarity in Ψ, then x is marked for positive polarity also in Θ. By the induction hypothesis, $\vdash_{\mathcal{L-OC}} \gamma \leq_\sigma \delta$ implies $\vdash_{\mathcal{L-OC}} \Pi strip(\Theta)[x/\gamma] \leq_{(\zeta\rho)} \Pi strip(\Theta)[x/\delta]$. By applying the FR rule, we can prove

$$\vdash_{\mathcal{L-OC}} \underbrace{(\Pi strip(\Theta)[x/\gamma])(\Pi strip(\Lambda))}_{\Pi strip(\Theta(\Lambda))[x/\gamma]} \leq_\rho \underbrace{(\Pi strip(\Theta)[x/\delta])(\Pi strip(\Lambda))}_{\Pi strip(\Theta(\Lambda))[x/\delta]}$$

    - $x \in Free(\Pi strip(\Lambda))$. If x is marked for positive polarity in Ψ, then (i) either '+'$\in F$ (the type of Θ is marked for upward monotonicity) and x is marked for positive polarity in Λ or (ii) '-'$\in F$ (the type of Θ is marked for downward monotonicity) and x is marked for negative polarity in Λ. Suppose w.l.o.g. that (i) holds. By the induction hypothesis, $\vdash_{\mathcal{L-OC}} \gamma \leq_\sigma \delta$ implies $\vdash_{\mathcal{L-OC}} \Pi strip(\Lambda)[x/\gamma] \leq_\zeta \Pi strip(\Lambda)[x/\delta]$. By applying MON+ rule, we can prove

$$\vdash_{\mathcal{L-OC}} \underbrace{(\Pi strip(\Theta)_{(\zeta^+\rho)})(\Pi strip(\Lambda)_\zeta[x/\gamma])}_{\Pi strip(\Theta(\Lambda))[x/\gamma]} \leq_\rho \underbrace{(\Pi strip(\Theta))_{(\zeta^+\rho)}(\Pi strip(\Lambda)[x/\delta])}_{\Pi strip(\Theta(\Lambda))[x/\delta]}$$

- $\Psi = \overline{\lambda} y_\zeta.\Phi$, $y \neq x$. Since x is marked for positive polarity in Ψ, then it is marked for positive polarity in Φ. By the induction hypothesis, $\vdash_{\mathcal{L-OC}} \gamma \leq_\sigma \delta$ implies $\vdash_{\mathcal{L-OC}} \Pi strip(\Phi)[x/\gamma] \leq_\rho \Pi strip(\Phi)[x/\delta]$. By applying the Ab rule, we can prove

$$\vdash_{\mathcal{L-OC}} \Pi strip(\overline{\lambda} y.\Phi)[x/\gamma] \leq_{(\zeta\rho)} \Pi strip(\overline{\lambda} y.\Phi)[x/\delta]$$

Lemma 6.1.4 *Let* $\Psi, \Phi \in$ ***EPLTerms***. *Then*

$$\vdash_{\mathcal{DDL}-OC} \Psi \leq_\tau \Phi \ \Rightarrow \vdash_{\mathcal{L}-OC} \Pi strip(\Psi) \leq_\tau \Pi strip(\Phi)$$

Proof: By showing the construction of a $\mathcal{L}$-OC proof Ω_2 of $\Pi strip(\Psi) \leq_\tau \Pi strip(\Phi)$ from a $\mathcal{DDL}$-OC proof P_1 of $\Psi \leq_\tau \Phi$.

If the proof contains no MON rule applications based on dynamic marking of types of abstraction terms, then Ω_2 is the proof obtained by replacing all EPL terms in Ω_1 by their mirrors.

Otherwise we delete all monotonicity marking from the types of abstraction terms in Ω_1. Now we have invalid instances of the MON rule, which are w.l.o.g. of form

$$\Omega \quad \dfrac{\boxed{\dfrac{\bigtriangledown}{\Pi strip(\Theta) \leq_\tau \Pi strip(\Delta)}}}{(\overline{\lambda} x_\tau.\Pi strip(\Psi))_{(\tau\rho)}\langle \Pi strip(\Theta_\tau)\rangle \leq_\rho (\overline{\lambda} x_\tau.\Pi strip(\Psi))_{(\tau\rho)}\langle \Pi strip(\Delta_\tau)\rangle}\ \text{MON} \qquad (1)$$

where the type of $\overline{\lambda} x_\tau.\Psi$ is not marked for monotonicity. We choose the innermost invalid MON instance, that is such that does not have invalid MON instances in Ω. Let us note the following facts. First, due to its being the innermost non-valid instance of MON, Ω is a valid $\mathcal{L}$-OC proof of the order statement $\Pi strip(\Theta) \leq \Pi strip(\Delta)$. Second, since the type of $\overline{\lambda} x.\Psi$ is marked for upward monotonicity, x is marked for positive polarity in Ψ. By lemma 6.1.3, $\vdash_{\mathcal{L}-OC} \Pi strip(\Psi)[x/\Pi strip(\Theta)] \leq_{(\tau\rho)} \Pi strip(\Psi)[x/\Pi strip(\Theta)]$. Thus we can replace (1) by the following valid $\mathcal{L}$-OC proof:

$$\dfrac{\dfrac{\emptyset}{\begin{array}{c}\overline{\lambda}x.\Pi strip(\Psi)\langle\Pi strip(\Theta)\rangle\\ \equiv_{(\tau\rho)}\ \Pi strip(\Psi)[x/\Theta]\end{array}}\beta \qquad \boxed{\dfrac{\bigtriangledown}{\begin{array}{c}\Pi strip(\Psi)[x/\Pi strip(\Theta)]\\ \leq_{(\tau\rho)}\ \Pi strip(\Psi)[x/\Pi strip(\Delta)]\end{array}}} \qquad \dfrac{\emptyset}{\begin{array}{c}\Pi strip(\Psi)[x/\Delta]\\ \equiv_{(\tau\rho)}\ \overline{\lambda}x.\Pi strip(\Psi)_{(\tau\rho)}\langle\Pi strip(\Delta)\rangle\end{array}}\beta}{(\overline{\lambda}x.\Pi strip(\Psi))\langle\Pi strip(\Theta)\rangle \leq (\overline{\lambda}x.\Pi strip(\Psi))\langle\Pi strip(\Delta)\rangle}\ \text{TRANS}$$

In this way we can systematically remove all invalid instances of MON, creating a valid $\mathcal{L}$-OC proof.

Note that the other direction of the above lemma does not hold, that is $\vdash_{\mathcal{L}-OC} \Pi strip(\Psi) \leq \Pi strip(\Phi) \not\Rightarrow \vdash_{\mathcal{DDL}-OC} \Psi \leq \Phi$. Consider, for example the inference Mary adores and passionately loves John $\vdash_{NatLog}$ Mary loves John, the $\mathcal{L}$-OC derivation of which is shown in fig. 6.1. It is not derivable in the proposed version of $\mathcal{DDL}$-OC.

We have shown that the $\mathcal{DDL}$-based Order Calculus is weaker than the $\mathcal{L}$-based one[5].

6.2 Multiple derivations

Basing the system on $\mathcal{L}$ brought about the emergence of non-normalized proof terms in $\mathcal{L}$-OC , which we discussed in the previous section. Note that β/η-equivalent proof terms represent semantically equivalent $\mathcal{L}$ derivations. Now we turn to another case - multiple derivations of the same sequent which are not necessarily semantically equivalent.

Assigning several derivation trees to a natural language sentence reflects several semantic readings that the sentence may have, due to reasons like scope ambiguity. For example, the sentence No man loves some woman has two possible semantic readings:

1. There exists such woman that no man loves. Under this reading, the sentence No man loves some tall woman entails the sentence No man loves some woman.

2. There is no such man that loves some woman. Under this reading, the sentence No man loves some woman entails the sentence No man loves some tall woman.

Basing the system on $\mathcal{L}$ allows to capture this kind of scope ambiguity. There are two possible proof terms representing the derivation of the sentence No man loves some woman:

[5] Note that integrating restrictivity marking into $\mathcal{DDL}$ would not help deriving the above inference in the $\mathcal{DDL}$-based Order Calculus, since deriving this inference involves η-reduction.

1. $[\overrightarrow{\lambda} x.(no(man))(loves(x))]some(woman)$ – corresponds to the first reading. The full $\mathcal{L}$ derivation is shown in fig. 6.4.

2. $no(man)(\overleftarrow{\lambda} x.([\overrightarrow{\lambda} y.[x]loves(y)]some(woman)))$ – corresponds to the second reading. The full $\mathcal{L}$ derivation is shown in fig. 6.5.

And indeed, both of the following order statements are derivable in $\mathcal{L}$-OC :

1. $[\overrightarrow{\lambda} y.(no(man))(loves(y))]some(tall(woman)) \leq [\overrightarrow{\lambda} y.(no(man))(loves(y))]some(woman)$ (shown in fig. 6.6)

2. $no(man)(\overleftarrow{\lambda} x.([\overrightarrow{\lambda} y.[x]loves(y)]some(woman)))$
   $\leq no(man)(\overleftarrow{\lambda} x.([\overrightarrow{\lambda} y.[x]loves(y)]some(tall(woman))))$ (shown in fig. 6.7)

Hence we can derive both of the following Natural Logic inferences:

1. No man loves some tall woman$\vdash_{NatLog}$No man loves some woman

2. No man loves some woman$\vdash_{NatLog}$No man loves some tall woman

However, this does not mean that the two sentences are equivalent. We should keep in mind that each case refers to a *different* semantic reading of the sentence.

$$\frac{\textsf{no}}{((s/^{-}(s\backslash np))/^{-}n) : no_{(et)^{-}((et)^{-}t)}} \quad \frac{\textsf{man}}{n : man_{(et)}} \quad \frac{\textsf{loves}}{((s\backslash np)/np) : loves_{(e(et))}} \quad \frac{-}{np : x_e}$$

$$\frac{((s/^{-}(s\backslash np))/^{-}n) : no_{(et)^{-}((et)^{-}t)} \qquad n : man_{(et)}}{(s/^{+}(s\backslash np)) : no(man)_{((et)^{+}t)}}\ (/E)$$

$$\frac{((s\backslash np)/np) : loves_{(e(et))} \qquad np : x_e}{(s\backslash np) : loves(x)}\ (/E)$$

$$\frac{(s/^{+}(s\backslash np)) : no(man)_{((et)^{+}t)} \qquad (s\backslash np) : loves(x)}{s : no(man)(loves(x))}\ (/E)$$

$$\frac{s : no(man)(loves(x))}{(s/np) : \overrightarrow{\lambda} x.no(man)(loves(x))}\ (/I)$$

$$\frac{\textsf{some}}{((s\backslash^{+}(s/np))/^{+}n) : some_{(et)^{+}((et)^{+}t)}} \quad \frac{\textsf{woman}}{n : woman_{(et)}}$$

$$\frac{((s\backslash^{+}(s/np))/^{+}n) : some_{(et)^{+}((et)^{+}t)} \qquad n : woman_{(et)}}{(s\backslash^{+}(s/np)) : some(woman)}\ (/E)$$

$$\frac{(s/np) : \overrightarrow{\lambda} x.no(man)(loves(x)) \qquad (s\backslash^{+}(s/np)) : some(woman)}{s : [\overrightarrow{\lambda} x.no(man)(loves(x))](some(woman))}\ (\backslash E)$$

