A 'Natural Logic' inference system using normalization

Anna Zamansky

Technion, I.I.T.
lifshits@cs.technion.ac.il

Nissim Francez

Technion, I.I.T
francez@cs.technion.ac.il

Yoad Winter

Technion, I.I.T
winter@cs.technion.ac.il

Abstract

This paper further develops a 'Natural Logic' inference system which is based on the Lambek calculus and works directly on the Curry-Howard counterparts of the natural language syntactic representations, with no intermediate translation to logical formulae. We show how adding normalization axioms allows the system to prove additional kinds of inferences and propose a proof search algorithm.

1 Introduction

The goal of this paper is to further extend the Lambek calculus ($\mathcal{L}$) based 'Natural Logic' system introduced in [7] for deriving inferences (with sentences that involve extraction, e.g. relative clauses, pied piping, etc.). This system transcends the system of [2], which is based on the applicative Categorial Grammar (AB).

The $\mathcal{L}$-based Order Calculus ($\mathcal{L}$-OC) manipulates *order statements* between $\mathcal{L}$ *proof terms* that represent $\mathcal{L}$-derivation trees of natural language expressions via the Curry-Howard (CH) correspondence. The order statements reflect semantic order relations between elements of partially ordered domains. Hence we can derive inferences directly from $\mathcal{L}$ proof terms, without reducing them to inferences in intermediate logical levels of representation, such as first order logic. We view $\mathcal{L}$-OC as a step towards a general system that supports various kinds of inferences in natural language, eventually based on the full Multi-Modal Type-logical paradigm [3].

While [7] focuses on the treatment of *abstraction*, which allows to derive inferences with sentences that involve extraction (relative clauses, pied piping etc.), the main focus of this paper is on adding normalization axioms to $\mathcal{L}$-OC . We use them as a remedy to a complication created in $\mathcal{L}$-OC – the emergence of proof terms that are not in normal form. This poses a problem in two main respects. First – some inferences involving non-normalized

proof terms cannot be derived by the system of [7]. Secondly, finding non-normalized $\mathcal{L}$ derivations is ineffective, due to their lack of the sub-formula property. We overcome these problems by augmenting $\mathcal{L}$-OC with normalization axioms, based on $\beta\eta$-reduction of terms. The use of the normalization axioms is demonstrated by deriving additional inferences that could not be derived without such axioms, for example: (1) John does and Mary doesn't move entails Mary doesn't walk, given the assumption walk $\leq$ move and (2) Some tall nice and smart boy walked entails Some boy walked.
We also propose a proof search algorithm that is based on the proposal by [2].

2 Preliminaries

Meanings of natural language expressions are associated with (semantic) *types*. The set of types and its subset of *partially ordered (PO) types* are defined standardly. The types e (for entities) and t (for truth values) are among the primitive types. Each primitive type τ is associated with a non-empty domain D_τ. The domain $D_{(\tau\sigma)}$ associated with a non-primitive type $(\tau\sigma)$ is $(D_\tau \rightarrow D_\sigma)$ (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 *pointwise partial order relation* for non-primitive PO types is defined as follows.

Definition 2.1 ***(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 define the sets of *decorated types* and *PO decorated types*, by which we replace the sets of standard types and PO types. $Feat = \{+, -, R, C, D\}$ is a set of semantic features, which abstract the semantic properties of denotations of natural language expressions as follows: '+'/'-' marks upward/downward monotonicity, 'R' marks restrictivity and 'C'/'D' mark conjunction/disjunction.

Definition 2.2 ***(Decorated types and decorated PO types)*** *Let T^0, T^0_{po} be sets of primitive types and primitive PO types resp. The sets of decorated types and PO decorated types are the smallest sets T_{dec}, T^{po}_{dec} so that:*

- *$T^0 \subseteq T_{dec}$, $T^0_{po} \subseteq T^{po}_{dec}$*
- *if $\tau \in T_{dec}$, $\sigma \in T_{dec}$ and $\rho \in T^{po}_{dec}$ then $(\tau^F\sigma) \in T_{dec}, (\tau^F\rho) \in T^{po}_{dec}$, where $F \subseteq Feat$ and the following conditions hold: (i) if $F \neq \emptyset$, then $\tau, \sigma \in T^{po}_{dec}$, (ii) if $R \in F$ then $\tau = \sigma$, and (iii) if C or $D \in F$ then (a) if[1] $\tau = (\tau_1{}^{F'}\tau_2)$ then $F' = \emptyset$ and (b) $\sigma = (\tau^\emptyset\tau)$.*

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 2.3 ***(Domains of decorated types)*** *Let $F \subseteq Feat$. For each non-primitive type $(\tau^F\sigma) \in T_{dec}\backslash T^0$, let $D^F_{(\tau\sigma)}$ 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, $D^+_{(\tau\sigma)}$ is the domain of upward monotone functions from D_τ to D_σ. For $F = \emptyset$, $D_{(\tau^F\sigma)} = D_{(\tau\sigma)}$. For σ a decorated PO type, $D^F_{(\tau\sigma)}$ inherits its partial ordering from $D_{(\tau\sigma)}$, where $\leq_{(\tau^F\sigma)}$ is the restriction of $\leq_{(\tau\sigma)}$ to $D^F_{(\tau\sigma)} \subseteq D_{(\tau\sigma)}$. Note that *all* decorated functional types are marked, some with $F = \emptyset$.
The set of syntactic categories in $\mathcal{L}$ is standardly defined as in the categorial grammar of

[1] This condition guarantees that an expression of a type marked by 'C' or 'D' is treated as denoting a binary function and consequently all its markings are specified in the same set of semantic features.

