A Monotonicity Calculus and Its Completeness

Thomas F. Icard
Stanford University
Stanford, CA, USA
icard@stanford.edu

Lawrence S. Moss
Indiana University
Bloomington, IN, USA
lsm@cs.indiana.edu

William Tune
Motlow State Community College
Smyrna, TN, USA
wtune@mscc.edu

Abstract

One of the prominent mathematical features of natural language is the prevalence of "upward" and "downward" inferences involving determiners and other functional expressions. These inferences are associated with negative and positive polarity positions in syntax, and they also feature in computer implementations of textual entailment. Formal treatments of these phenomena began in the 1980's and have been refined and expanded in the last 10 years. This paper takes a large step in the area by extending typed lambda calculus to the *ordered setting*. Not only does this provide a formal tool for reasoning about upward and downward inferences in natural language, it also applies to the analysis of monotonicity arguments in mathematics more generally.

1 Introduction

Monotonicity reasoning is pervasive across many domains, from mathematics to natural language, indeed in any setting that deals with functions of ordered sets. A function f is *monotone* if it preserves order, that is, if $x \leq y$ implies $f(x) \leq f(y)$. Anti-monotone (or *antitone*) functions f are those that reserve order, that is, for which $x \leq y$ implies $f(y) \leq f(x)$. Natural language constructions that exhibit these patterns are ubiquitous, spanning semantic and grammatical categories. Algorithms have been devised and studied for deriving monotonicity patterns in complex expressions composed of simpler functional expressions (van Benthem, 1986; Sánchez-Valencia, 1991; van Eijck, 2007). For instance, the interaction of quantifier and temporal expressions, together with the fact that $2 \leq 5$, guarantee that *Any play that lasts more than 2 hours is too long* entails (is "less than" in a sense to be made precise) *Any play that lasts more than 5 hours is too long*. There has been recent theoretical work on monotonicity reasoning as part of a general interest in "natural logic" (Bernardi, 2002; Zamansky et al., 2006; MacCartney and Manning, 2009; Muskens, 2010; Icard, 2012; Moss, 2012; Icard and Moss, 2013; Tune, 2016), and much of this work has made its way into psycholinguists (Geurts, 2003; Geurts and van der Slik, 2005) and natural language processing (MacCartney and Manning, 2007; Angeli and Manning, 2014; Bowman et al., 2015; Abzianidze, 2015). (For review see Icard and Moss 2014.)

Whereas monotonicity reasoning in natural language is often seen as comprising a fragment of higher-order logic, we can also construe it as encoding a logical system in its own right relative to a suitably coarsened model-theoretic interpretation. In this context standard metalogical questions such as *completeness* can be raised. A completeness result would tell us that a proof system is sufficient to derive everything that follows on the intended model of monotonicity reasoning.

Though our primary interest here is natural language, it bears mention that such reasoning in higher-order settings is also ubiquitous in other areas, e.g., in mathematics. Consider the convergence test for improper integrals, which states that if $0 \leq f(x) \leq g(x)$ on an interval $[a, \infty)$, then $\int_a^\infty f(x)dx$ converges if $\int_a^\infty g(x)dx$ does. As an example of this, note that knowing $\int_1^\infty e^{-x}dx = e^{-1}$ converges allows, by monotonicity reasoning alone, to infer that $\int_1^\infty e^{-x^2}$ also converges:

$$\dfrac{\dfrac{\dfrac{\dfrac{1 \leq x}{x \leq x^2}}{-x^2 \leq -x}}{e^{-x^2} \leq e^{-x}}}{\int_1^\infty e^{-x^2}dx \leq \int_1^\infty e^{-x}dx}$$

Proceedings of the 15th Meeting on the Mathematics of Language, pages 75–87,
London, UK, July 13–14, 2017. ©2017 Association for Computational Linguistics