Figure 6.4: $\mathcal{L}$ derivation of No man loves some woman corresponding to the first semantic reading

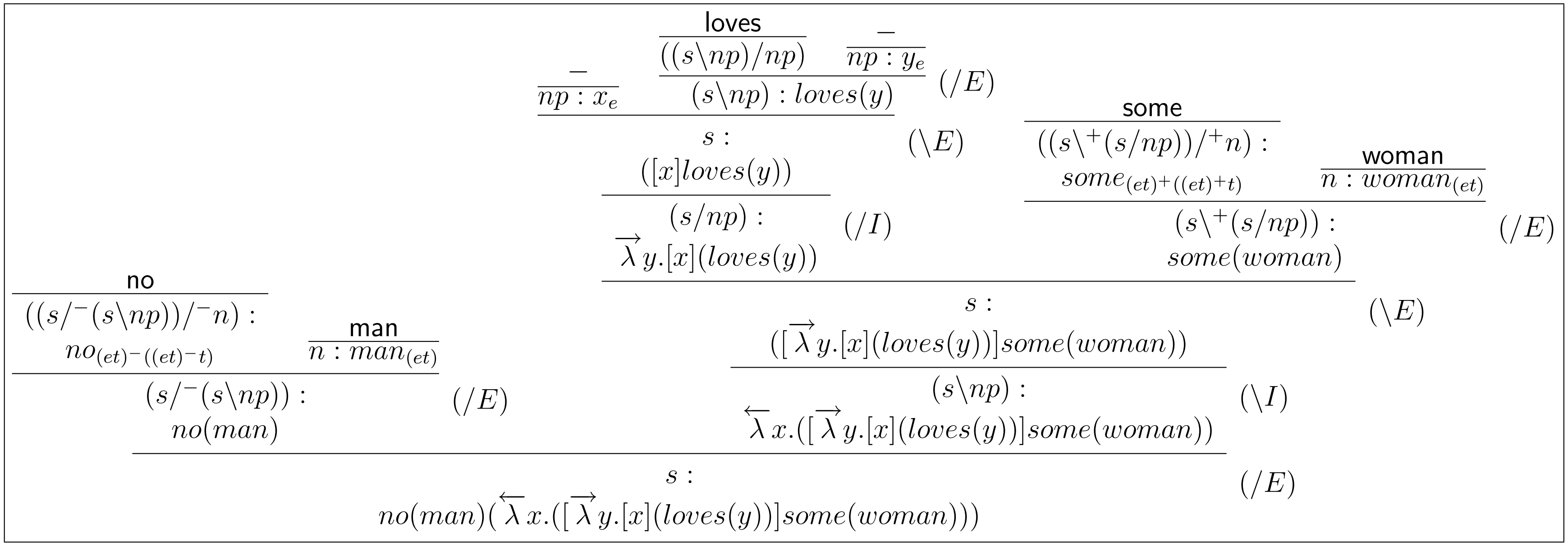


Figure 6.5: $\mathcal{L}$ derivation of No man loves some woman corresponding to the second semantic reading

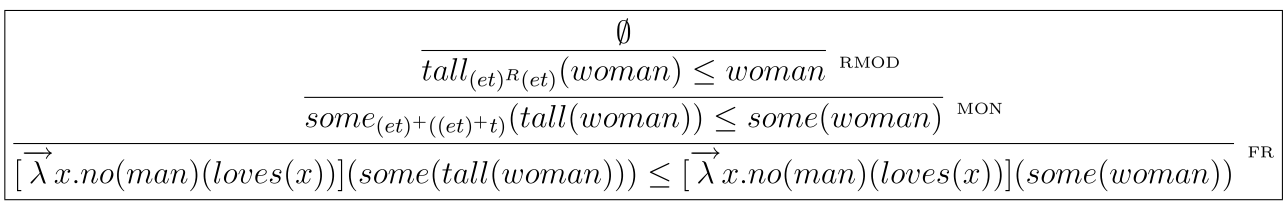


Figure 6.6: $\mathcal{L}$-OC derivation of No man loves some tall woman$\vdash_{NatLog}$No man loves some woman

$$\dfrac{\dfrac{\dfrac{\dfrac{\emptyset}{tall_{(et)^R(et)}(woman) \leq woman}\ \textsc{rmod}}{some_{(et)^+((et)^+t)}(tall(woman)) \leq some(woman)}\ \textsc{mon}}{[\overrightarrow{\lambda} y.[x]loves(y)](some(tall(woman))) \leq [\overrightarrow{\lambda} y.[x]loves(y)](some(woman))}\ \textsc{fr}}{(no(man)_{((et)^-t)})([\overrightarrow{\lambda} y.[x]loves(y)](some(woman)))) \leq (no(man))([\overrightarrow{\lambda} y.[x]loves(y)](some((tall(woman)))))}\ \textsc{mon}$$

Figure 6.7: $\mathcal{L}$-OC derivation of No man loves some woman$\vdash_{NatLog}$No man loves some tall woman

Chapter 7

The Proof Search Procedure

7.1 Description of the algorithm

Below we present a proof search procedure for $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$, which we have proven to be equivalent to $\mathcal{L}$-OC w.r.t. order statements with NF terms. The procedure is a variation on a proposal by [4]. It is a recursive function **derive**$(\alpha_0, \alpha, Goals)$ which, given a (possibly empty) finite set $A = \{[\ \alpha_i, \alpha_i'\ ] \mid 1 \leq i \leq n\}$ of non-logical axioms, searches for a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of a given goal order statement in a top-down manner, attempting to generate simpler subgoals and prove them recursively. In order to prevent the algorithm from diverging, we use the *Goals* parameter, which keeps track of all the pairs of terms that appear as arguments of **derive**.

The proposed algorithm can be used for proving an inference $S_1, ..., S_m \vdash_{NatLog} S$ ($S_1, ..., S_m, S$ are natural language indicative sentences) as follows[1]:

1. the terms $\psi_t^1, ..., \psi_t^m, \psi_t$ representing (NF) $\mathcal{L}$ derivations of $S_1, ..., S_m, S$ resp., are obtained by the $\mathcal{L}$ parser.

2. the order statements $w_t^T \leq \psi_t^i$ for $1 \leq i \leq m$ are added to the set of the non-logical axioms.

3. **derive**$(w_t^T, \psi_t, \emptyset)$ is called.

[1] If at least one of the sentences $S_1, ..., S_m$ is ambiguous, these stages should be repeated for every combination of the corresponding proof terms.

Let us observe the following fact about the axioms C1 and RMOD. Given the left hand term ψ_l in the conclusion $\psi_l \leq \psi_r$ of these axioms, the term ψ_r can be defined as a function of ψ_l, denoted by $f_{C1,r}$, $f'_{C1,r}$, and $f_{RMOD,r}$. Similarly we define the functions $f_{D1,l}$, $f'_{D1,l}$. In C2, the premises can be expressed as a function of the conclusion: $\psi_l \leq f_{C2,1}(\psi_r)$ and $\psi_l \leq f_{C2,2}(\psi_r)$. Based on this observation, we define the following (not necessarily disjoint) classes of $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ rules and axioms: (i) R-PROD – where the righthand term ψ_r in the derived order statement $\psi_l \leq \psi_r$ is a function of the lefthand term ψ_l: REFL, RMOD, C1 and D2, (ii) L-PROD – (the symmetric case): REFL, C2 and D1, (iii) STR – rules in which ψ_r is a replacement of a subterm of ψ_l: FR, MON, (iv) NL – the non-logical axioms.

The **derive** and **subderive** functions are given below[2]. Recall that the algorithm has an implicit parameter $A = \{[\ \alpha_i, \alpha'_i\ ]\ |1 \leq i \leq n\}$ – the set of the non-logical axioms. We denote by $\alpha \in$ **MON** $\uparrow$ ($\alpha \in$ **MON** $\downarrow$) a case of a term $\alpha_{(\tau^+\rho)}$ ($\alpha_{(\tau^-\rho)}$), and abbreviate $\alpha \in$ **MON** $\uparrow \vee \alpha \in$ **MON** $\downarrow$ by $\alpha \in$ **MON**. We refer to a term $\psi_{(\tau^C(\tau\tau))}$ as **coor**^{C} and to $\psi_{(\tau^D(\tau\tau))}$ – as **coor**^{D} .

derive$(\alpha_0, \alpha, Goals) =$

1. If $[\ \alpha_0, \alpha\ ] \in Goals$ then return **false**.
2. $Goals' \leftarrow Goals \cup \{[\ \alpha_0, \alpha\ ]\}$
3. If **subderive**$(\alpha_0, \alpha, Goals')$ then return **true**.
4. for each non-logical axiom $[\ \alpha_i, \alpha'_i\ ]$: if **subderive**$(\alpha_0, \alpha_i, Goals')$ and **derive**$(\alpha'_i, \alpha, Goals')$ then return **true**.
5. return **false**.

subderive$(\alpha_0, \alpha, Goals) =$

1. If $\alpha_0 = \alpha$ then return **true**

2. for each $ax \in \{C1, D1, RMOD\}$

2.1 If $f_{ax,r}(\alpha_0)$ is defined and **derive**$(f_{ax,r}(\alpha_0), \alpha, Goals)$ then return **true**.

2.2 If $f_{ax,l}(\alpha)$ is defined and **derive**$(\alpha_0, f_{ax,l}(\alpha), Goals)$ then return **true**.

3. 3.1 If $f_{D2,1}(\alpha_0)$ and $f_{D2,2}(\alpha_0)$ are defined and **derive**$(f_{D2,1}(\alpha_0), \alpha, Goals)$ and **derive**$(f_{D2,2}(\alpha_0), \alpha, Goals)$ then return **true**.

[2] For abbreviation only the use of the RM set is presented. The LM set is used symmetrically.

3.2 If $f_{C2,1}(\alpha)$ and $f_{C2,2}(\alpha)$ are defined and **derive**$(\alpha_0, f_{C2,1}(\alpha), Goals)$ and **derive**$(\alpha_0, f_{C2,2}(\alpha), Goals)$ then return **true**.

4. If $\alpha_0 = \phi\langle\psi\rangle$ and $\alpha = \gamma\langle\delta\rangle$ and **derive**$(\phi, \gamma, Goals)$

4.1 If $\phi \in$ **MON** $\uparrow$ or $\gamma \in$ **MON** $\uparrow$ and **derive**$(\psi, \delta, Goals)$ return **true**.

4.2 If $\phi \in$ **MON** $\downarrow$ or $\gamma \in$ **MON** $\downarrow$ and **derive**$(\delta, \psi, Goals)$ return **true**.