[3] in the slash format 'result on left' of [5]. Meta variables A, B range over categories. Each syntactic category in $\mathcal{L}$ is assigned a directed lambda term φ_τ that encodes the $\mathcal{L}$-derivation of this category using the Curry-Howard isomorphism (see [6] for similar use of directed lambda terms). Henceforth, terms encoding $\mathcal{L}$-derivations are referred to as *peripherally-linear (PL) terms.* Variables in PL terms are partitioned into two disjoint sets: **LexVAR** and **VAR**. In a proof term, members of **LexVAR** originate from lexical entries and cannot be discharged by introduction rules, while members of **VAR** correspond to undischarged assumptions in the derivation. For a term ψ, the set of free variables $Free(\psi)$ and the ordered sequence of free variables $\overline{Free}(\psi)$ are defined standardly. We use the following notation:

- A variable $w_\tau \in \mathbf{LexVAR}$ is assigned to a lexical item $\mathbf{w}$ of (decorated) type τ.
- A variable $x_\tau \in \mathbf{VAR}$ is assigned to a dischargeable assumption x of type τ.
- /-application (\-application) between terms $\varphi_{(\tau^*\sigma)}, \psi_\tau$ is denoted by $(\varphi(\psi))_\sigma$ $([\psi]\varphi)_\sigma$. We use the pattern $\varphi\langle\psi\rangle$ for $\varphi(\psi)$ or $[\psi]\varphi$.
- 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 $\varphi_\tau^{x_\sigma}$.
- /-abstraction (\-abstraction) in a term $\varphi_\tau^{\overrightarrow{x_\rho}}$ is denoted by $\overrightarrow{\lambda} x_\rho.\varphi^{\overrightarrow{x_\rho}}$ $(\overleftarrow{\lambda} x_\rho.\varphi^{\overleftarrow{x_\rho}})$, both abbreviated to $\overline{\lambda} x.\varphi^{x_\rho}$.

Formally equivalent types are types that are equal up to their decoration, denoted by $\tau \equiv_f \sigma$. Note that if $\tau \equiv_f \tau'$, then $\leq_\tau$ is compatible with $\leq_{\tau'}$. We now define $\mathcal{L}$, the only difference of which from the standard Lambek calculus is that proof terms have decorated types.

Definition 2.4 *($\mathcal{L}$) Let $\Gamma, \Gamma_1, \Gamma_2$ range over finite non-empty sequences of pairs $A_i : \psi_{i_{\tau_i}}$, where A_i is a syntactic category and ψ_i a term of a (decorated) type τ_i. Let $\tau, \tau_1, \tau_2, ...$ range over decorated types. The notation $\Gamma \triangleright A : \psi_\tau$ means that the sequence Γ is $\mathcal{L}$-reducible to $A : \psi_\tau$. The rules of $\mathcal{L}$ are as follows :*

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

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

Elimination rules:

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

Introduction rules:

$$(/I)\frac{\Gamma_1,\ B : x_{\tau_1} \triangleright A : \psi_{\tau_2}^{\overrightarrow{x_{\tau_1}}}}{\Gamma_1 \triangleright (A/B) : (\overrightarrow{\lambda} x_{\tau_1}.\psi_{\tau_2}^{\overrightarrow{x_{\tau_1}}})_{(\tau_1\tau_2)}} \quad (\backslash I)\frac{B : x_{\tau_1},\ \Gamma_1 \triangleright A : \psi_{\tau_2}^{\overleftarrow{x_{\tau_1}}}}{\Gamma_1 \triangleright (A\backslash B) : (\overleftarrow{\lambda} x_{\tau_1}.\psi_{\tau_2}^{\overleftarrow{x_{\tau_1}}})_{(\tau_1\tau_2)}} \ for\ \Gamma_1\ not\ empty,\ x_{\tau_1} \in \mathbf{VAR}$$

3 $\mathcal{L}$-OC and deriving inferences

$\mathcal{L}$-OC is a calculus for proving order statements of the form $\varphi_\tau \leq_{\tau^\circ} \psi_{\tau'}$ between $\mathcal{L}$ proof terms of formally equivalent (PO) types ($\tau \equiv_f \tau'$). Note that lexical markings are abstracted from standard denotations. However, there is no appeal to denotations when deriving $\mathcal{L}$-OC order-statements. Doing so is in contrast with Natural Logic of [4] and [1].

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 $\vdash_{\mathcal{L}-OC} \varphi_\tau \equiv_{\tau^\circ} \psi_{\tau'}$. When types are clear from context or are implicitly universally quantified over, $\leq_{\tau^\circ}$ is abbreviated to $\leq$.

Definition 3.1 ***(The set of subterms)*** *For a term ψ, the set of subterms of ψ ($ST(\psi)$) is defined recursively: for $\psi \in \mathbf{VAR} \cup \mathbf{LexVAR}$ $ST(\psi) = \{\psi\}$, for $\psi = \varphi\langle\phi\rangle$ $ST(\psi) =$*

$\{\psi\} \cup ST(\varphi) \cup ST(\phi)$*, for* $\psi = \overline{\lambda} x.\phi$ $ST(\psi) = \{\phi\} \cup ST(\phi)$.

The term $\psi_\rho[x_\tau/\delta_\tau]$ (s.t. $x_\tau \in \mathbf{VAR}$ and no variables bound in ψ_ρ are in $Free(\delta_\tau)$) is obtained by substituting $x_\tau \in ST(\psi_\rho)$ by the term δ_τ.

Definition 3.2 ***($\mathcal{L}$-OC :)***

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

$$\textbf{(Refl)}\frac{\emptyset}{\psi_\tau \leq_{\tau^\circ} \psi_{\tau'}} \quad \textbf{(Trans)}\frac{\psi_\tau \leq_{\tau^\circ} \phi_{\tau'} \quad \phi_{\tau'} \leq_{\tau^\circ} \varphi_{\hat{\tau}}}{\psi_\tau \leq_{\tau^\circ} \varphi_{\hat{\tau}}}$$

$$\textbf{(Mon+)}\frac{\psi_\tau \ \leq_{\tau^\circ} \ \phi_{\tau'}}{\varphi_{(\hat{\tau}+\rho)}\langle\psi_\tau\rangle \ \leq_{\rho^\circ} \ \varphi_{(\hat{\tau}+\rho)} \ \langle\phi_{\tau'}\rangle} \quad \textbf{(Mon-)}\frac{\phi_{\tau'} \ \leq_{\tau^\circ} \ \psi_\tau}{\varphi_{(\hat{\tau}-\rho)}\langle\psi_\tau\rangle \ \leq_{\rho^\circ} \ \varphi_{(\hat{\tau}-\rho)} \ \langle\phi_{\tau'}\rangle}$$