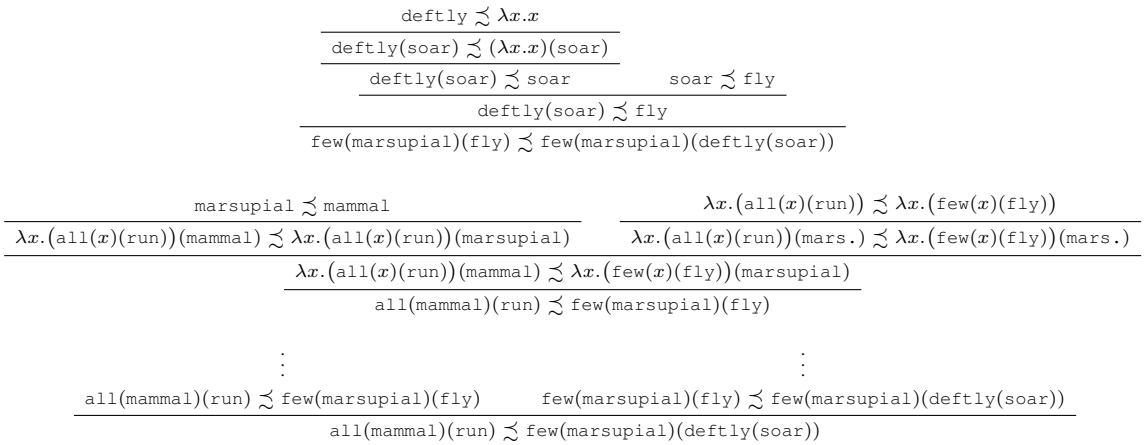

Figure 1: Example proof that *All mammals run* implies *Few marsupials deftly soar* (in three parts).

This argument, similar to those we will be considering, depends only on the *monotonicity profiles* of the relevant functions (multiplication, exponentiation, etc.) on the relevant domains.

The aim of the present contribution is to formulate a suitable system for monotonicity reasoning in a higher-order setting, appropriate to the task of capturing common entailment patterns in natural language in particular, and to prove a completeness result for an associated proof system. Along the way we also prove an analogue of Lyndon's (1959) Theorem for first order logic, showing exactly when, in our general setting, a subterm occurrence stands in a monotone or antitone position. For reasons of space, we skip some of the less central proofs.

2 Motivating Example

To motivate the specific formal apparatus that we will employ, consider the following small fragment. As in previous work (Icard and Moss, 2013), we will be considering an extended simply typed lambda calculus where the functional types can be "marked" with monotonicity information, $+$ for monotone, $-$ for antitone, and $\cdot$ for neither (or unknown). Suppose we have two base types t and p, corresponding to truth values and predicates (more commonly, functions from entity type to truth value type), and the following typed terms:

$$\begin{array}{lcl} \texttt{all} & : & p \xrightarrow{-} (p \xrightarrow{+} t) \\ \texttt{few} & : & p \xrightarrow{\cdot} (p \xrightarrow{-} t) \\ \texttt{mammal},\ \texttt{marsupial} & : & p \\ \texttt{run},\ \texttt{fly},\ \texttt{soar} & : & p \\ \texttt{deftly} & : & p \xrightarrow{\cdot} p \end{array}$$

Let us furthermore assume the following background entailment facts Γ, where $M \precsim N$ is understood as entailment generalized to all types:

$$\begin{array}{rcl} \lambda x.\big(\texttt{all}(x)(\texttt{run})\big) & \precsim & \lambda x.\big(\texttt{few}(x)(\texttt{fly})\big) \\ \texttt{deftly} & \precsim & \lambda x.x \\ \texttt{marsupial} & \precsim & \texttt{mammal} \\ \texttt{soar} & \precsim & \texttt{fly} \end{array}$$

The first statement encodes the assumption that if all members of a given category run, it can be inferred that few members of that category fly. The second statement essentially says that *deftly* is subsective (Kamp and Partee, 1995): deftly v'ing involves v'ing (see Figure 1). The third and fourth capture basic lexical entailments. We can then use these assumptions to derive *Few marsupials deftly soar* from *All mammals run*. A proof using our monotonicity calculus appears in Figure 1.

There are several important points to notice about this example. First, in order to state and use assumptions such as the first two above, we make crucial use of lambda abstraction and β-reduction.

Second, note that we can state entailment facts between terms even when those terms have different (marked) types, as in the first two statements. For example, though in our typing system $\lambda x.x$ will be of type $p \xrightarrow{+} p$, it can nonetheless be compared with `deftly` because (denotations of) terms of type $p \xrightarrow{+} p$ can be semantically "coerced" to type $p \xrightarrow{\cdot} p$. This is simply because the domain $D_{p \xrightarrow{\cdot} p}$ for terms of type $p \xrightarrow{\cdot} p$ will be the class of all functions from D_p to D_p, which certainly includes all the monotone functions.

Third, it can be useful to derive monotonicity information for complex terms, e.g., so that

we can derive $\lambda x.\big(\texttt{all}(x)(\texttt{run})\big)(\texttt{mammal}) \precsim \lambda x.\big(\texttt{all}(x)(\texttt{run})\big)(\texttt{marsupial})$ in one step. Theorem 8.2 below guarantees that the way we type lambda abstractions is in a sense optimal.