4.3 If ($\phi \in$ **MON** $\downarrow$ and $\gamma \in$ **MON** $\uparrow$) or ($\phi \in$ **MON** $\uparrow$ and $\gamma \in$ **MON** $\downarrow$) return **true**.

4.4 If **derive**$(\psi, \delta, Goals)$ and **derive**$(\delta, \psi, Goals)$ return **true**.

5. If $\exists\delta_\tau \in RM(\alpha_0)$ and $\exists\delta'_\tau \in RM(\alpha)$ s.t. **derive**$(\delta, \delta', Goals)$ and **derive**$(\delta', \delta, Goals)$, then for each such (δ, δ'):

5.1 for each $ax \in \{C1, D1, RMOD\}$:

5.1.1 If $\alpha_0 = \psi(\delta)$ and $f_{ax,r}(\psi)$ is defined and **derive**$(\overline{norm}(f_{ax,r}(\psi)(\delta)), \alpha, Goals)$ then return **true**.

5.1.2 If $\alpha = \psi(\delta')$ and $f_{ax,l}(\psi)$ is defined and **derive**$(\alpha_0, \overline{norm}(f_{ax,l}(\psi)(\delta')), Goals)$ then return **true**.

5.2 If $\alpha_0 = \psi(\delta)$ and $f_{D2,1}(\psi)$ and $f_{D2,2}(\psi)$ are defined and **derive**$(\overline{norm}(f_{D2,1}(\psi)(\delta)), \alpha, Goals)$ and **derive**$(\overline{norm}(f_{D2,2}(\psi)(\delta)), \alpha, Goals)$ then return **true**.

5.3 If $\alpha = \psi(\delta')$ and $f_{C2,1}(\psi)$ and $f_{C2,2}(\psi)$ are defined and **derive**$(\alpha_0, \overline{norm}(f_{C2,1}(\psi)(\delta')), Goals)$ and **derive**$(\alpha_0, \overline{norm}(f_{C2,2}(\psi)(\delta')), Goals)$ then return **true**.

5.4 For each two non-logical axioms [α_i, α'_i] and [α_j, α'_j]:

5.4.1 If $\overline{norm}(\alpha_i(\delta)) = \alpha_0$ and $\overline{norm}(\alpha'_j(\delta')) = \alpha$ and **derive**$(\alpha'_i, \alpha_j, Goals)$ then return **true**.

5.4.2 If $\overline{norm}(\alpha_i(\delta)) = \alpha_0$ and $\alpha = \psi(\delta')$ and **derive**$(\alpha'_i, \psi, Goals)$ then return **true**.

5.4.3 If $\overline{norm}(\alpha'_j(\delta')) = \alpha$ and $\alpha_0 = \psi(\delta)$ and **derive**$(\psi, \alpha_j, Goals)$ then return **true**.

6. If $\alpha_0 = \overline{\lambda}x.\psi^x$ and $\alpha = \overline{\lambda}x.\varphi^x$ and **derive**$(\psi^x, \varphi^x, Goals)$ then return **true**.

7. **7.1** If $\alpha_0 = \overline{\lambda} x_\tau . \psi^{x_\tau}$ and $x \notin ST(\alpha)$ and **derive**$(\psi^{x_\tau}, \alpha\langle x_\tau \rangle, Goals)$ then return **true**.

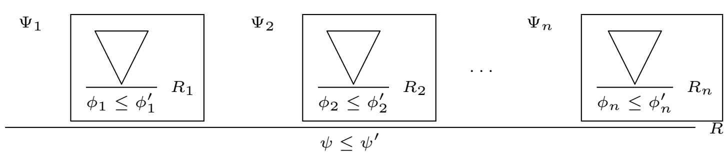
7.2 If $\alpha = \overline{\lambda} x_\tau . \psi^{x_\tau}$ and $x \notin ST(\alpha_0)$ and **derive**$(\alpha_0\langle x_\tau \rangle, \psi^{x_\tau}, Goals)$ then return **true**.

8. return **false**.

The algorithm searches for $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proofs of a specific form, which we call *the* $\mathcal{L}$-*OC*$^{\hat{FR},\hat{Ab}}$ *canonic proofs*. Let $\mathcal{R}$ be the regular language NL* R-PROD* ((STR | $\hat{\text{FR}}$)* | $\hat{\text{Ab}}$ | Ab) L-PROD* (NL$^+$ R-PROD* ((STR | $\hat{\text{FR}}$)* | $\hat{\text{Ab}}$ | Ab) L-PROD*)* NL*.

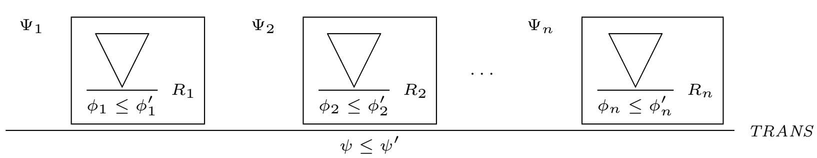
Definition 7.1.1 ($\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ canonic proof) *Let* $\vdash_{\mathcal{L}-OC^{\hat{FR},\hat{Ab}}} \psi \leq \psi'$. *Then a* canonic proof *of* $\psi \leq \psi'$ *is one of the following structures:*

1. *A* Type 1 *canonic proof is*



where (i) $R \neq TRANS$ *, (ii)* $n \in \{1, 2\}$*, (iii)* $\Psi_1, \ldots, \Psi_n$ *are canonic proofs.*

2. *A* Type 2 *canonic proof is*



where (i) $n \geq 2$*, (ii)* $\Psi_1, \ldots, \Psi_n$ *are Type 1 canonic proofs, (iii)* $\psi = \phi_1, \phi'_1 = \phi_2, \phi'_2 = \phi_3, \ldots, \phi'_{n-1} = \phi_n, \phi'_n = \psi'$*, (iv) the string formed by the rules* $R_1 ... R_n$ *belongs to the regular language* $\mathcal{R}$.

The algorithm attempts to find a canonic proof by first separating it into a number of canonic subproofs of Type 2 (step 4 in **derive**), and then by restoring these subproofs from their terminal terms α_0, α (in **subderive**). The functions we defined for the rules C1, C2, D1, D2, and RMOD are used to search for shorter subproofs (steps 2 and 3 in **subderive**). MON, FR, Ab and $\hat{\text{Ab}}$ are also treated straightforwardly (steps 4,6,7 in **subderive**). $\hat{\text{FR}}$ rules are treated in two different ways: (a) in steps 5.1–5.3 of **subderive** the algorithm tries to construct the righthand/lefthand side of the result of $\hat{\text{FR}}$ from its lefthand/righthand side, (b) in step 5.4 of **subderive** the algorithm tries to recover the premises of $\hat{\text{FR}}$ from its conclusion.

7.2 Correctness

Below we briefly discuss the correctness of the proposed algorithm. The full proof of *termination* and *completeness* results appear in appendixes A and B resp.

The *soundness* of the algorithm, that is the fact that any order statement for which the algorithm returns **true** has a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof, follows quite straightforwardly from its definition.

Termination is guaranteed by the following facts. First, it can be shown that only a finite set of terms can appear as arguments in the call tree of the **derive** function. This set depends on the subterms of terms that appear in the goal order statement and the non-logical axioms. Secondly, due to the *Goals* parameter, any pair of terms can appear as arguments in the call tree of **derive** at most once.

As for *completeness* results, it can be shown that for any order statement that has a $\hat{\text{FR}}$-free $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof, that it is provable by the algorithm. The proof has two stages:

1. Under some limitations on the lexicon and $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ -derivable order-statements (defined in appendix B), any order statement that has a $\hat{\text{FR}}$ -free $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof has a ($\hat{\text{FR}}$ -free) canonic proof.

2. Any order statement that has a $\hat{\text{FR}}$-free canonic proof is provable by the algorithm.

The algorithm also finds $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proofs with instances of $\hat{\text{FR}}$ of several forms. In fig. 7.1 we show the call tree of the procedure for the $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of the order statement $[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move) \leq Mary(doesn't(walk))$ appearing in fig. 6.3.

To demonstrate why the $\hat{\text{FR}}$ is problematic, consider the following $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof:

$$\dfrac{\dfrac{\boxed{\dfrac{\nabla}{\psi \leq \overline{\lambda} x.\alpha(\gamma(\delta(x)))}} \quad \boxed{\dfrac{\nabla}{\zeta \equiv \theta}}}{\psi(\zeta) \leq \alpha(\gamma(\delta(\theta)))}\,\hat{\text{FR}} \qquad \dfrac{\boxed{\dfrac{\nabla}{\overline{\lambda} y.\alpha(\gamma(y)) \leq \phi}} \quad \boxed{\dfrac{\nabla}{\delta(\theta) \equiv \mu}}}{\alpha(\gamma(\delta(\theta))) \leq \phi(\mu)}\,\hat{\text{FR}}}{\psi(\zeta) \leq \phi(\mu)}\,\text{TRANS}$$

The problem is that there is no direct relation between the terms μ and ζ (they are non-directly related through the term $\delta(\theta)$), and while attempting to prove the order statement $\psi(\zeta) \leq \phi(\mu)$, the current algorithm has no way of constructing the term $\alpha(\gamma(\delta(\theta)))$.

derive$([\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move), Mary(doesn't(walk)), \emptyset)$

In step 3 of **derive**:

subderive$([\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move), Mary(doesn't(walk)), Goals')$

$\big(Goals' = \{< [\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move), Mary(doesn't(walk)) >\}\big)$

In step 5.1.1. of **subderive**:

$f_{ax,r}([\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))) = (\overrightarrow{\lambda} y.Mary(doesn't(y)))$

$\overline{norm}(\overrightarrow{\lambda} y.Mary(doesn't(y))(move)) = Mary(doesn't(move))$

derive$(Mary(doesn't(move)), Mary(doesn't(walk)), Goals')$

In step 3 of **derive**:

subderive$(Mary(doesn't(move)), Mary(doesn't(walk)), Goals'')$

$\big(Goals'' = \{< [\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move), Mary(doesn't(walk)) >,$
$< Mary(doesn't(move)), Mary(doesn't(walk)) >\}\big)$

In step 4 of **subderive**:

derive$(Mary, Mary, Goals'')$

In step 3 of **derive**:

subderive$(Mary, Mary, Goals''')$

In step 1 of **subderive**: return **true**.

derive$(doesn't(walk), doesn't(move), Goals'')$

In step 3 of **derive**:

subderive$(doesn't(walk), doesn't(move), Goals''')$

In step 4 of **subderive**:

derive $(doesn't, doesn't, Goals'''')$: return **true**.

derive$(move, walk, Goals''''')$: return **true**.

return **true**.

return **true**.

return **true**.

return **true**.

return **true**.

return **true**.