$$\textbf{(FR)}\frac{\phi_{(\tau^*\rho)} \ \leq_{(\tau\rho)^\circ} \ \psi_{(\tau'^*\rho')} \quad \varphi_{\hat{\tau}} \ \equiv_{\tau^\circ} \ \varphi'_{\tilde{\tau}}}{\phi_{(\tau^*\rho)} \ \langle\varphi_{\hat{\tau}}\rangle \ \leq_{\rho^\circ} \ \psi_{(\tau'^*\rho')} \ \langle\varphi'_{\tilde{\tau}}\rangle} \quad \textbf{(Rmod)}\frac{\emptyset}{\psi_{(\tau^R\tau')} \ \langle\phi_{\hat{\tau}}\rangle \ \leq_{\tau^\circ} \ \phi_{\hat{\tau}}}$$

$$\textbf{(Ab)}\frac{\psi_\rho^{x_\tau} \ \leq_{\rho^\circ} \ \phi_{\rho'}^{x_{\tau'}}}{\overline{\lambda} x_\tau.{\psi_\rho}^{x_\tau} \ \leq_{(\tau\rho)^\circ} \ \overline{\lambda} x_{\tau'}.{\phi_\rho}^{x_{\tau'}}}$$

$$\overline{\lambda} x.\psi^x, \overline{\lambda} x.\phi^x \ \textit{contain at least one free variable from } \mathbf{VAR} \cup \mathbf{LexVAR}$$

$$\textbf{(C}_1\textbf{)}\frac{\emptyset}{([\phi_{\tau'}]\varphi_{(\tau^C(\tau\tau))}) \ \ (\psi_{\hat{\tau}}) \ \leq_{\tau^\circ} \ \Psi} \qquad \textbf{(C}_2\textbf{)}\frac{\alpha_{\tilde{\tau}} \ \leq_{\tau^\circ} \ \psi_{\tau'} \quad \alpha_{\tilde{\tau}} \ \leq_{\tau^\circ} \ \phi_{\hat{\tau}}}{\alpha_{\tilde{\tau}} \ \leq_{\tau^\circ} \ ([\phi_{\hat{\tau}}]\varphi_{(\tau^C(\tau\tau))}) \ (\psi_{\tau'})}$$

$$\Psi = \psi_{\hat{\tau}} \ \textit{or} \ \Psi = \phi_{\tau'}, \ \phi, \ \psi, \ \alpha \ \textit{do not contain free variables from } \mathbf{VAR}$$

$$\textbf{(D}_1\textbf{)}\frac{\emptyset}{\Psi \ \leq_{\tau^\circ} \ ([\phi_{\tau'}]\varphi_{(\tau^D(\tau\tau))}) \ (\psi_{\hat{\tau}})} \qquad \textbf{(D}_2\textbf{)}\frac{\psi_{\tau'} \ \leq_{\tau^\circ} \ \alpha_{\hat{\tau}} \quad \phi_{\tilde{\tau}} \ \leq_{\tau^\circ} \ \alpha_{\hat{\tau}}}{([\phi_{\tilde{\tau}}]\varphi_{(\tau^D(\tau\tau))}) \ (\psi_\tau) \ \leq_{\tau^\circ} \ \alpha_{\hat{\tau}}}$$

$$\Psi = \psi_{\tau'} \ \textit{or} \ \Psi = \phi_{\tilde{\tau}}, \ \phi, \ \psi, \ \alpha \ \textit{do not contain free variables from } \mathbf{VAR}$$

Normalization axioms :

$$(\beta)\frac{\emptyset}{(\phi_\tau^{y_\rho}[y_\rho/\varphi_{\rho'}])_\tau \equiv_{\tau^\circ} (\overline{\lambda} y_\rho.\phi_\tau^{y_\rho})_{(\rho\tau)}\langle\varphi_{\rho'}\rangle} \quad (\eta)\frac{\emptyset}{\psi_{(\tau^*\rho)} \equiv_{(\tau\rho)^\circ} (\overline{\lambda} x_\tau.\psi_{(\tau^*\rho)}\langle x_\tau\rangle)_{(\tau\rho)}} \ x_\tau \in \mathbf{VAR}, x_\tau \notin ST(\psi)$$

Definition 3.3 ***(Size of $\mathcal{L}$-OC proof)*** *Then 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

$$\frac{\Psi_1 \ \boxed{\begin{array}{c}\bigtriangledown \\ \psi_1 \leq \psi'_1\end{array}} \quad \Psi_2 \ \boxed{\begin{array}{c}\bigtriangledown \\ \psi_2 \leq \psi'_2\end{array}} \quad \ldots \quad \Psi_m \ \boxed{\begin{array}{c}\bigtriangledown \\ \psi_m \leq \psi'_m\end{array}}}{\alpha_0 \leq \alpha} \ R$$

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

Let us now illustrate how $\mathcal{L}$-OC as defined above can be used for deriving inferences in natural language. We introduce a lexicon (for a toy fragment of English) and some additional non-logical axioms.