The framework developed in this paper is motivated by the desire to capture patterns like these. Such patterns could be derived in an inequational system of full higher-order logic: given a constant $\vee$ for disjunction at a given type, it is easy to see that a term f will be a monotone function just in case we have $\lambda x.\lambda y.f(x) \precsim \lambda x.\lambda y.f(x \vee y)$. Proofs of facts like that above might then be derived in a higher order logic proof system with monotonicity declarations as additional premises. We of course could not have completeness in this setting, but more importantly, we would rather like to isolate and understand what is characteristic of monotonicity reasoning as such.

There are many instances of this kinds of reasoning outside of natural language. As a simple illustration of the main concepts and definitions, throughout the paper we will be considering a running example of elementary mathematical reasoning about real number functions.

3 Types and Domains

Our set $\mathcal{T}$ of types is defined inductively from a set $\mathcal{B}$ of base types b:

$$\tau \quad ::= \quad b \mid \tau \dot{\to} \tau \mid \tau \overset{+}{\to} \tau \mid \tau \overset{-}{\to} \tau$$

Definition 3.1 (Markings and types)**.** The set Mar of *markings* is $\{+, -, \cdot\}$. We use m and m' to denote markings. We always take Mar to be ordered with $+ \sqsubseteq \cdot$, $- \sqsubseteq \cdot$, and $m \sqsubseteq m$ for all m. We also define a binary operation $\circ$ on Mar by $+ \circ + = +$, $+ \circ - = -$, $- \circ + = -$, $- \circ - = +$; and otherwise $m \circ m' = \cdot$. Notice that $\circ$ is associative.

We have a natural ordering $\preceq$ on types, where $\sigma \preceq \tau$ can be read as: any term of type σ could also be considered of type τ (cf. Def. 3.5 below).

Definition 3.2 ($\preceq$ on types)**.** Define $\preceq \subseteq \mathcal{T} \times \mathcal{T}$ to be the least preorder with the property that whenever $\sigma' \preceq \sigma$ and $\tau \preceq \tau'$, and $m \sqsubseteq m'$, we have $\sigma \overset{m}{\to} \tau \preceq \sigma' \overset{m'}{\to} \tau'$.

Definition 3.3 (the functions $\uparrow$, $\vee$, and $\sigma \mapsto \hat{\sigma}$ on types)**.** $(\text{Mar}, \sqsubseteq)$ is an upper semilattice. So we have an operation $\vee$ on it. Explicitly, $m \vee m = m$ for all m, and for $m \neq m'$, $m \vee m' = \cdot$. We also define $\uparrow$ to be the smallest relation on types, and $\vee$ to be the smallest partial function on types, with the properties that for all σ, τ_1, and τ_2:

1. $\sigma \uparrow \sigma$, and $\sigma \vee \sigma = \sigma$.
2. If $\tau_1 \uparrow \tau_2$, then $(\sigma \overset{m_1}{\to} \tau_1) \uparrow (\sigma \overset{m_2}{\to} \tau_2)$ for all $m_1, m_2 \in \text{Mar}$, $(\sigma \overset{m_1}{\to} \tau_1) \vee (\sigma \overset{m_2}{\to} \tau_2) = \sigma \overset{m_1 \vee m_2}{\longrightarrow} (\tau_1 \vee \tau_2)$.

Finally, we define $\sigma \mapsto \hat{\sigma}$ on $\mathcal{T}$ by $\hat{\sigma} = \sigma$ for σ basic, and $\widehat{(\sigma \overset{m}{\to} \tau)} = \sigma \dot{\to} \hat{\tau}$.

We also note the following characterization of the order $\preceq$. In it, we use the height function defined by: $\text{ht}(\sigma) = 0$ for σ basic, and $\text{ht}(\sigma \overset{m}{\to} \tau) = 1 + \max(\text{ht}(\sigma), \text{ht}(\tau))$.

Proposition 3.4. Let R_0 be the identity relation on the set $\mathcal{T}$ of types. Given R_n, let

$$R_{n+1} = \{(\sigma \overset{m}{\to} \tau, \sigma' \overset{m'}{\to} \tau') : \sigma, \sigma', \tau, \tau' \text{ have height} \leq n; m \sqsubseteq m'; \text{ and both } (\sigma', \sigma), (\tau, \tau') \text{ belong to } R_n\}$$

Then $\bigcup_n R_n$ is the order $\preceq$.