Figure 7.1: The call tree of the algorithm for the $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof from fig. 6.3

Chapter 8

Conclusions

In this thesis we have proposed a Natural Logic inference system that is based on $\mathcal{L}$, transcending a previous system of [4] based on the AB calculus. We have shown that extending the system to be based on $\mathcal{L}$ is rewarding, as it allows deriving new kinds of inferences, such as inferences involving sentences with extraction, pied piping etc.
It is clear, however, that $\mathcal{L}$ is not the optimal categorial formalism to underly a Natural Logic inference system, due to its own syntactic limitations, such as overgeneration, incapability to deal with non-peripheral extraction, etc. For example, the grammaticality of the following sentence cannot be derived in $\mathcal{L}$:

The boy whom Mary loves dearly smiled.

Hence, an $\mathcal{L}$-based Natural Logic inference system cannot account for the following valid entailment:

The boy whom Mary loves dearly smiled $\Rightarrow$ The boy whom Mary loves smiled

Thus we view the proposed inference system as an intermediate step towards a more complex one, to be finally based on some decidable fragment of the multi-modal type-logical grammar. Much work still has to be done in the direction of extending the inference system to be based on more complex categorial formalisms. Another possible research direction is looking for more semantic properties, which in addition to monotonicity, restrictivity etc. can be used to account for natural language inferences.

We believe, however, that the present work has shown some advances in extending the Natural Logic paradigm to a more substantial system of reasoning in natural language.

Appendix A

Termination of the proof search procedure

We will now prove that the proof search algorithm terminates, by showing that only a finite set of terms can appear as arguments in the call tree of **derive**. Since no pair of terms can appear more than once in a call tree, termination follows.

We will need the following definitions for describing the finite set of the possible arguments of the algorithm.

Definition A.1 (Wrapping) *Given a term* $\varphi_{(\sigma_1(\ldots(\sigma_{n-1}\sigma_n)\ldots))}$ *and an ordered finite sequence of variables from* ***VAR****:* $S = \overline{x_{\sigma_1}\ldots x_{\sigma_{n-1}}}$, *the wrapping of* ψ *w.r.t.* S *is* $\lceil\varphi\rceil^S = \langle\ldots\langle\varphi\langle x_{\sigma_1}\rangle\rangle\ldots\langle x_{\sigma_{n-1}}\rangle\rangle_{\sigma_n}$.

Definition A.2 (Wrapping closure) *Let* $\psi_{(\sigma_1(\ldots(\sigma_{n-1}\sigma_n)\ldots))}$ *be a term and* $A = \{x^1_{\sigma_1}, \ldots, x^m_{\sigma_m}\}$ *a set of variables from* ***VAR****. Then the wrapping closure of* ψ *w.r.t.* A *is defined as follows:*

$$\mathcal{WC}_A(\psi) = \{\lceil\psi\rceil^S \mid S = \overline{x^{i_1}_{\sigma_{i_1}}\ldots x^{i_k}_{\sigma_{i_k}}},\ for\ \{x^{i_1}_{\sigma_{i_1}}, \ldots, x^{i_k}_{\sigma_{i_k}}\} \subseteq A,\ 1 \leq k \leq m,\ \lceil\psi\rceil^S\ is\ defined\}$$

Definition A.3 $(Sub_{S_1,S_2}(\psi))$ *Let* ψ *be a term and* $S_1 = \overline{x^1_{\tau_1}\ldots x^n_{\tau_n}}, S_2 = \overline{y^1_{\tau_1}\ldots y^n_{\tau_n}}$ *two finite ordered sequences of variables (of equal size) from* ***VAR*** *s.t. all members of* S_1 *are free* *in* ψ. *Then*

$$Sub_{S_1,S_2}(\psi) = \psi[y^1_{\tau_1}/x^1_{\tau_1}, ..., y^n_{\tau_n}/x^n_{\tau_n}]$$

In words, $Sub_{S_1,S_2}(\psi)$ is the term obtained from ψ by substituting its free variables from S_1 by the respective variables from S_2.

Definition A.4 ($\mathcal{SUB}_V(\psi)$) *Let* ψ *be a term and* V *a finite set of variables from* ***VAR***: $V = \{y^1_{\tau_1}, ..., y^n_{\tau_n}\}$. *The set* $\mathcal{SUB}_V(\psi)$ *is defined as follows:*

$$\mathcal{SUB}_V(\psi) = \{Sub_{S_1,S_2}(\psi) \mid S_1 = \overline{x^{i_1}_{\tau_{i_1}}, ..., x^{i_m}_{\tau_{i_m}}},\ S_2 = \overline{y^{i_1}_{\tau_{i_1}}, ..., x^{i_m}_{\tau_{i_m}}},\ \{\, x^{i_1}_{\tau_{i_1}}, ..., x^{i_m}_{\tau_{i_m}}\} \subseteq Free(\psi),$$

$$1 \leq m \leq n,\ \{y^{i_1}_{\tau_{i_1}}, ..., y^{i_m}_{\tau_{i_m}}\} \subseteq V,\ Sub_{S_1,S_2}(\psi)\ is\ defined\}$$

In words, the set $\mathcal{SUB}_V(\psi)$ is the set of terms obtained from ψ by substituting some of its free variables from **VAR** by variables from a given set V.

Next, we define *coordination and RMOD transformations* for some term ψ. Let ψ be a term s.t. $(\gamma\ \mathbf{coor}^{C/D}\ \delta)(\varphi) \in ST(\psi)$. Then applying coordination transformation to ψ is replacing $(\gamma\ \mathbf{coor}^{C/D}\ \delta)(\varphi)$ by $(\overline{norm}(\gamma(\varphi))$ or by $\overline{norm}(\delta(\varphi)))$ in ψ. Now let μ be another term s.t. $(\phi_{(\tau^{R}\tau)}(\alpha))(\varphi) \in ST(\mu)$. Then applying RMOD transformation to ψ is replacing $(\phi_{(\tau^{R}\tau)}(\alpha_\tau))(\varphi)$ by $\overline{norm}(\alpha(\varphi))$ in μ.

Definition A.5 (Set of all available subterms) *Let* α_0, α *(* α_0, α *are NF terms) be the terms supplied as arguments to* ***derive***. *Let* (ax_i) $\alpha_i \leq \alpha'_i$ *for* $1 \leq i \leq n$ *be the finite set of non-logical axioms (we assume that* α_i, α'_i *are NF terms). Let* B *be the set* $\{\alpha_0, \alpha\} \cup \{\alpha_i, \alpha'_i \mid 1 \leq i \leq n\}$. *Let* V *be the set* $\{x \mid x \in$ ***VAR*** *and* $x \in \bigcup_{\varphi \in B} ST(\varphi)\}$.
Let V' *be the set* $\{x_{(\tau\rho)}\langle y_\tau\rangle \mid x_{(\tau\rho)}, y_\tau \in V\}$.
Let B' *be the set of all terms obtained from terms in* B *by applying some number of coordination and RMOD transformations.*
Let the set $A = \bigcup_{\gamma \in B \cup B'} ST(\gamma)$.
Let C_V *be the set* $\bigcup_{\gamma \in A} \mathcal{SUB}_V(\gamma)$.
Let Ext *be the set* $\bigcup_{\psi \in A \cup C_V \cup V'} \mathcal{WC}_V(\psi)$. *Then the set* $AS = A \cup C_V \cup Ext \cup V'$ *is the set of all available subterms.*

Observation A.1 *For any α, α_0 and any finite set A of non-logical axioms, AS is a finite set.*

Lemma A.1 *Let $\psi \in AS$. Then for each $\varphi \in ST(\psi)$, $\varphi \in AS$.*

Proof: Let $\psi \in AS = A \cup C_V \cup Ext \cup V'$ and $\varphi \in ST(\psi)$. Let us show that $\varphi \in AS$. One of the following holds:

1. $\psi \in A = \bigcup_{\gamma \in B \cup B'} ST(\gamma)$. Then $\varphi \in ST(\gamma)$ for $\gamma \in B \cup B'$. Since $ST(\gamma) \subseteq A$, $\varphi \in A \subseteq AS$.

2. $\psi \in C_V$. Then there exists some $\psi' \in A$ s.t. ψ is obtained from ψ' by substituting some of its free variables from **VAR** by variables from V. ψ' has a subterm φ' s.t. φ is obtained from φ' by substituting some of its free variables from **VAR** by variables from V. Thus $\varphi \in C_V \subseteq AS$.

3. $\psi \in Ext$. Then $\psi = \langle ... \langle \mu \langle x_1 \rangle ... \rangle \langle x_n \rangle$ for $\mu \in A \cup C_V \cup V'$ and $x_1, ..., x_n \in V \subseteq A$. Then all proper subterms of ψ (that is, all subterms that are not ψ itself) are in $A \cup Ext$.

4. $\psi \in V'$. Then the proper subterms of ψ are in $V \subseteq A$.

Lemma A.2 (Available pairs) *Given the (NF) goal terms α, α_0, only terms from AS appear as arguments in the call tree of* ***derive****.*

Proof: by induction on d — the depth of the call in the call tree.

Base: $d = 0$. The first call is **derive**$(\alpha_0, \alpha, \emptyset)$. $\alpha_0, \alpha \in B \subseteq AS$.

Assumption: For depth less or equal to d the lemma holds.

Step: The depth of the call is $d + 1$. One of the following holds:

- **derive** was called in **derive**$(\psi, \phi, Goals)$ at step 4, that is the call is **derive**$(\alpha_i', \phi, Goals')$. $\phi \in AS$ by the induction hypothesis and $\alpha_i' \in B \subseteq AS$.

- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 2.1, that is $\alpha_0 = f_{ax,r}(\psi)$, $\alpha = \phi$. $\psi, \phi \in AS$ by the induction hypothesis. Thus $\alpha \in AS$. Let us prove that $\alpha_0 \in AS$. The function $f_{ax,r}$ is one of:

1. $f_{RMOD,r}$. Then $\psi = \mu_{(\tau R_\tau)}(\gamma)$ for some terms μ, γ and $\alpha_0 = \gamma$. Since by the induction hypothesis $\psi \in AS$, by lemma A.1 its subterm α_0 is also in AS.

2. $f_{C1,r}$. Then α_0 is a subterm of $\psi \in AS$ and by lemma A.1, $\alpha_0 \in AS$.

- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 2.2. The proof is symmetric.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 3.2, that is $\alpha_0 = f_{C2,1}(\psi)$ or $\alpha_0 = f_{C2,1}(T_0)$ and $\alpha = \phi$. The proof is similar to the previous case.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 3.1, that is $\alpha = f_{D2,1}(\phi)$ or $\alpha = f_{D2,1}(\phi)$ and $\alpha_0 = \psi$. The proof is similar to the previous case.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 4, that is $\psi = C_1(D_1)$. Then α_0, α are subterms of terms in AS, and by lemma A.1 they are in AS.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 6, that is $\psi = \overline{\lambda} x.\mu^x$ and $\phi = \overline{\lambda} x.\gamma^x$ for some terms μ, γ. Then $\alpha_0 = \mu^x$ and $\alpha = \gamma^x$. By the induction hypothesis, $\psi, \phi \in AS$, thus by lemma A.1 α_0, α are also in AS.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 5.1.1. That is, $\alpha = \phi$, $\psi = \mu(\delta)$ and $\alpha_0 = \overline{norm}(f_{ax,r}(\mu)(\delta))$. By the induction hypothesis, $\psi, \phi \in AS$ and so is α. ax can be either C1 or RMOD.