- *The lexicon:*

  In the lexicon we use the set of primitive types $\{e, t\}$. A fragment of the lexicon (with decorated types) is given below:

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

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

- *Natural Logic inferences*

  In order to derive an entailment of S_2 from S_1 (S_1, S_2 are NL indicative sentences), we prove an $\mathcal{L}$-OC order statement $\psi_t^{S_1} \le \psi_t^{S_2}$ where $\psi_t^{S_1}, \psi_t^{S_2}$ are proof terms representing $\mathcal{L}$-derivations of S_1, S_2 resp. In general, we represent the Natural Logic inferences in $\mathcal{L}$-OC as follows:

  Definition 3.4 *($\vdash_{NatLog}$) Let $S, S_1, ..., S_n$ be NL sentences. Let $\alpha_t^S, \alpha_t^{S_1}, ..., \alpha_t^{S_n}$ be the proof terms representing $\mathcal{L}$-derivation trees of $S, S_1, ..., S_n$ resp. Then $S_1, ..., S_n \vdash_{NatLog} S$ iff $\vdash_{\mathcal{L}-OC} \mathsf{w}_\mathsf{t}^\mathsf{T} \le \alpha_t^{S_1}, ..., \vdash_{\mathcal{L}-OC} \mathsf{w}_\mathsf{t}^\mathsf{T} \le \alpha_t^{S_n}$ implies $\vdash_{\mathcal{L}-OC} \mathsf{w}_\mathsf{t}^\mathsf{T} \le \alpha_t^S$.*

- *Non-logical axioms*

  We postulate the following non-logical axioms of $\mathcal{L}$-OC . For example, (a_5) reflects the fact that a $\mathsf{creative\ intelligent\ x}$ is a $\mathsf{smart\ x}$.

$$\frac{\emptyset}{walked \le moved}\,(a_1) \quad \frac{\emptyset}{walk \le move}\,(a_2)$$

$$\frac{\emptyset}{kissed \le touched}\,(a_3) \frac{\emptyset}{student \le person}\,(a_4) \quad \frac{\emptyset}{\overrightarrow{\lambda} x_e.creative_{(et)}(intelligent_{(et)}(x_e)) \le smart_{(et)}}\,(a_5)$$

4 Examples of inferences using normalization

In fig. 1 and 2 we show examples of inferences that could not be proven in $\mathcal{L}$-OC without the normalization axioms (for reasons to be further discussed in section 5). Instances of the Reflexivity rule are omitted.

In fig. 1 we see the abstraction term $\overrightarrow{\lambda} x.tall_{(et)^R(et)}(nice_{(et)^R(et)}(x_{(et)}))$, which is a composition of terms denoting restrictive functions. Consequently, its denotation is a restrictive function, but its type is not marked with 'R'. Thus we cannot derive the order statement $\overrightarrow{\lambda} x.tall_{(et)^R(et)}(nice_{(et)^R(et)}(x_{(et)}))(boy) \le boy$ directly and use η-normalization. In fig. 2 β-normalization is used to normalize the non-NF proof term $\overrightarrow{\lambda} x.creative(intelligent(x))(boy)$ (see section 5 for details).

5 Why is normalization needed?

We now turn to the use of the normalization axioms. Without these axioms, 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

$$\dfrac{\emptyset}{[\overrightarrow{\lambda} x_e.(tall_{((et)R(et))}(nice_{((et)R(et))}(x_{(et)}))]and_{((et)C((et)(et)))}(smart_{(et)}) \leq \overrightarrow{\lambda} x_e.tall_{((et)R(et))}(nice_{((et)R(et))}(x_{(et)}))}\ (C_1)$$

$$\dfrac{\dfrac{\emptyset}{tall_{(et)R(et)}(nice(x)) \leq nice(x)}\ (Rmod)}{\overrightarrow{\lambda} x.tall(nice(x)) \leq \overrightarrow{\lambda} x.nice(x)}\ (Ab) \qquad \dfrac{\emptyset}{\overrightarrow{\lambda} x.nice(x) \equiv nice}\ (\eta)$$

$$\dfrac{\dfrac{\dfrac{\cdots}{[\overrightarrow{\lambda} x.tall(nice(x))]and(smart) \leq nice}\ (Trans^*)}{([\overrightarrow{\lambda} x.tall(nice(x))]and(smart))(boy_{(et)}) \leq nice(boy_{(et)})}\ (FR) \qquad \dfrac{\emptyset}{nice_{(et)R(et)}(boy_{(et)}) \leq boy_{(et)}}\ (Rmod)}{([(\overrightarrow{\lambda} x.tall(nice(x))]and(smart))(boy)) \leq boy}\ (Trans)$$

$$\dfrac{\dfrac{([(\overrightarrow{\lambda} x.tall(nice(x))]and(smart))(boy)) \leq boy}{some_{((et)+((et)+t))}(([\overrightarrow{\lambda} x.tall(nice(x))]and(smart))(boy)) \leq some_{((et)+((et)+t))}(boy)}\ (MON)}{(some([\overrightarrow{\lambda} x.tall(nice(x))]and(smart))(boy))(walked_{(et)}) \leq (some(boy))(walked_{(et)})}\ (FR)$$

Fig. 1. Some tall nice and smart boy walked $\vdash_{NatLog}$ Some boy walked

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

$$\dfrac{\dfrac{\dfrac{\cdots}{creative(intelligent(boy)) \leq smart(boy)}\ (Trans)}{Some_{((et)+((et)+t))}(creative(intelligent(boy))) \leq Some_{((et)+((et)+t))}(smart(boy))}\ (Mon)}{Some(creative(intelligent(boy)))(smiled) \leq Some(smart(boy))(smiled)}\ (FR)$$

Fig. 2. Some creative intelligent boy smiled $\vdash_{NatLog}$ Some smart boy smiled, using the non-logical axiom ($a5$).

examples (where the non-NF terms are emphasized in boldface).

$$\dfrac{\dfrac{\emptyset}{[\overrightarrow{\lambda} x_{(et)}.John_{((et)+t)}(does_{((et)+(et))}(x_{(et)}))]and_{((et)t)C(((et)t)((et)t))}(\overrightarrow{\lambda} y_{(et)}.Mary(doesn't_{((et)-(et))}(y))) \leq \overrightarrow{\lambda} y.Mary(doesn't(y_{(et)}))}\ (C_1)}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(walk_{(et)}) \leq \overrightarrow{\lambda}\mathbf{y.Mary(doesn't(y))(walk_{(et)})}}\ (FR)$$

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

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

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

$$\dfrac{\dfrac{\emptyset}{happy_{(et)R(et)}(tall_{(et)R(et)}(x_{(et)})) \leq tall_{(et)R(et)}(x_{(et)})}\ (Rmod)}{\overrightarrow{\lambda} x.happy_{(et)R(et)}(tall_{(et)R(et)}(x_{(et)})) \leq \overrightarrow{\lambda}\mathbf{x.tall_{(et)R(et)}(x_{(et)})}}\ (Ab)$$

The term $\overrightarrow{\lambda}\mathbf{x.tall(x)}$ is not in NF and it η-reduces to $tall$.
There are two main reasons why non-NF terms in $\mathcal{L}$-OC proofs pose a problem. We now briefly describe each of the problems and explain how they are solved by the normalization axioms. The first problem is *abstraction terms* with unmarked semantic types. Basing

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

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

Fig. 3. Deriving $S_1, ..., S_n \vdash_{NatLog} S$ in the system

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. Furthermore, 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 specific 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].