Definition 3.5. A *pre-structure* $\mathbb{D} = \{\mathbb{D}_\tau\}_{\tau \in \mathcal{T}}$ is given by a class of preorders $\mathbb{D}_\tau = (D_\tau, \leq_\tau)$ for each type $\tau \in \mathcal{T}$, and a family of maps

$$\pi_{\sigma,\tau} : \mathbb{D}_\sigma \to \mathbb{D}_\tau$$

when $\sigma \preceq \tau$, subject to the following constraints:

1. $D_{\sigma \overset{+}{\to} \tau} \cup D_{\sigma \overset{-}{\to} \tau} \subseteq D_{\sigma \dot{\to} \tau} \subseteq D_\tau^{D_\sigma}$.
2. $f \in D_{\sigma \overset{+}{\to} \tau}$ and $a \leq_\sigma b$ imply $f(a) \leq_\tau f(b)$.
3. $f \in D_{\sigma \overset{-}{\to} \tau}$ and $a \leq_\sigma b$ imply $f(b) \leq_\tau f(a)$.
4. $f \leq_{\sigma \overset{m}{\to} \tau} g$ iff for all $a \in D_\sigma : f(a) \leq_\tau g(a)$.
5. $\pi_{\sigma,\sigma}$ is the identity on $\mathbb{D}_\sigma$.
6. If $\sigma \preceq \tau \preceq \mu$, then $\pi_{\sigma,\mu} = \pi_{\tau,\mu} \circ \pi_{\sigma,\tau}$.
7. Each map $\pi_{\sigma,\tau}$ is order-preserving.

Definition 3.6. Here is a family of pre-structures called the *full pre-structures* based on an assignment of preorders $\mathbb{D}_\sigma$ to base types σ. Then one defines $\mathbb{D}_\sigma$ by recursion on the height of σ:

$$\begin{aligned} \mathbb{D}_{\sigma \dot{\to} \tau} &= (D_\tau)^{D_\sigma}, \text{ all functions from } D_\sigma \text{ to } D_\tau \\ \mathbb{D}_{\sigma \overset{+}{\to} \tau} &= \{f \in \mathbb{D}_{\sigma \dot{\to} \tau} : f \text{ is monotone}\} \\ \mathbb{D}_{\sigma \overset{-}{\to} \tau} &= \{f \in \mathbb{D}_{\sigma \dot{\to} \tau} : f \text{ is antitone}\} \end{aligned}$$

The order in all cases is the pointwise order. We define the maps $\pi_{\sigma,\tau}$ in terms of the characterization in Proposition 3.4. For $n = 0$, the only time we have $(\sigma, \tau) \in R_0$ is when $\sigma = \tau$; in this case,

we set $\pi_{\sigma,\sigma}$ to be the identity on $\mathbb{D}_\sigma$. Notice that each $\pi_{\sigma,\tau}$ is order preserving (since τ must be σ when $n = 0$), and also that $\pi_{\sigma,\mu} = \pi_{\tau,\mu} \circ \pi_{\sigma,\tau}$.

Given $\pi_{\sigma,\tau}$ for all pairs $(\sigma, \tau) \in R_n$, here is how we extend the definition to $R_{n+1} \setminus R_n$. Given $\sigma' \preceq \sigma$ and $\tau \preceq \tau'$, $m \sqsubseteq m'$, and also $\pi_{\sigma,\sigma'}$ and $\pi_{\tau,\tau'}$, we define $\pi_{\sigma \xrightarrow{m} \tau, \sigma' \xrightarrow{m'} \tau'}$ to be

$$k \in D_{\sigma \xrightarrow{m} \tau} \quad \mapsto \quad \pi_{\tau,\tau'} \circ k \circ \pi_{\sigma',\sigma}. \qquad (1)$$

It is easy to verify the properties in Def. 3.5.

Example 3.7. Take $\mathcal{B} = \{r\}$, with r intuitively standing for *real numbers*. Then we will have types in $\mathcal{T}$ such as those shown below. As an example of $\uparrow$, $(r \xrightarrow{-} r) \uparrow (r \xrightarrow{+} r)$, while $(r \xrightarrow{-} r) \vee (r \xrightarrow{+} r) = r \xrightarrow{\cdot} r$. We build the full pre-structure using $\mathbb{D}_r = \mathbb{R}$, the reals with the usual order $\leq$. Then we have, e.g.,

σ	$\mathbb{D}_\sigma$
$r \xrightarrow{\cdot} r$	functions from $\mathbb{R}$ to $\mathbb{R}$
$r \xrightarrow{+} r$	monotone functions from $\mathbb{R}$ to $\mathbb{R}$
$r \xrightarrow{-} r$	antitone functions from $\mathbb{R}$ to $\mathbb{R}$
$r \xrightarrow{+} (r \xrightarrow{-} r)$	monotone functions from $\mathbb{R}$ to $\mathbb{D}_{r \xrightarrow{-} r}$

4 The Language $\mathcal{L}_\lambda$ of Terms

Our language $\mathcal{L}_\lambda$ is a variant of the typed λ-calculus which makes use of the marked types that we saw in Section 3.