  If ax =C1, then w.l.o.g. $\psi = (\mu\ \mathbf{coor}^C\ \gamma)(\delta)$ for some terms μ, γ and $f_{ax,r}(\psi) = \mu$. $\psi \in AS = A \cup C_V \cup Ext \cup V'$. Then one of the following cases holds:

  1. If $\psi \in A$ then there exists some term $\Psi \in B \cup B'$ s.t. $\psi \in ST(\Psi)$. Let the term Ψ' be obtained from Ψ by replacing its subterm ψ by $\overline{norm}(\mu(\delta))$. By definition of B', $\Psi' \in B'$ and $\alpha_0 = \overline{norm}(\mu(\delta)) \in ST(\Psi') \subseteq AS$.

  2. If $\psi \in C_V$ then there exists some $\psi' \in A$ s.t. ψ is obtained from ψ' by substituting some of its free variables from **VAR** by variables from V. That is, $\psi' = (\mu'\ \mathbf{coor}^C\ \gamma')(\delta')$ s.t. μ, γ, δ are obtained from μ', γ', δ' resp. by substituting some of their free variables from **VAR** by variables from V. Since $\psi' \in A$, there exists some $\Psi' \in B \cup B'$ s.t. $\psi' \in ST(\Psi')$. Let the term Ψ'' be obtained from Ψ' by replacing its subterm $(\mu'\ \mathbf{coor}^C\ \gamma')(\delta')$ by $\overline{norm}(\mu'(\delta'))$. By definition of B', $\Psi'' \in B'$. Then $\overline{norm}(\mu'(\delta')) \in A$ and by definition of C_V, $\alpha_0 = \overline{norm}(\mu(\delta)) \in C_V$.

3. If $\psi \in Ext$, then $\delta \in V$. If $\mu \in V$, then $\overline{norm}(\mu(\delta)) = \mu(\delta) \in V'$.

   Otherwise $(\mu \ \mathbf{coor}^C \ \gamma) \in A \cup C_V$ and $\mu \in A \cup C_V$. If μ is not an abstraction term, then $\alpha_0 = \overline{norm}(\mu(\delta)) = \mu(\delta) \in Ext$. Otherwise $\mu = \overline{\lambda} x.\zeta^x$ and $\overline{norm}(\mu(\delta)) = \zeta^x[x/\delta]$. Since $\mu \in A \cup C_V$, also $\zeta^x \in A \cup C_V$ and $\alpha_0 = \zeta^x[x/\delta] \in C_V$.

4. $\psi \in V'$ – impossible, since $\mathbf{coor}^C \in ST(\psi)$.

Otherwise ax =RMOD. Then w.l.o.g. $\psi = (\mu_{(\tau^R \tau)}(\gamma))(\delta)$, $\alpha_0 = \overline{norm}(\gamma(\delta))$ and $\alpha = \phi \in AS$. $\psi \in AS = A \cup C_V \cup Ext \cup V'$. One of the following possibilities holds for ψ:

1. $\psi \in A$. Then there exists some term $\Psi \in B \cup B'$ s.t. $\psi \in ST(\Psi)$. Let Ψ' be the term obtained from Ψ by replacing its subterm ψ by $\overline{norm}(\gamma(\delta))$. By definition of B', $\Psi' \in B'$, thus $\alpha_0 = \overline{norm}(\gamma(\delta)) \in A \subseteq AS$.

2. $\psi \in C_V$. Then there exists some term $\psi' \in A$ s.t. ψ is obtained from ψ' by substituting some of its free variables from **VAR** by variables from V. That is, $\psi' = (\varphi'(\gamma'))(\delta')$ s.t. φ, γ, δ are obtained from $\varphi', \gamma', \delta'$ resp. by substituting some of their free variables from **VAR** by other variables from V. Since $\psi' \in A$ there exists some term $\Psi' \in B \cup B'$ s.t. $\psi' \in ST(\Psi')$. Let Ψ'' be the term obtained from Ψ' by replacing its subterm ψ' by $\overline{norm}(\gamma'(\delta'))$. By definition of B', $\Psi'' \in B'$ and thus $\overline{norm}(\gamma'(\delta')) \in A$. Therefore, $\alpha_0 = \overline{norm}(\gamma(\delta))$, which is obtained from $\overline{norm}(\gamma'(\delta'))$ by substituting some of its free variables from **VAR** by some variables from V is in C_V.

3. $\psi \in Ext$. Then $\delta \in V$. If also $\gamma \in V$, then $\overline{norm}(\gamma(\delta)) = \gamma(\delta) \in V'$.

   Otherwise $\mu(\gamma) \in A \cup C_V \cup V'$. If $\mu(\gamma) \in V'$, then $\mu, \gamma \in V$ and $\overline{norm}(\gamma(\delta)) = \gamma(\delta) \in V'$.

   If $\mu(\gamma) \in A \cup C_V$, then $\gamma \in A \cup C_V$. If γ is an abstraction term $\overline{\lambda} x.\zeta^x$, then $\zeta^x \in A \cup C_V$ and $\overline{norm}(\gamma(\delta)) = \zeta^x[x/\delta] \in C_V$. Otherwise $\overline{norm}(\gamma(\delta)) = \gamma(\delta) \in Ext$.

4. $\psi \in V'$ – impossible due to the structure of ψ.

- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 5.5.2. Then $\alpha_0 = \alpha_i \in B \subseteq AS$. $\phi = \mu(\delta)$ and $\alpha = \mu$. By the induction hypothesis, $\phi \in AS$ and so is its subterm α.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 5.5.1/5.5.3 – the proof is similar to the previous case.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 7.1. Then $\psi = \overline{\lambda} x.\mu^x$ and $\alpha_0 = \mu^x$. By the induction hypothesis, ψ is in AS and so is its subterm α_0. Also, w.l.o.g. $\alpha = \phi(x)$. Note that since $\mu \in AS$, $x \in V$. By the induction hypothesis, $\phi \in AS = A \cup C_V \cup Ext \cup V'$. By definition of Ext, $\phi(x) \in Ext$.
- **derive** was called in **subderive**$(\psi, \phi, Goals)$ at Step 7.2. The proof is similar to the previous case.

Termination immediately follows from the above lemma, since (i) there is a finite set of terms that can appear as arguments in the call tree of **derive**, and (ii) no pair of terms appears more than once. Thus the call tree of **derive** is always finite, and the algorithm terminates when it is traversed.

Appendix B

Completeness of the proof search procedure

We prove that the proof search procedure defined above is complete with respect to $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proofs with no instances of $\hat{\mathrm{FR}}$ rule (henceforth *$\hat{F}R$-free* proofs). The proof has the following stages:

1. Any order statement that has a $\hat{\mathrm{FR}}$-free $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof has a ($\hat{\mathrm{FR}}$-free) canonic proof.

2. Any order statement that has a $\hat{\mathrm{FR}}$-free canonic proof is provable by the algorithm.

First we prove that any order statement $\alpha \leq \gamma$ that has a $\hat{\mathrm{FR}}$-free $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof P has a canonic proof by constructing a canonic proof of $\alpha \leq \gamma$ from P. We rely on the following assumptions[1]:

1. (*) No coordinator is marked for conjunctive and disjunctive behaviour simultaneously.

2. (*) No non-logical axiom is an order statement $\alpha \leq \beta$ or $\beta \leq \alpha$, for a term $\alpha_{(\tau\tau)}$ for a "half-coordinated" expression *coor X*.

3. (*) A term α of type $(\tau\tau)$ for a "half-coordinated" expression *coor X* combines only with terms of type τ (and not of type $(\rho(\tau\tau))$.

[1]The assumptions defined in [4] are marked with (*).

4. (*) There does not exist a disjunctive or conjunctive coordinator in the lexicon (or derivable from the lexicon), s.t. the resulting category in coordination is marked for restrictivity.

5. (*) No terms of form $\alpha(\gamma)$ are marked for restrictivity.

6. No order statements of the following forms are derivable in $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$:

   (a) (*) An order statement containing a proof term the denotation of which is a constant zero function, that is a function that sends every argument to the bottom element of the appropriate domain.

   (b) An order statement of form $\psi \leq (\phi\langle x\rangle)\langle y\rangle$ s.t. $x, y \in$ **VAR** and $\psi \neq (\phi'\langle x\rangle)\langle y\rangle$ for some term ϕ'.

7. There are no free variables from **VAR** in the terms of the non-logical axioms.

Observation B.1 *Given the above assumptions, let P be the following* $\mathcal{L}$-*OC*$^{\hat{FR},\hat{Ab}}$ *proof of size s:*

$$\dfrac{\boxed{\dfrac{\nabla}{\alpha \leq \gamma}\ R}}{\psi \leq \phi\langle x\rangle}\ \hat{Ab}$$

Then a proof P' of $\psi \leq \phi\langle x\rangle$ *of size at most s of the following form can be constructed:*

$$\dfrac{\boxed{\dfrac{\nabla}{\alpha \leq \gamma}}}{\psi \leq \phi\langle x\rangle}\ R$$

s.t. $R \neq \hat{Ab}$.

Proof: By assumption 6b, $\psi = \phi'\langle x\rangle$. Then P is of form:

$$\dfrac{\boxed{\dfrac{\nabla}{\alpha = \phi'\langle x\rangle\langle y\rangle \leq \phi\langle x\rangle\langle y\rangle = \gamma}}}{\psi = \phi'\langle x\rangle \leq \phi\langle x\rangle}\ \hat{\mathrm{Ab}}$$

for some $y \in \mathbf{VAR}$. Then it can be shown (using assumption 7 and the side conditions of the coordination rules) that P has a subproof Ω of $\phi' \leq \phi$ of size at most $s - 1$. Thus P' can be constructed as follows:

$$\frac{\Omega}{\psi = \phi'\langle x \rangle \leq \phi\langle x \rangle}\ \text{FR}$$

Lemma B.1 *Let P be an $\hat{FR}$-free $\mathcal{L}$-$OC^{\hat{FR},\hat{Ab}}$ proof of size s s.t. its main rule is R and $x \in$ **VAR**:*

$$\boxed{\frac{\bigtriangledown}{\phi^x \leq \psi\langle x \rangle}\ R}$$

Then P can be rewritten s.t. its size is at most s and R is one of the following rules: (i) REFL, (ii) RMOD, (iii) FR, or (v) TRANS.

Proof:

- R is not C1/D1/C2/D2, since variables from **VAR** can not appear in the conclusion of these rules.
- R is not Ab, since the right handside of its conclusion is not an abstraction term.
- IF R is $\hat{\text{Ab}}$ then by observation B.1 the proof can be rewritten.
- R is not NL, since by assumption 7 free variables from **VAR** do not appear in non-logical axioms.
- If R is MON, then the proof is of the following form:

$$\frac{\boxed{\dfrac{\bigtriangledown}{x_\tau \leq x_\tau}}}{\psi_{(\tau+\rho)}(x_\tau) \leq \psi_{(\tau+\rho)}(x_\tau)}\ \text{MON}$$

  and can be replaced by an instance of REFL.