Efficiency is another problematic aspect of non-NF terms in $\mathcal{L}$-OC . Recall the general structure of our system (see fig. 3). 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. Again we are led to the need to find a way to exchange the non-NF terms representing $\mathcal{L}$ derivations of the goal sentences by their normal form equivalents.

To demonstrate the effect of normalization, first we define $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ – a modification of $\mathcal{L}$-OC , where $\beta-$ and $\eta-$normalization is applied only to the results of FR and Ab rules resp. $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$ is similar to $\mathcal{L}$-OC , except that (β) and (η) are not explicitly among its axioms. Instead additional inference rules – $\hat{\text{FR}}$ and $\hat{\text{Ab}}$ implicitly encapsulate $\beta-$ and

[2] In [7] an alternative solution of the mentioned problem, using Dynamic marking of abstraction terms is presented.

$\eta-$normalization:

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

ψ *or* ϕ *is an abstraction term,* γ *or* δ *is of form* $\gamma\langle x\rangle$.

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

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

Definition 5.1 ***(RM (rightmost) subterms)*** *Let* ψ *be a term and* $\alpha \in \boldsymbol{LexVAR} \cup \boldsymbol{VAR}$ *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 5.2 ***(Anti-substitution)*** *For terms* ψ_τ, φ_ρ *s.t (i)* $\varphi_\rho \in ST(\psi_\tau)$ *and (ii) no variable* $z \in FV(\varphi_\rho) \cap \boldsymbol{VAR}$ *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* $x_\rho \in \boldsymbol{VAR}$ *s.t.* $x_\rho \notin ST(\psi_\tau)$.

Note that for any[3] $\delta \in RM(\psi)$, $\overline{norm}((\overrightarrow{\lambda} x.\psi \ll x/\delta \gg)(\delta)) = \psi$.
We now prove the following statements for $\mathcal{L}$-OC$^{\hat{\text{FR}},\hat{\text{Ab}}}$: (i) $\vdash_{\mathcal{L}-OC} \alpha' \le \gamma' \ \Rightarrow \vdash_{\mathcal{L}-OC^{\hat{\text{FR}},\hat{\text{Ab}}}} \alpha \le \gamma$ for α, γ the NF of α', γ' resp., (ii) $\vdash_{\mathcal{L}-OC^{\hat{\text{FR}},\hat{\text{Ab}}}} \alpha \le \gamma \ \Rightarrow \vdash_{\mathcal{L}-OC} \alpha \le \gamma$.

Lemma 5.3 *Let the order statement* $\psi' \le \phi'$ *have a* $\mathcal{L}$*-OC proof P. Then there exists a* $\mathcal{L}$*-OC*$^{\hat{FR},\hat{Ab}}$ *proof P' of* $\psi \le \phi$ *s.t.* ψ, ϕ *are the NF of* ψ', ϕ' *resp. and P' contains NF terms only.*

Proof: By induction on the size s of P.
Base: s=1. Then P has one of the following forms:

- P= $\boxed{\dfrac{\emptyset}{\psi' = \alpha' \le \alpha' = \phi'}\ (REFL)}$. Then P' = $\boxed{\dfrac{\emptyset}{\alpha \le \alpha}\ (REFL)}$ for α NF of α'.
- P= $\boxed{\dfrac{\emptyset}{\alpha_{(\tau^R\tau)}\langle\mu'\rangle \le \mu'}\ \text{RMOD}}$. Since the type of α is marked for restrictivity, $\alpha \in$ **LexVAR**.

  Thus P' = $\boxed{\dfrac{\emptyset}{\alpha_{(\tau^R\tau)}\langle\mu\rangle \le \mu}\ (Rmod)}$ for μ NF of μ'.
- $\boxed{\dfrac{\emptyset}{[\alpha']\ \mathbf{coor}^{C/D}\ (\gamma') \le \alpha'/\gamma'}\ (C1)}$, where $\mathbf{coor}^{C/D}$ is a term $\delta_{(\tau^{C/D}(\tau\tau))}$.

  Then P' = $\boxed{\dfrac{\emptyset}{[\alpha]\ \mathbf{coor}^{C/D}\ (\gamma) \le \alpha/\gamma}\ (C1)}$ 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 P' = $\boxed{\dfrac{\emptyset}{\psi^x[x/\gamma] \equiv \psi^x[x/\gamma]}\ \text{REFL}}$ for ψ^x, γ NF of ψ'^x, γ' resp.

[3] Note that there can be more than one such δ.

- Similarly for η.

Induction hypothesis: Assume that the lemma holds for any P of size less or equal to n.

Step: Let P be a $\mathcal{L}$-OC proof of size n+1. It has one of the following forms:

(i) $$\boxed{\dfrac{\Psi'_1\ \boxed{\dfrac{\bigtriangledown}{\psi_1 \leq \psi'_1}\,R_1} \quad \Psi'_2\ \boxed{\dfrac{\bigtriangledown}{\psi_2 \leq \psi'_2}\,R_2} \quad \cdots \quad \Psi'_n\ \boxed{\dfrac{\bigtriangledown}{\psi_n \leq \psi'_n}\,R_n}}{\psi_1 \leq \psi'_n}\,\textsc{trans}}$$

where $\psi_i = \psi'_{i+1}$ for $1 \leq i \leq n$. Then $\Psi'_1, ..., \Psi'_n$ are $\mathcal{L}$-OC proofs of size at most n. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ 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 P' $= \boxed{\dfrac{\Psi_1 \quad \ldots \quad \Psi_n}{\phi_1 \leq \phi_n}\,\textsc{trans}}$.