We begin with a set $\mathcal{C}$ of constants, each coming with a unique type, and a set $\mathcal{V}$ of variables, also with their types. We define the language $\mathcal{L}_\lambda$ of all terms using a typing calculus. Beginning with a set of typing statements determined from $\mathcal{C}$ and $\mathcal{V}$, we define several things simultaneously: *terms with their types* (denoted $M : \sigma$, $N : \tau$, etc.), *occurrences of free variables in terms*, and *the valence of each free variable occurrence in* M.

1. For a variable $x : \tau$ in $\mathcal{V}$, $x : \tau$ is a term. Further, x occurs free in itself in the evident way, and x is the only variable that occurs free in itself. The valence is $+$.

2. Each constant $c : \sigma$ is a term, so there are no free occurrences of any variables in c.

3. Let $m \in \mathrm{Mar}$. We have the following rule:

$$\frac{M : (\sigma \xrightarrow{m} \tau) \qquad N : \sigma'}{M(N) : \tau} \; \sigma' \preceq \sigma$$

   The free occurrences of x in $M(N)$ are the free occurrences of x in M together with the free occurrences of x in N.

   Any free occurrence of x in $M(N)$ is either an occurrence in M, or an occurrence in N:

   (a) For a free occurrence of x in M, the valence in $M(N)$ is that in M.

   (b) For a free occurrence of x in N, with valence m', the valence of x in $M(N)$ is $m \circ m'$, where $M : \sigma \xrightarrow{m} \tau$.

4. Finally, if x is a variable,

$$\frac{x : \sigma \qquad M : \tau}{\lambda x.M : \sigma \xrightarrow{m} \tau}$$

   If all free occurrences of x in M are $+$, and if there is at least one free occurrence of x in M, then $m = +$. If all free occurrences of x in M are $-$, and if there is at least one free occurrence of x in M, then $m = -$. If there are either no free occurrences of x in M, or if there are free occurrences but they are not all $+$ and also not all $-$, then $m = \cdot$.

   There are no free occurrences of x in $\lambda x.M$. The free occurrences of variables $y \neq x$ in $\lambda x.M$ are the free occurrences of y in M. Those occurrences have the same valence in $\lambda x.M$ as they have in M.

We define $FV(M)$ to be the variables with free occurrences in M. We define $BV(M)$ to be the variables with bound occurrences in M. (We have not defined these, but they are defined as usual.) The main point about the valences of variable occurrences will come shortly, in Lemma 5.2.

Remark 4.1. Note that every term M has a unique type. For this reason, we often omit the type when it is not pertinent to the discussion.

Example 4.2. We build on Example 3.7. Let us take the set $\mathcal{C}$ of constants to be given as follows:

constant c	type σ	standard interpretation
0	r	0
1	r	1
$+$	$r \xrightarrow{+} r \xrightarrow{+} r$	$a \mapsto (b \mapsto a + b)$
$-$	$r \xrightarrow{+} r \xrightarrow{-} r$	$a \mapsto (b \mapsto a - b)$

We shall present the semantics of terms in Section 5 below. The "standard interpretation" is not quite the semantics $[\![\ ]\!]$ in our sense because $[\![\ ]\!]$ is defined in terms of valuations. The difference is

term M	type σ	$[\![M]\!]_\phi$
$+(0)(1)$	r	1
$+(1)(1)$	r	2
$-(0)(1)$	r	-1
x	r	$\phi(x)$
$-(x)$	$r \bar{\to} r$	$b \mapsto (\phi(x) - b)$
$-(0)$	$r \bar{\to} r$	$b \mapsto -b$

term M	type σ	$[\![M]\!]_\phi$
$-(1)(x)$	r	$1 - \phi(x)$
$-(x)(1)$	r	$\phi(x) - 1$
$-(x)(y)$	r	$\phi(x) - \phi(y)$
$\lambda x.-(1)(x)$	$r \bar{\to} r$	$a \mapsto (1 - a)$
$\lambda x.-(x)(1)$	$r \overset{+}{\to} r$	$a \mapsto (a - 1)$
$\lambda f.\lambda x.-(0)(f(x))$	$(r \dot{\to} r) \to (r \dot{\to} r)$	$f \mapsto -f$

Figure 2: Examples of terms, types, and interpretations, under an arbitrary valuation ϕ. See Example 3.7.

harmless. Further, we take variables $x, y, z : r$, and $f, g : r \dot{\to} r$. Figure 2 has examples of terms, again with types and semantics under a valuation ϕ. We assume that the semantics interprets the constants as above. Of course, it would be more sensible to write $1 + 1$ instead of $+(1)(1)$.