Next, following [4], we define a *pre-canonic $\mathcal{L}$-$OC^{\hat{FR},\hat{Ab}}$ proof* as a canonic proof without the requirement (*) that the string $R_1 ... R_n$ belongs to $\mathcal{R}$. That is, we only require that the TRANS rule occurrences are 'flattened'. It is straightforward to show that any order statement that has a $\mathcal{L}$-$\text{OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof, has a pre-canonic $\mathcal{L}$-$\text{OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof. Types 1 and 2 of pre-canonic proofs are defined similarly to the ones of canonic proofs (without requirement (*)).

Lemma B.2 *(***Canonic proof existence***) Let the order statement $\alpha_0 \leq \alpha$ have a pre-canonic $\hat{F}R$-free $\mathcal{L}$-$OC^{\hat{FR},\hat{Ab}}$ proof P of size s. Then there exists a ($\hat{F}R$-free) canonic proof of $\alpha_0 \leq \alpha$ of size at most s.*

Proof: We construct a canonic proof from the pre-canonic proof P. If the requirement (*) is satisfied, then the proof is already canonic. Otherwise there is a violation of (*), that is a violation of the relative ordering between R-PROD, L-PROD, NL, STR, Ab and $\hat{\text{A}}$b. We divide the violations into 3 mutually disjoint classes:

1. impossible – violations that can not occur in $\mathcal{L}$-$\text{OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proofs:
   - D1, C1
   - C2, D2
   - MON, C1
   - MON, D2
   - C2, MON
   - D1, MON
   - Ab, C1
   - Ab, D2
   - Ab, RMOD
   - Ab, STR
2. removable – violations that can be removed from $\mathcal{L}$-$\text{OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proofs:
   - C2, C1
   - D1, D2
   - FR, C1
   - D1, FR
3. rewritable – violations that can be rewritten, producing a $\mathcal{L}$-$\text{OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of at most the same size, with less violations:
   - STR, D2
   - $\hat{\text{A}}$b, C1

- $\widehat{\text{Ab}}$, D2
- C2, STR
- STR, RMOD
- $\widehat{\text{Ab}}$, RMOD
- $\widehat{\text{Ab}}$, STR
- Ab/$\widehat{\text{Ab}}$, Ab/$\widehat{\text{Ab}}$

Below we treat the violations involving the rules $\widehat{\text{Ab}}$ and Ab. The proof of [4] can be easily adapted to our formulation for the rest of the violations.

Impossible violations.

The following violations are impossible in $\mathcal{L}$-OC$^{\widehat{\text{FR}},\widehat{\text{Ab}}}$ given the assumptions $1-7$.

- (Ab, C1) – then the part of the proof violating the ordering is of form:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\overline{\lambda}x.\psi^x \leq \overline{\lambda}x.\phi^x}\,\text{Ab}} \qquad \boxed{\dfrac{\bigtriangledown}{\alpha\ \mathbf{coor}^C\ \gamma \leq \gamma}\,\text{C1}}}{\overline{\lambda}x.\psi^x \leq \gamma}\,\text{TRANS}$$

  However, $\overline{\lambda}x.\phi^x$ can not be equal to $\alpha\ \mathbf{coor}^C\ \gamma$, since $\alpha\ \mathbf{coor}^C\ \gamma$ is not an abstraction term.

- (Ab, D2), (Ab, MON), (Ab, FR) – the proof is similar to the previous case.

Rewritable violations.

Now we prove that a canonic proof of $\alpha_0 \leq \alpha$ can be constructed from the pre-canonic $\widehat{\text{FR}}$-free proof P of $\alpha_0 \leq \alpha$ that has only rewritable violations. The proof is by induction on the size n of P.

Base. $n = 1$. The proof P is an axiom. Thus the proof is already canonic.

Hypothesis. From a pre-canonic proof of size less or equal to n, a canonic proof can be constructed.

Step. Let $\alpha_0 \leq \alpha$ have a pre-canonic proof P of size $n+1$.

If P is a proof of Type 1:

$$\dfrac{R_1 \boxed{\dfrac{\nabla}{\psi_1 \leq \psi_1'}\ \Psi_1'} \quad R_2 \boxed{\dfrac{\nabla}{\psi_2 \leq \psi_2'}\ \Psi_2'} \quad \cdots \quad R_k \boxed{\dfrac{\nabla}{\psi_k \leq \psi_k'}\ \Psi_k'}}{\alpha_0 = \psi_1 \leq \psi_k' = \alpha}\ R$$

for $R \neq$TRANS, then each of the subproofs $\Psi_1, \ldots, \Psi_k$ is a pre-canonic proof of size at most n, and by the induction hypothesis, canonic proofs $\Psi_1, ..., \Psi_k$ can be constructed from them. Then the proof of $\alpha_0 \leq \alpha$ obtained by replacing $\Psi_1', ..., \Psi_k'$ by $\Psi_1, ..., \Psi_k$ resp. is canonic.

Otherwise P is of Type 2:

$$\dfrac{R_1 \boxed{\dfrac{\nabla}{\psi_1 \leq \psi_1'}\ \Psi_1'} \quad R_2 \boxed{\dfrac{\nabla}{\psi_2 \leq \psi_2'}\ \Psi_2'} \quad \cdots \quad R_k \boxed{\dfrac{\nabla}{\psi_k \leq \psi_k'}\ \Psi_k'}}{\alpha_0 = \psi_1 \leq \psi_k' = \alpha}\ \textsc{trans}^*$$

If one of the main rules R_i in Ψ_i' is ax_i, then each of the proofs of $\dfrac{\Psi_1' \quad \ldots \quad \Psi_{i-1}'}{\psi_1 \leq \psi_i}\ \textsc{trans}$ and $\dfrac{\Psi_{i+1}' \quad \ldots \quad \Psi_k'}{\psi_{i+1} \leq \psi_k}\ \textsc{trans}$ is pre-canonic of size at most n. By the induction hypothesis, canonic proofs Ω_1, Ω_2 can be constructed from them and then the proof

$$\dfrac{\Omega_1 \quad \dfrac{\emptyset}{\psi_i \leq \psi_i'}\ ax_i \quad \Omega_2}{\alpha_0 = \psi_1 \leq \psi_k' = \alpha}\ \textsc{trans}$$

is canonic.

Otherwise there are no occurrences of ax_i. By the induction hypothesis, canonic proofs $\Psi_1, ..., \Psi_k$ can be constructed from the proofs $\Psi_1', \ldots, \Psi_k'$. Let P' be the proof obtained from P by replacing $\Psi_1', ..., \Psi_k'$ by $\Psi_1, ..., \Psi_k$ resp. If the string formed by the rules $R_1 R_2 \ldots R_k$ in P' belongs to the regular language $\mathcal{R}$, then P' is canonic. Otherwise there is a (rewritable) violation of the order of the rules. It can be shown that each such violation can be rewritten. We treat only the violations involving $\hat{\text{A}}$b and Ab rules. The proof of [4] can be easily adapted for the rest of the rewritable violations.

- ($\hat{\text{A}}$b, C1) – then w.l.o.g. the part of proof violating the ordering is of the following form:

$$\dfrac{\dfrac{\boxed{\dfrac{\bigtriangledown}{\psi^x \leq (\alpha\ \mathbf{coor}^C\ \gamma)\langle x\rangle}\ R}}{\overline{ab}(\psi^x) \leq \alpha\ \mathbf{coor}^C\ \gamma}\ \hat{\text{Ab}} \qquad \boxed{\dfrac{\bigtriangledown}{\alpha\ \mathbf{coor}^C\ \gamma \leq \gamma}\ \text{C1}}}{\overline{ab}(\psi^x) \leq \gamma}\ \text{TRANS}$$

By lemma B.1 R can be FR, RMOD, $\hat{\text{Ab}}$ or TRANS. By observation B.1, the proof can be rewritten s.t. $R \neq \hat{\text{Ab}}$.

If $R =$ TRANS, then the order statement $\psi^x \leq \gamma\langle x\rangle$ has a $\hat{\text{FR}}$-free $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof Ω' of size n:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\psi^x \leq \psi'}} \quad \dots \quad \boxed{\dfrac{\bigtriangledown}{\phi \leq (\alpha\ \mathbf{coor}^C\ \gamma)\langle x\rangle}} \quad \dfrac{\dfrac{\emptyset}{\alpha\ \mathbf{coor}^C\ \gamma \leq \gamma}\ \text{C1}}{(\alpha\ \mathbf{coor}^C\ \gamma)\langle x\rangle \leq \gamma\langle x\rangle}\ \text{FR}}{\psi^x \leq \gamma\langle x\rangle}\ \text{TRANS}$$

By the induction hypothesis, a canonic proof Ω can be constructed from Ω'. Thus the following $\hat{\text{FR}}$-free proof of size $n+1$ is canonic :

$$\dfrac{\Omega\ \boxed{\dfrac{\bigtriangledown}{\psi^x \leq \gamma\langle x\rangle}}}{\overline{\lambda} x.\psi^x \leq \gamma}\ \hat{\text{Ab}}$$

If $R =$RMOD, the proof is similar to the TRANS case.

Otherwise $R =$FR. Then the proof is of the following form:

$$\dfrac{\dfrac{\dfrac{\boxed{\dfrac{\bigtriangledown}{\delta \leq \alpha\ \mathbf{coor}^C\ \gamma}} \quad x \equiv x}{\delta\langle x\rangle \leq (\alpha\ \mathbf{coor}^C\ \gamma)\langle x\rangle}\ \text{FR}}{\delta \leq \alpha\ \mathbf{coor}^C\ \gamma}\ \hat{\text{Ab}} \qquad \boxed{\dfrac{\bigtriangledown}{\alpha\ \mathbf{coor}^C\ \gamma \leq \gamma}\ \text{C1}}}{\delta \leq \gamma}\ \text{TRANS}$$

The proof can be rewritten as follows:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\delta \leq \alpha\ \mathbf{coor}^C\ \gamma}} \qquad \boxed{\dfrac{\bigtriangledown}{\alpha\ \mathbf{coor}^C\ \gamma \leq \gamma}\ \text{C1}}}{\delta \leq \gamma}\ \text{TRANS}$$

The size of the above proof is at most n, thus by the induction hypothesis, a canonic proof of size at most n can be constructed from it.

- ($\hat{\text{Ab}}$, D2) – the proof is similar to the previous case.
- ($\hat{\text{Ab}}$, RMOD) – then the part of the proof violating the ordering is of form:

$$\dfrac{\dfrac{\Delta_1 \quad \boxed{\dfrac{\nabla}{\phi^x \leq (\psi\langle\gamma\rangle)\langle x\rangle}\, R}}{\overline{ab}(\phi^x) \leq \psi\langle\gamma\rangle}\,\hat{\text{Ab}} \qquad \dfrac{\emptyset}{\psi(\gamma) \leq \gamma}\,\text{RMOD}}{\overline{ab}(\phi^x) \leq \gamma}\,\text{TRANS}$$

The proof is similar to the previous case.