(ii) $$\boxed{\dfrac{\Psi'_1\ \boxed{\dfrac{\bigtriangledown}{\psi_1 \leq \psi'_1}} \quad \cdots \quad \Psi'_n\ \boxed{\dfrac{\bigtriangledown}{\psi_n \leq \psi'_n}}}{\alpha'_0 \leq \alpha'}\,R}$$

where $n \in \{1, 2\}$ and $R \neq$ TRANS, then R is one of the following rules:

- $R =$ C2. Then P is of form $\boxed{\dfrac{\Psi'_1\ \boxed{\dfrac{\bigtriangledown}{\psi' \leq \gamma'}} \quad \Psi'_2\ \boxed{\dfrac{\bigtriangledown}{\psi' \leq \delta'}}}{\psi' \leq ([\gamma']\ \mathbf{coor}^C\ )(\delta')}\,(C2)}$. The proofs Ψ'_1, Ψ'_2 are of size at most n. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ 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 P'$= \boxed{\dfrac{\Psi_1\ \boxed{\dfrac{\bigtriangledown}{\psi \leq \gamma}} \quad \Psi_2\ \boxed{\dfrac{\bigtriangledown}{\psi \leq \delta}}}{\alpha \leq ([\gamma]\ \mathbf{coor}^C\ )(\delta)}\,C2}$
- $R =$ D2/MON. The proof is similar to the previous case.
- $R =$Ab. Then P is of form $\boxed{\dfrac{\Psi'\ \boxed{\dfrac{\bigtriangledown}{\mu'^x \leq \varphi'^x}}}{\overline{\lambda} x.\mu'^x \leq \overline{\lambda} x.\varphi'^x}\,(Ab)}$. By the induction hypothesis, there exists a $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ 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

P' $= \dfrac{\Psi \quad \mu^x \leq \varphi^x}{\overline{\lambda} x.\mu^x \leq \overline{\lambda} x.\varphi^x}\,(Ab)$. Otherwise P' $= \dfrac{\Psi \quad \mu^x \leq \varphi^x}{ab(\mu^x) \leq ab(\varphi^x)}\,(\hat{A}b)$.

- $R =$ FR. Then P is of form $\dfrac{\Psi'_1 \quad \mu' \leq \varphi' \qquad \Psi'_2 \quad \delta' \equiv \sigma'}{\mu'\langle\delta'\rangle \leq \varphi'\langle\sigma'\rangle}\,(FR)$. By the induction hypothesis, there exist $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proofs Ψ_1, Ψ_2 of $\mu \leq \varphi$ and $\delta \equiv \sigma$ s.t. $\mu, \varphi, \delta, \sigma$ are NF of $\mu', \varphi', \delta', \sigma'$ resp. which contains NF terms only. If none of the terms μ, φ is an abstraction term, then P' $= \dfrac{\Psi_1 \quad \mu \leq \varphi \qquad \Psi_2 \quad \delta \equiv \sigma}{\mu\langle\delta\rangle \leq \varphi\langle\sigma\rangle}\,(FR)$.

  Otherwise P' $= \dfrac{\Psi_1 \quad \mu \leq \varphi \qquad \Psi_2 \quad \delta \equiv \sigma}{\overline{norm}(\mu\langle\delta\rangle) \leq \overline{norm}(\varphi(\sigma))}\,(\hat{FR})$.

Lemma 5.4 *Let the order statement $\alpha \leq \gamma$ have a $\mathcal{L}$-OC proof P. Then there exists a $\mathcal{L}$-OC$^{\hat{\mathrm{FR}},\hat{\mathrm{Ab}}}$ proof P' of $\alpha \leq \gamma$.*

Proof sketch: We show the construction of a $\mathcal{L}$-OC proof P' of $\alpha \leq \gamma$ from P by elimination of $\hat{\mathrm{FR}}$ and $\hat{\mathrm{Ab}}$. If P does not contain any instances of $\hat{\mathrm{FR}}$ and $\hat{\mathrm{Ab}}$ then P is already a $\mathcal{L}$-OC proof of $\alpha \leq \gamma$. Otherwise suppose P contains an instance of $\hat{\mathrm{FR}}$ $\dfrac{\psi \leq \varphi \qquad \delta \equiv \delta'}{\overline{norm}(\psi(\delta)) \leq \overline{norm}(\varphi(\delta'))}\,(\hat{FR})$ 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:

$$\dfrac{\dfrac{\emptyset}{\zeta^{\overrightarrow{x}}[x/\delta] \equiv \overrightarrow{\lambda} x.\zeta^{\overrightarrow{x}}(\delta)}\,(\beta) \qquad \dfrac{\overrightarrow{\lambda} x.\zeta^{\overrightarrow{x}} \leq \varphi \qquad \delta \equiv \delta'}{\overrightarrow{\lambda} x.\zeta^{\overrightarrow{x}}(\delta) \leq \varphi(\delta')}\,(FR)}{\overline{norm}(\psi(\delta)) = \zeta^{\overrightarrow{x}}[x/\delta] \leq \varphi(\delta') = \overline{norm}(\varphi(\delta'))}\,(Trans)$$

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

$$\dfrac{\boxed{\begin{array}{c}\bigtriangledown\\ \psi^{\overrightarrow{x}} \leq \varphi^{\overrightarrow{x}}\end{array}}}{\overline{ab}(\psi^{\overrightarrow{x}}) \leq \overline{ab}(\varphi^{\overrightarrow{x}})}\ (\hat{Ab})$$

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

$$\dfrac{\dfrac{\emptyset}{\gamma \equiv \overrightarrow{\lambda} x.\gamma(x)}\ (\eta) \qquad \dfrac{\boxed{\begin{array}{c}\bigtriangledown\\ \gamma(x) \leq \varphi^{\overrightarrow{x}}\end{array}}}{\overrightarrow{\lambda} x.\gamma(x) \leq \overrightarrow{\lambda} x.\varphi^{\overrightarrow{x}}}\ (Ab)}{\overline{ab}(\psi^{\overrightarrow{x}}) = \gamma \leq \overrightarrow{\lambda} x.\gamma^{\overrightarrow{x}} = \overline{ab}(\varphi^{\overrightarrow{x}})}\ (Trans)$$