5 Semantics of $\mathcal{L}_\lambda$: Structures

At this point, we turn to the semantics of our language. We interpret $\mathcal{L}_\lambda$ in what we call *structures*. These are pre-structures together with additional information needed to interpret variables and constant symbols.

Definition 5.1. Let $\mathbb{D}$ be a pre-structure. We let $\Phi = \Phi(\mathbb{D})$ be the set of functions ϕ whose domain is the set of (typed) variables, with the property that if $x : \sigma$, then $\phi(x) \in D_\sigma$. We call such functions ϕ *valuations in* $\mathbb{D}$.

An *interpretation function in* $\mathbb{D}$ is a function

$$[\![\ ]\!] : \mathcal{L}_\lambda \times \Phi \to \mathbb{D},$$

mapping the terms M of the language $\mathcal{L}_\lambda$ together with valuations to elements of $\mathbb{D}$. As before, we require that if $M : \sigma$, then $[\![M]\!]_\phi \in D_\sigma$.

A *structure* is a pair

$$\mathbb{S} \quad = \quad (\mathbb{D}, [\![\ ]\!]),$$

where $\mathbb{D}$ is a pre-structure, and $[\![\ ]\!]$ is an interpretation in $\mathbb{D}$ such that the following conditions hold:

1. For $M : \sigma \overset{m}{\to} \tau$, and $N : \sigma' \preceq \sigma$, $[\![M(N)]\!]_\phi = [\![M]\!]_\phi\big(\pi_{\sigma',\sigma}[\![N]\!]_\phi\big)$.
2. For $M : \sigma$, $x : \tau$, and $a \in \mathbb{D}_\tau$, $[\![\lambda x.M]\!]_\phi(a) = [\![M]\!]_{\phi^a_x}$.

In the last point, we use our notation for modifying functions when we write

$$\phi^a_x(y) \quad = \quad \begin{cases} \phi(y) & \text{if } y \neq x \\ a & \text{if } y = x \end{cases}$$

Again, see Figure 2 for examples.

5.1 Positivity Entails Monotonicity; Negativity Entails Antitonicity

Recall that positive or negative occurrences of variables are syntactic notions, whereas monotonicity and antitonicity are semantic notions. One of the contributions of this paper is to explore the connection between these notions.

Lemma 5.2. Let $M : \tau$, and let $x : \sigma$ be a variable. Let $\mathbb{S} = (\mathbb{D}, [\![\ ]\!])$ be any structure.

1. If all free occurrences of x in M are $+$, then for all ϕ, $a \mapsto [\![M]\!]_{\phi^a_x}$ is monotone.
2. If all free occurrences of x in M are $-$, then for all ϕ, $a \mapsto [\![M]\!]_{\phi^a_x}$ is antitone.

Proof. By induction on M. We prove both parts simultaneously.

Let M be a variable. We have two cases, depending on whether $M = x$ or not. If $M = x$, then all occurrences of x in x are $+$. Moreover, $a \mapsto [\![M]\!]_{\phi^a_x}$ is the identity and hence monotone in a. (Also, it is not the case that all occurrences of x in x are $-$.) If M is a variable $y \neq x$, then all occurrences of x in M (there are none) are both $+$ and $-$. And in this case, $[\![M]\!]_{\phi^a_x} = \phi(y)$. So $a \mapsto [\![M]\!]_{\phi^a_x}$ is a constant function. As such, it is both monotone and antitone.

Now suppose (1) and (2) for M and for N, and consider $M(N)$. First, suppose that all free occurrences of x in $M(N)$ are $+$. Then all free occurrences of x in M are $+$. We have two cases.

First, we consider the case when M is of functional type $\sigma \overset{+}{\to} \tau$. In this case, all occurrences of x in N must be $+$. By induction hypothesis, $a \mapsto [\![M]\!]_{\phi^a_x}$ is monotone, and so is $a \mapsto [\![N]\!]_{\phi^a_x}$. Hence so is $a \mapsto [\![M(N)]\!]_{\phi^a_x}$. In more detail, let $a \leq b$. Then

$$\begin{aligned} [\![M(N)]\!]_{\phi^a_x} &= [\![M]\!]_{\phi^a_x}([\![N]\!]_{\phi^a_x}) \\ &\leq [\![M]\!]_{\phi^a_x}([\![N]\!]_{\phi^b_x}) \\ &\leq [\![M]\!]_{\phi^b_x}([\![N]\!]_{\phi^b_x}) \\ &= [\![M(N)]\!]_{\phi^b_x}. \end{aligned}$$

We have suppressed the type information on the inequality signs $\leq$.