- ($\hat{\text{Ab}}$, MON) – then the part of the proof violating the ordering is of form:

$$\dfrac{\dfrac{\Delta_1 \quad \boxed{\dfrac{\nabla}{\phi^x \leq (\psi\langle\gamma\rangle)\langle x\rangle}\, R}}{\overline{ab}(\phi^x) \leq \psi_{(\tau+\rho)}\langle\gamma\rangle}\,\hat{\text{Ab}} \qquad \dfrac{\Delta_2 \quad \boxed{\dfrac{\nabla}{\gamma \leq \delta}}}{\psi_{(\tau+\rho)}\langle\gamma\rangle \leq \psi_{(\tau+\rho)}\langle\delta\rangle}\,\text{MON}}{\overline{ab}(\phi^x) \leq \psi_{(\tau+\rho)}\langle\delta\rangle}\,\text{TRANS}$$

The proof is similar to the previous case.

- ($\hat{\text{Ab}}$, FR) – the proof is similar to the previous case.
- (Ab,Ab)– the part of the proof that violates the ordering proof is

$$\dfrac{\dfrac{\boxed{\dfrac{\nabla}{\alpha^x \leq \delta^x}}}{\overline{\lambda}x.\alpha^x \leq \overline{\lambda}x.\delta^x}\,\text{Ab} \qquad \dfrac{\boxed{\dfrac{\nabla}{\delta^x \leq \gamma^x}}}{\overline{\lambda}x.\delta^x \leq \overline{\lambda}x.\gamma^x}\,\text{Ab}}{\overline{\lambda}x.\alpha^x \leq \overline{\lambda}x.\gamma^x}\,\text{TRANS}$$

This part of proof can be rewritten:

$$\dfrac{\dfrac{\boxed{\dfrac{\nabla}{\alpha^x \leq \delta^x}} \qquad \boxed{\dfrac{\nabla}{\delta^x \leq \gamma^x}}}{\alpha^x \leq \gamma^x}\,\text{TRANS}}{\overline{\lambda}x.\alpha^x \leq \overline{\lambda}x.\gamma^x}\,\text{Ab}$$

The proof is of size at most n, thus by the induction hypothesis a canonic proof of size at most n can be constructed from it.

- ($\hat{\text{A}}$b, $\hat{\text{A}}$b), (Ab, $\hat{\text{A}}$b), ($\hat{\text{A}}$b, Ab) – the proof is similar to the previous case.

In this way we can remove all the rewritable violations, constructing a canonic proof. Next we prove that any order statement that has a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{A}}\text{b}}$ $\hat{\text{F}}$R-free canonic proof is provable by the algorithm.

Definition B.1 (Mixed Monotonicity) *([4]) Let $\psi_1(\gamma_1), \ldots, \psi_n(\gamma_n)$ be terms. We say that a part of a canonic form:*

$$\psi_1\langle\gamma_1\rangle \leq \psi_2\langle\gamma_2\rangle \leq \ldots \leq \psi_n\langle\gamma_n\rangle$$

has mixed monotonicity iff one the following holds:

(a) $\psi_1, \psi_n \in \boldsymbol{MON}$ *and there exist* i, j, k*:* $1 \leq i \leq j \leq k \leq n$*, s.t.* $\psi_i, \psi_k \in \boldsymbol{MON}\uparrow, \psi_j \in \boldsymbol{MON}\downarrow$ *or* $\psi_i, \psi_k \in \boldsymbol{MON}\downarrow, \psi_j \in \boldsymbol{MON}\uparrow$.

(b) $\psi_1 \notin \boldsymbol{MON}$ *or* $\psi_n \notin \boldsymbol{MON}$ *and there exist* i, j*:* $1 \leq i \leq j \leq n$ *s.t.*$\psi_i \in \boldsymbol{MON}\uparrow, \psi_j \in \boldsymbol{MON}\downarrow$ *or* $\psi_i \in \boldsymbol{MON}\downarrow, \psi_j \in \boldsymbol{MON}\uparrow$.

(c) $\psi_1, \psi_n \notin \boldsymbol{MON}$ *and there exists* i*:* $1 < i < n$*, s.t.*$\psi_i \in \boldsymbol{MON}$.

Definition B.2 (Minimal $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{A}}\text{b}}$ canonic proof) *Let the order statement $\psi \leq \psi'$ have a $\mathcal{L}$-OC$^{\hat{FR},\hat{A}b}$ canonic proof P of size t. P is a* minimal *canonic proof iff there does not exist a $\mathcal{L}$-OC$^{\hat{FR},\hat{A}b}$ canonic proof P' of size less than t.*

Observation B.2 *If an order statement has a canonic $\mathcal{L}$-OC$^{\hat{FR},\hat{A}b}$ proof, then it has a* minimal *canonic $\mathcal{L}$-OC$^{\hat{FR},\hat{A}b}$ proof.*

Lemma B.3 *Let α_0, α be NF terms and let $\alpha_0 \leq \alpha$ be an order statement that has a $\hat{F}R$-free minimal canonic proof P of size n, free of sequences of mixed monotonicity. If for each $< \psi, \psi' > \in$ Goals there does not exist a canonic $\mathcal{L}$-OC$^{\hat{FR},\hat{A}b}$ proof of the order statement $\psi \leq \psi'$ of size less than or equal to n, then* $\boldsymbol{derive}(\alpha_0, \alpha, Goals)$ *returns* $\boldsymbol{true}$.

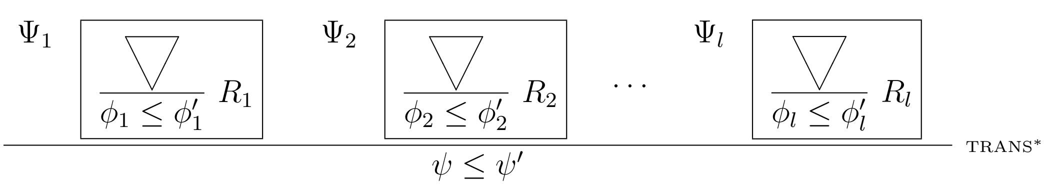
Proof: by induction on the size n of the canonic proof P.

Base: $n = 1$. The proof has the form $\dfrac{\emptyset}{\alpha_0 \leq \alpha}$ R, where R is one of the axioms REFL, NL, RMOD, C1, D1 and $< \alpha_0, \alpha > \notin$ *Goals*.

- REFL. $\alpha_0 = \alpha$. In step 1 of **subderive** the algorithm compares α_0 and α and returns **true**.
- NL. Then $\alpha_0 = \alpha_i$ and $\alpha = \alpha'_i$ for $[\ \alpha_i, \alpha'_i\ ]$ a non-logical axiom. In step 4.1 of **derive** the algorithm calls **subderive**$(\alpha_0, \alpha_i, Goals')$ and **derive**$(\alpha'_i, \alpha, Goals')$ which return **true** according to the previous case.
- RMOD. The proof is $\dfrac{\emptyset}{\alpha_0 = \psi_{(\tau R_\tau)}(\gamma) \leq \gamma = \alpha}$ RMOD . In step 2.1 of **subderive**, the algorithm computes $f_{RMOD,r}((\psi_{(\tau R_\tau)})(\gamma)) = \gamma_{(\tau\tau)}$ and then calls **derive**$(\gamma, \gamma, Goals')$ which returns **true** according to the REFL case.
- C1. The proof is $\dfrac{\emptyset}{\alpha_0 = \psi \ \ \mathbf{coor}^C \ \ \varphi \leq \alpha}$ C1 and $\alpha = \psi$ or $\alpha = \varphi$. In step 2.1 of **subderive**, the algorithm computes $f_{C1,r}(\psi \ \ \mathbf{coor}^C \ \ \varphi) = \alpha$ and then calls **derive**$(\alpha, \alpha, Goals')$ which returns **true** according to the REFL case.
- D1. The proof is $\dfrac{\emptyset}{\alpha_0 \leq \psi \ \ \mathbf{coor}^D \ \ \varphi = \alpha}$ D1 and $\alpha_0 = \psi$ or $\alpha_0 = \varphi$. In step 2.2 of **subderive**, the algorithm computes $f_{D1,l}(\psi \ \ \mathbf{coor}^D \ \ \varphi) = \alpha_0$ and then calls **derive**$(\alpha, \alpha, Goals')$ which returns **true** according to the REFL case.

Hypothesis: Assume that for every order statement $\alpha_0 \leq \alpha$ (for α_0, α NF terms), that has a minimal canonic proof P of size $t \leq n$, if for each $< \psi, \psi' > \in Goals$ there does not exist a canonic $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof of the order statement $\psi \leq \psi'$ of size less than t, then **derive**$(\alpha_0, \alpha, Goals)$ returns **true**.

Step: Let $\alpha_0 \leq \alpha$ be an order statement that has a minimal canonic proof of size $n + 1$. The proof can have one of the two types. We show first that an order statement that has a canonic proof of Type 2 is provable:



where (i) $n \geq 2$, (ii) $\Psi_1, \ldots, \Psi_l$ are Type 1 canonic proofs, (iii) $\psi = \phi_1, \phi'_1 = \phi_2, \phi'_2 = \phi_3, \ldots, \phi'_{l-1} = \phi_l, \phi'_l = \psi'$, (iv) the string formed by the rules $R_1 ... R_l$ belongs to the regular language $\mathcal{R}$.

We distinguish between two cases depending on whether the string $R_1 R_2 \ldots R_l$ includes instances of the non-logical axioms.

If there exist $i_1, \ldots, i_m$ s.t. $1 \leq i_1 \leq i_2 \leq \ldots \leq i_m \leq l$ and for every $1 \leq j \leq l$, $R_{i_j} = a_i$, (that is $\psi_{i_j} \leq \psi'_{i_j}$ is a non-logical axiom) we will look at the leftmost non-logical axiom instance, i.e. $\psi_{i_1} \leq \psi'_{i_1}$. In step 4.1 of **derive**, the algorithm attempts the non-logical axiom [ψ_{i_1}, ψ'_{i_1}] and calls **subderive**($\alpha_0, \psi_{i_1}, Goals'$) and **derive**($\psi'_{i_1}, \alpha, Goals'$). The order statement $\alpha_0 \leq \psi_{i_1}$ has a minimal canonic proof Ω_1 [2]:

$$\frac{\Psi_1 \quad \Psi_2 \quad \ldots \quad \Psi_{i_1}}{\alpha_0 \leq \psi_{i_1}} \ \text{TRANS}^*,$$

where none of the $R_1 \ldots R_{i-1}$ is a non-logical axiom, therefore **derive**($\alpha_0, \psi_{i_1}, Goals$) returns **true** iff **subderive**($\alpha_0, \psi_{i_1}, Goals'$) returns **true** and the two calls are equivalent. Similarly, the order statement $\psi_{i_1+1} \leq \alpha$ has a minimal canonic proof Ω_2:

$$\frac{\Psi_{i_1+1} \quad \Psi_{i_1+2} \quad \ldots \quad \Psi_l}{\psi_{i_1+1} \leq \alpha} \ \text{TRANS}^*$$

Both of the proofs of $\alpha_0 \leq \psi_{i_1}$ and $\psi'_{i_1} \leq \alpha$ are of size $s \leq n$. Also note that $Goals' = Goals \cup \{< \alpha_0, \alpha >\}$ and since the proof P is minimal, there does not exist a smaller $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of $\alpha_0 \leq \alpha$. Therefore, the set $Goals'$ contains only tuples $< \psi, \psi' >$ s.t. the order statement $\psi \leq \psi'$ does *not* have a $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of size $\leq n$. By the induction hypothesis, **derive**($\alpha_0, \psi_{i_1}, Goals'$) and **derive**($\psi'_{i_1}, \alpha, Goals'$) return **true** and so does **derive**($\alpha_0, \alpha, Goals$).