In this way we can eliminate all instances of $\hat{\text{FR}}$ and $\hat{\text{Ab}}$ in P to obtain a valid $\mathcal{L}$-OC proof.
Now let us demonstrate how normalization in $\mathcal{L}\text{-OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ (encapsulated within $\hat{\text{FR}}$ and $\hat{\text{Ab}}$) solves both of the above mentioned problems. Consider again the problematic inference we mentioned above – we show in fig. 4 that it is derivable in $\mathcal{L}\text{-OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$. As

$$\dfrac{\boxed{\dfrac{\dfrac{\emptyset}{\begin{array}{c}[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))\\ \leq \overrightarrow{\lambda} y.Mary(doesn't(y))\end{array}}\ (C_1) \qquad \dfrac{\emptyset}{move \equiv move}\ (Refl)}{\begin{array}{c}[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move)\\ \leq Mary(doesn't(move))\end{array}}\ (\hat{FR})} \qquad \boxed{\dfrac{\dfrac{\dfrac{\emptyset}{walk_{(et)} \leq move_{(et)}}\ (a_2)}{\begin{array}{c}doesn't_{(et)-(et)}(move)\\ \leq doesn't(walk)\end{array}}\ (Mon-)}{\begin{array}{c}Mary_{((et)+t)}(doesn't(move))\\ \leq Mary_{((et)+t)}(doesn't(walk))\end{array}}\ (Mon+)}}{[\overrightarrow{\lambda} x.John(does(x))]and(\overrightarrow{\lambda} y.Mary(doesn't(y)))(move) \leq Mary(doesn't(walk))}\ (Trans)$$

Fig. 4. The problematic inference is derivable in $\mathcal{L}$-OC augmented with normalization axioms: John does and Mary doesn't move $\vdash_{NatLog}$ Mary doesn't walk

for the *effectiveness* consideration, the goal we specified above is also achieved by the normalization axioms. Recall that we aimed at being able to use NF terms to represent $\mathcal{L}$ derivations of the goal sentences. We have shown that for any order statement s.t. $\vdash_{\mathcal{L}-OC} \alpha' \leq \gamma'$, a $\mathcal{L}\text{-OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ proof of the (semantically equivalent) order statement $\alpha \leq \gamma$ s.t. α, γ are NF of α', γ' resp. can be constructed.

6 Proof search procedure

Below we present a proof search procedure for $\mathcal{L}\text{-OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$, which is a variation of a proposal by [2]. 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 $\mathcal{L}\text{-OC}^{\hat{\text{FR}},\hat{\text{Ab}}}$ 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 $S_1, ..., S_m \vdash_{NatLog} S$ in the following way: (a) 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, (b) the order statements $\mathsf{w}_\mathsf{t}^\mathsf{T} \leq \psi_t^i$ for $1 \leq i \leq m$ are added to A (the set of the non-logical axioms), (c) **derive**$(\mathsf{w}_\mathsf{t}^\mathsf{T}, \psi_t, \emptyset)$ is called.

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 $f_{\mathrm{RMOD},r}$. 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$^{\mathrm{F\hat{R},\hat{A}b}}$ 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 [4]. 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 $\alpha \in \mathbf{MON} \uparrow$ ($\alpha \in \mathbf{MON} \downarrow$) a case of a term $\alpha_{(\tau^+\rho)}$ ($\alpha_{(\tau^-\rho)}$), and abbreviate $\alpha \in \mathbf{MON} \uparrow \vee \alpha \in \mathbf{MON} \downarrow$ by $\alpha \in \mathbf{MON}$. We refer to a term $\psi_{(\tau^C(\tau\tau))}$ as $\mathbf{coor}^C$ and to $\psi_{(\tau^D(\tau\tau))}$ – as $\mathbf{coor}^D$.

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**.

- **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 $\phi \in \mathbf{MON} \uparrow$ or $\gamma \in \mathbf{MON} \uparrow$ and **derive**$(\psi, \delta, Goals)$ return **true**.
- **4.2** If $\phi \in \mathbf{MON} \downarrow$ or $\gamma \in \mathbf{MON} \downarrow$ and **derive**$(\delta, \psi, Goals)$ return **true**.
- **4.3** If ($\phi \in \mathbf{MON} \downarrow$ and $\gamma \in \mathbf{MON} \uparrow$) or ($\phi \in \mathbf{MON} \uparrow$ and $\gamma \in \mathbf{MON} \downarrow$) 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**.

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

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 If $\alpha = (\psi_{\tau\tau}(\gamma_\tau))(\delta')$ and **derive**$(\psi_{(\tau\tau)}, id_{(\tau\tau)})$ and **derive**$(\overline{norm}(\gamma(\delta')), \alpha)$ then return **true**.

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

(i) 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**.

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

(iii) 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**.

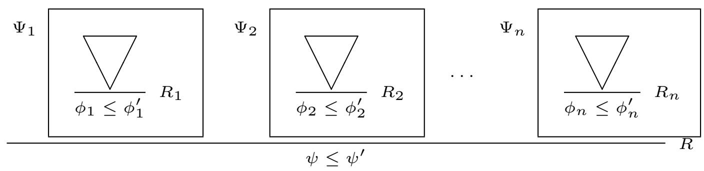
7. **7.1** If $\alpha_0 = \overline{\lambda}x_\tau.\psi^{x_\tau}$ and $x \notin ST(\alpha)$ and **derive**$(\psi^{x_\tau}, \alpha\langle x_\tau\rangle, Goals)$ then return **true**.
7.2 If $\alpha = \overline{\lambda}x_\tau.\psi^{x_\tau}$ and $x \notin ST(\alpha_0)$ and **derive**$(\alpha_0\langle x_\tau\rangle,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

8. return **false**

The algorithm searches for $\mathcal{L}$-OC$^{\mathrm{F\hat{R}},\mathrm{\hat{A}b}}$ proofs of a specific form, which we call *the* $\mathcal{L}$*-OC*$^{F\hat{R},\hat{A}b}$ *normal form.* Let $\mathcal{R}$ be the regular language NL* (R-PROD* ((STR | F̂R)* | Âb | Ab) L-PROD*) (NL$^+$ R-PROD* ((STR | F̂R)* | Âb | Ab) L-PROD*) NL*.