The case when M is of negative functional type is similar. This concludes our (abridged) discussion of point (1).

We turn to (2). Suppose that all free occurrences of x in $M(N)$ are $-$. Then all free occurrences of x in M are $-$. We again have two cases.

First, we consider the case when M is $+$. So all free occurrences of x in N are $-$. Thus $a \mapsto [\![M]\!]_{\phi_x^a}$ is monotone, and $a \mapsto [\![N]\!]_{\phi_x^a}$ is antitone. Each $[\![M]\!]_{\phi_x^a}$ is antitone. Now let $a \leq b$. Then

$$\begin{array}{rcl} [\![M(N)]\!]_{\phi_x^b} & = & [\![M]\!]_{\phi_x^b}([\![N]\!]_{\phi_x^b}) \\ & \leq & [\![M]\!]_{\phi_x^a}([\![N]\!]_{\phi_x^b}) \\ & \leq & [\![M]\!]_{\phi_x^a}([\![N]\!]_{\phi_x^a}) \\ & = & [\![M(N)]\!]_{\phi_x^a}. \end{array}$$

Finally, we have the case that M is $-$. Each $[\![M]\!]_\phi$ is antitone, and $a \mapsto [\![M]\!]_{\phi_x^a}$ is antitone. Further, all free occurrences of x in N are $+$, so $a \mapsto [\![N]\!]_{\phi_x^a}$ is monotone. And for $a \leq b$ we have

$$\begin{array}{rcl} [\![M(N)]\!]_{\phi_x^b} & = & [\![M]\!]_{\phi_x^b}([\![N]\!]_{\phi_x^b}) \\ & \leq & [\![M]\!]_{\phi_x^b}([\![N]\!]_{\phi_x^a}) \\ & \leq & [\![M]\!]_{\phi_x^a}([\![N]\!]_{\phi_x^a}) \\ & = & [\![M(N)]\!]_{\phi_x^a}. \end{array}$$

This concludes our work on application terms.

We conclude the overall induction by considering abstraction terms $\lambda y.M$. Let $a \leq b$. To see that $[\![\lambda y.M]\!]_{\phi_x^a} \leq_{\tau \stackrel{m}{\to} \sigma} [\![\lambda y.M]\!]_{\phi_x^b}$, let $d \in \mathbb{D}_\sigma$:

$$\begin{array}{rcll} [\![\lambda y.M]\!]_{\phi_x^a}(d) & = & [\![M]\!]_{(\phi_x^a)_y^d} & \\ & = & [\![M]\!]_{(\phi_y^d)_x^a} & \\ & \leq & [\![M]\!]_{(\phi_y^d)_x^b} & \text{ind. hyp.} \\ & = & [\![M]\!]_{(\phi_x^b)_y^d} & \\ & = & [\![\lambda y.M]\!]_{\phi_x^b}(d) & \end{array}$$

Note that we apply the induction hypothesis to ϕ_y^d, not to ϕ. This for all d shows (1) for $\lambda y.M$. (Recall that we are using the pointwise order for functional types $\tau \stackrel{m}{\to} \sigma$.) The same reasoning applies to (2) for the term $\lambda y.M$.

This concludes the proof. $\dashv$

Example 5.3. The full pre-structures introduced in Definition 3.6 give structures in the following way. We define $[\![M]\!]_\phi$ by recursion on M, simultaneously for all ϕ, and we also at the same time verify that for $M : \sigma$, $[\![M]\!]_\phi \in D_\sigma$. The definitions are related to (but not identical to) what we saw in the definition of a structure:

1. For $M : \sigma \stackrel{m}{\to} \tau$, and $N : \sigma' \preceq \sigma$, $[\![M(N)]\!]_\phi = [\![M]\!]_\phi\big(\pi_{\sigma',\sigma}[\![N]\!]_\phi\big)$.
2. For $M : \sigma$, and $x : \tau$, $[\![\lambda x.M]\!]_\phi$ is the function $a \mapsto [\![M]\!]_{\phi_x^a}$.