Otherwise, there does not exist such i that $1 \leq i \leq l$ and $R_i = a_i$.

Then, since the proof is $\hat{\text{FR}}$-free, the string formed by the rules $R_1 R_2 \ldots R_l$ belongs to the regular language R-PROD* (STR* | Ab | $\hat{\text{Ab}}$) L-PROD*.

If R_1 is an R-PROD rule, the algorithm can effectively produce subgoals of a smaller size, which are provable by the induction hypothesis. The following cases are possible:

1. R_1 is an axiom, that is $\psi'_1 = f_{ax,r}(\psi_1)$. In step 2.1 of **subderive**, the algorithm calls **derive**($\psi'_1, \alpha, Goals'$). The order statement $\psi'_1 \leq \alpha$ has a minimal canonic proof Ω:

$$\frac{\Psi_2 \quad \ldots \quad \Psi_l}{\psi'_1 \leq \psi_l = \alpha} \ \text{TRANS}^*$$

   of size n at most. $Goals' = Goals \cup \{< \alpha_0, \alpha >\}$. Since the proof P is minimal, there does not exist a smaller $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of $\alpha_0 \leq \alpha$. Therefore, the set $Goals'$

[2] The proof Ω_1 is minimal because it is a part of a minimal $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof.

contains only tuples $< \psi, \psi' >$ s.t. the order statement $\psi \leq \psi'$ does *not* have a $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof of size $\leq n$. By the induction hypothesis, **derive**$(\psi_1', \alpha, Goals')$ returns **true** and so does **derive**$(\alpha_0, \alpha, Goals)$.

2. If $R_1 =$ D2, the proof Ψ_1 of $\psi_1 \leq \psi_1'$ is of form:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\psi_{1,1} \leq \psi_1'}\,P_1} \qquad \boxed{\dfrac{\bigtriangledown}{\psi_{1,2} \leq \psi_1'}\,P_2}}{\psi_1 = \psi_{1,1}\ \mathbf{coor}^D\ \psi_{1,2} \leq \psi_1'}\,\text{D2}$$

In step 3.1 of **subderive**, the algorithm calls **derive**$(\psi_{1,1}, \alpha, Goals')$ and **derive**$(\psi_{1,2}, \alpha, Goals')$. Also, $\vdash_{\mathcal{L}-OC} \psi_{1,i} \leq \alpha$:

$$\dfrac{\boxed{\dfrac{\bigtriangledown}{\psi_{1,i} \leq \psi_1'}\,P_i} \qquad \Psi_2 \quad \ldots \Psi_l}{\psi_{1,i} \leq \alpha}\,\text{TRANS}^* \quad \text{, where } i \in \{1,2\}$$

By lemma B.2 and observation B.2, both of the order statements $\psi_{1,1} \leq \alpha$ and $\psi_{1,2} \leq \alpha$ have minimal canonic proofs of size n at most. $Goals' = Goals \cup \{< \alpha_0, \alpha >\}$ and since the proof P is minimal, there does not exist a canonic $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof of $\alpha_0 \leq \alpha$ of size $\leq n+1$. Therefore, the set $Goals'$ contains tuples $< \psi, \psi' >$ s.t. the order statement $\psi \leq \psi'$ does *not* have a $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof of size $\leq n$. By the induction hypothesis, **derive**$(\psi_{1,1}, \alpha, Goals')$ and **derive**$(\psi_{1,2}, \alpha, Goals')$ return **true** and so does **derive**$(\alpha_0, \alpha, Goals)$.

If R_l is an L-PROD rule, the proof is symmetric.

Otherwise the string formed by the rules $R_1 R_2 \ldots R_l$ belongs to the regular language (STR* | Ab | $\hat{\text{A}}$b). Since $l > 2$, it can only belong to the regular language STR*, then the proof is similar to the proof of [4] for the matching case.

Next, we show that an order statement that has a canonic proof of Type 1 is provable:

$$\dfrac{\Psi_1\ \boxed{\dfrac{\bigtriangledown}{\phi_1 \leq \phi_1'}\,R_1} \quad \cdots \quad \Psi_l\ \boxed{\dfrac{\bigtriangledown}{\phi_l \leq \phi_l'}\,R_l}}{\psi \leq \psi'}\,R$$

where (i) $R \neq$TRANS, (ii) $l \in \{1,2\}$, and (iii) $\Psi_1, \ldots, \Psi_l$ are canonic form proofs. One of the following cases holds:

1. $R = \mathrm{Ab}$. Then the proof of the order statement $\alpha_0 \leq \alpha$ is of the form:

$$\frac{\boxed{\dfrac{\nabla}{\mu^x \leq \mu'^x}}}{\alpha_0 = \overline{\lambda} x.\mu^x \leq \overline{\lambda} x.\mu'^x = \alpha}\ \mathrm{Ab}$$

   The order statement $\mu^x \leq \mu'^x$ has a minimal canonic proof of size at most n. In step 6 of **subderive**, the algorithm calls **derive**$(\mu^x, \mu'^x, \mathit{Goals}')$. $\mathit{Goals}' = \mathit{Goals} \cup \{< \alpha_0, \alpha >\}$ and since the proof P is minimal, there does not exist a canonic $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof of $\alpha_0 \leq \alpha$ of size $\leq n+1$. Therefore, the set Goals' contains tuples $< \psi, \psi' >$ s.t. the order statement $\psi \leq \psi'$ does *not* have a $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof of size $\leq n$. By the induction hypothesis, **derive**$(\mu^x, \mu'^x, \mathit{Goals}')$ returns **true** and so does **derive**$(\alpha_0, \alpha, \mathit{Goals})$.

2. $R = \widehat{\mathrm{Ab}}$. Then w.l.o.g. the proof has one of the following forms:

   - $$\frac{\boxed{\dfrac{\nabla}{\mu^x \leq \mu'\langle x\rangle}}}{\alpha_0 = \overline{\lambda} x.\mu^x \leq \mu' = \alpha}\ \widehat{\mathrm{Ab}}$$

     The order statement $\mu^x \leq \mu'\langle x\rangle$ has a minimal canonic proof of size at most n. In step 7 of **subderive** the algorithm calls **derive**$(\mu^x, \mu'\langle x\rangle, \mathit{Goals}')$. $\mathit{Goals}' = \mathit{Goals} \cup \{< \alpha_0, \alpha >\}$ and since the proof P is minimal, there does not exist a canonic $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof of $\alpha_0 \leq \alpha$ of size $\leq n+1$. Therefore, the set Goals' contains tuples $< \psi, \psi' >$ s.t. the order statement $\psi \leq \psi'$ does *not* have a $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof of size $\leq n$. By the induction hypothesis, **derive**$(\mu^x, \mu'\langle x\rangle, \mathit{Goals}')$ returns **true** and so does **derive**$(\alpha_0, \alpha, \mathit{Goals})$.

   - $$\frac{\boxed{\dfrac{\nabla}{\mu\langle x\rangle \leq \mu'^x}}}{\alpha_0 = \mu \leq \overline{\lambda} x.\mu'^x = \alpha}\ \widehat{\mathrm{Ab}}$$

     The proof is similar to the previous case.

- $$\dfrac{\boxed{\dfrac{\nabla}{\mu\langle x\rangle \leq \mu'\langle x\rangle}\ R}}{\alpha_0 = \mu \leq \mu' = \alpha}\ \hat{\mathrm{A}}\mathrm{b}$$

  Let us show that this case is impossible. By observation B.1 and lemma B.1, the proof can be rewritten s.t. R is either FR or TRANS, the premises of which are conclusions of FR rules. From this follows that the order statement $\mu \leq \mu'$ has a shorter $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof, in contradiction with the minimality of the given proof.

3. The proof for the rest of the rules is similar to one of the previous cases.

Corollary B.1 (Provability of order statements having a $\hat{\mathrm{FR}}$-free canonic $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof) *Let α_0, α be NF terms and let $\alpha_0 \leq \alpha$ be an order statement that has a $\hat{FR}$-free canonic proof P, which is free of sequences of mixed monotonicity. Then* ***derive***$(\alpha_0, \alpha, \emptyset)$ *returns* ***true****.*

The corollary follows directly from the above lemma.

Bibliography

[1] R. Bernardi, *Reasoning with Polarity in Categorial Type Grammar*, Ph.D. thesis, Utrecht University, 2002.

[2] W. Buszkowski, *The logic of types*, Initiatives in Logic, J.T. Srzednicki, editor, 1987.

[3] D. Dowty, *The role of Negative Polarity and Concord Marking in Natural Language reasoning*, Semantics and Linguistic Theory **4** (1994), 114–144.

[4] Y. Fyodorov, Y. Winter, and N. Francez, *Order-Based Inference in Natural Logic*, Logic Journal of the IGPL (2003).

[5] L.T.F Gamut, *Logic, Language and Meaning*, vol. 2, The University of Chicago Press, Chicago, 1991.

[6] A. Lifshits-Zamansky, Y. Winter, and N. Francez, *Order-Based Inference using the Lambek calculus*, Proceedings of the 7-th conference on Formal Grammar, University of Trento (2002).

[7] D. McAllester and R. Givan, *Natural Language syntax and First Order Inference* , Artificial Intelligence **56** (1992), 1–20.

[8] M. Moortgat, *Categorial Type Logics, Handbook for Logic and Language*, Johan van Benthem and Alice ter Muelen eds., Elsevier/MIT Press, 1997.

[9] V. Sánchez, *Studies on Natural Logic and Categorial Grammar*, Ph.D. thesis, University of Amsterdam, 1991.

[10] P. Suppes, *Logical Inference in English: a preliminary analysis*, Studia Logica **38** (1979), 375–391.

[11] J. van Benthem, *Meaning: interpretation and inference*, Synthese **73** (1987), 451–470.

[12] H. Wansing, *The logic of information structures*, Springer Lecture Notes, Springer-Verlag, Berlin, 1993.

[13] A. Zamansky, Y. Winter, and N. Francez, *A 'Natural Logic' Inference System using the Lambek calculus*, Proceedings of the 4-th workshop on Inference in Computational Semantics, LORIA, Nancy (2003).