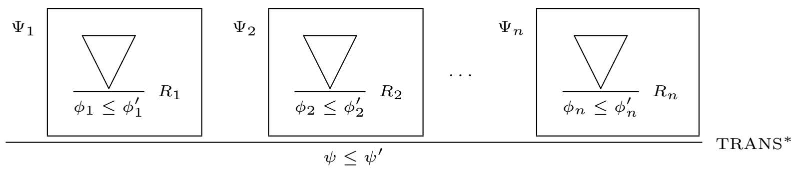
Definition 6.1 ***($\mathcal{L}$-OC$^{F\hat{R},\hat{A}b}$ normal form)*** *Let* $\vdash_{\mathcal{L}-OC^{\mathrm{F\hat{R}},\mathrm{\hat{A}b}}} \psi \leq \psi'$. *Then a* normal form *of a proof of* $\psi \leq \psi'$ *is one of the following structures:*

(i) *A* Type 1 *normal form is*



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

(ii) *A* Type 2 *normal form is*



where (i) $n \geq 2$, *(ii)* $\Psi_1, \ldots, \Psi_n$ *are Type 1 normal form 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 structure of the normal form allows the algorithm to move one step at a time, searching only for the intermediate terms that are structurally related to the terms from the goal order statement. The algorithm attempts to find a proof in a normal form by first separating it into a number of subproofs in the Type 2 normal form (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 Âb are also treated straightforwardly (steps 4,6,7 in **subderive**). F̂R rules are treated in two different ways: (a) in steps 5.1–5.4 of **subderive** the algorithm tries to construct the righthand/lefthand side of the result of F̂R from its lefthand/righthand side, (b) in step 5.5 the algorithm tries to recover the premises of F̂R from its conclusion.

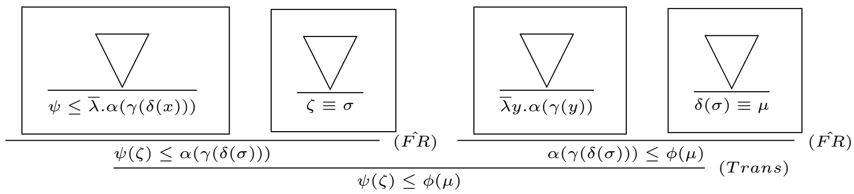
Lemma 6.2 *For any two terms* α_0, α *the call to* **derive**$(\alpha_0, \alpha, \emptyset)$ *terminates.*

The termination is guaranteed by the following facts. First, it can be shown that only a finite set of terms can appear as arguments 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 an argument of **derive** at most once.
As for *completeness* results, it can be shown that the proof search procedure is complete w.r.t. $\widehat{\mathrm{FR}}$ -free $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proofs. The proof has two stages:

(i) Under certain limitations on the lexicon and the $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ -derivable order statements, any $\widehat{\mathrm{FR}}$ -free $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof has a $\widehat{\mathrm{FR}}$ -free normal form.

(ii) Any $\widehat{\mathrm{FR}}$ -free $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof in normal form is found by the algorithm.

The algorithm also finds $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proofs with instances of $\widehat{\mathrm{FR}}$ of several forms. In fig. 5 we show the call tree of the procedure for the $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ 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. 4.
To demonstrate why the $\widehat{\mathrm{FR}}$ is problematic, consider the following $\mathcal{L}$-OC$^{\widehat{\mathrm{FR}},\widehat{\mathrm{Ab}}}$ proof:



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

7 Conclusions

In this paper we focus on augmenting $\mathcal{L}$-OC with normalization axioms, overcoming the problems arising from the emergence of non-NF terms in $\mathcal{L}$-OC proofs. We demonstrate that the normalization axioms allow proving additional order statements in $\mathcal{L}$-OC, leading to new Natural Logic inferences. We also propose a terminating proof search procedure. We believe that the present work has shown some advances in extending the Natural Logic paradigm to a substantial system of reasoning in natural language.

References

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

[2] Fyodorov, Y., Y. Winter and N. Francez, *Order-Based Inference in Natural Logic*, to appear in Journal of Language and Computation (2002).

[3] Moortgat, M., *Categorial type logics*, Handbook for Logic and Language, Johan van Benthem and Alice ter Muelen eds., Elsevier/MIT press (1997), pp. 93–177.

[4] Sánchez, V., *Studies on Natural Logic and Categorial Grammar*, Ph.D. dissertation, University of Amsterdam (1991).

[5] Steedman, M., *The syntactic process*, MIT press (2000).

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

In step 3 of **derive**:

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

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

In step 5.1.1. of **subderive**:

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

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

$\mathbf{derive}(Mary(doesn't(move)), Mary(doesn't(walk)), Goals')$

In step 3 of **derive**:

$\mathbf{subderive}(Mary(doesn't(move)), Mary(doesn't(walk)), Goals'')$

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

In step 4 of **subderive**:

$\mathbf{derive}(Mary, Mary, Goals'')$

In step 3 of **derive**:

$\mathbf{subderive}(Mary, Mary, Goals''')$

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

$\mathbf{derive}(doesn't(walk), doesn't(move), Goals'')$

In step 3 of **derive**:

$\mathbf{subderive}(doesn't(walk), doesn't(move), Goals''')$

In step 4 of **subderive**:

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

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

Fig. 5. The call tree of the algorithm for the $\mathcal{L}$-OC$^{\hat{FR},\hat{Ab}}$ proof from fig. 4

[6] Wansing, H., *The logic of information structures*, Springer Lecture Notes, Springer-Verlag, Berlin. AI 681 (1993).

[7] Zamansky, A., 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).