Now along with the definition, we carry along the proof of Lemma 5.2. We do this in order to know that the typings of the abstractions $\lambda x.M$ are correct. For example, suppose that our typing has $\lambda x.M : \sigma \stackrel{+}{\to} \tau$. According to our definition in Section 4, we know that all free occurrences of x in M are $+$. So by Lemma 5.2, we know that $[\![\lambda x.M]\!]_\phi$ as defined above really is a monotone function; that is, it really belongs to $D_{\sigma \stackrel{+}{\to} \tau}$.

6 Term Substitution and Reduction

A *substitution* is a function s from variables to terms, sending $x : \sigma$ to some $s(x) : \sigma' \preceq \sigma$.

One example is the identity substitution Id. For any substitution s, and any variable $x : \sigma$ and $M : \sigma' \preceq \sigma$, we get a new substitution s_x^M, defined by $s_x^M(x) = M$, and for $y \neq M$, $s_x^M(y) = s(y)$. When the subscript/superscript notation becomes cumbersome, we might change it. For example, we usually write Id_x^M as $[M/x]$.

The notion of capture-avoiding substitution is something of a challenge to get correct. We adopt the definitions of Stoughton (1988) and then quote the results from this paper, adapted to our setting.

Given a term M and a substitution s, we define $M[s]$ by induction on s. It represents the result of substituting, for each x, $s(x)$ for every free occurrence of x in s. We only use the notation $M[s]$ when no variable occurs bound in M and free in any $s(x)$. (That is, we insist that no variable free in any $s(x)$ has bound occurrences in M.)

$$\begin{array}{lcl} x[s] & = & s(x) \\ M(N)[s] & = & M[s](N[s]) \\ (\lambda x.M)[s] & = & \lambda y.(M[s_x^y]) \end{array} \qquad (2)$$

In the last line, y is the least variable in some pre-set list such that y is not free in M, nor in any $s(z)$ for z free in M. Also s_x^y is just like s except that $s_x^y(x) = y$. But y can be any variable z with those properties; by Corollary 3.11 of Stoughton (1988), the result $\lambda z.(M[s_x^z])$ will be α-equivalent to $\lambda y.(M[s_x^y])$. (We define α-equivalence below.)

Lemma 6.1. For all terms M and substitutions s, $M[s]$ is a proper term, and the type of $M[s]$ is $\preceq$ the type of M.

Lemma 6.2. Let M be a term, and consider a free occurrence of a variable x in M with valence m. Let s be a substitution, and let y be a variable which occurs free in $s(x)$ with valence m'. Then in $M[s]$, the free occurrences of y which arise as substitutions for the given occurrence of x all have valence $m \circ m'$.

Proof. By induction on M. When M is a variable, this variable must be x. Since the valence of x in itself is $+$, and since $+$ is a neutral element for $\circ$, our result follows.

When M is a constant, our result is vacuous.

Consider an application term $M(N)$, and assume our lemma for M and for N. Recall that $M(N)[s] = (M[s])(N[s])$. Consider a free occurrence of x in $M(N)$.

First, we consider the case when our free occurrence of x in $M(N)$ is actually a free occurrence in M. In this case, the valence of all the corresponding occurrences of y in $M(N)[s]$ is the same as the valence of those occurrences in $M[s]$. And so the result in this case follows easily from the induction hypothesis.

Second, we have the case when our free occurrence of x in $M(N)$ is a free occurrence of x in N. Let m_1 and m_2 be such that $M : \sigma \xrightarrow{m_1} \tau$ and the valence of our occurrence in N is m_2. Then m, the valence of x in $M(N)$, is $m_1 \circ m_2$. The corresponding occurrences of y in $(M(N))[s]$ are free occurrences of y in $N[s]$, and by induction hypothesis, their valences there are $m_2 \circ m'$. So their valences in $(M(N))[s]$ are $m_1 \circ (m_2 \circ m') = (m_1 \circ m_2) \circ m' = m \circ m'$. (We have used the associativity of $\circ$.) This is as desired.

We conclude with the induction step for abstraction. Let M be $\lambda z.N$ with $z \neq x$. We assume our lemma for N, and we have an occurrence of x in M; its valence there is the same as the valence of the corresponding occurrence in N. Recall that $M[s]$ is $\lambda w.N[s]$, with w suitably fresh. A free occurrence of y in $M[s]$ corresponds to a free occurrence in $N[s]$, and the valence is the same. Our result follows from the induction hypothesis. $\dashv$

The next two results will guarantee that the usual reduction rules of lambda calculus involve well-defined operations on our set of terms.

Theorem 6.3 (Subject Reduction Theorem)**.** The type of $M[N/x]$ is $\preceq$ the type of $(\lambda x.M)N$.

Proof. The type of $(\lambda x.M)N$ is the type of M, so the result follows from Lemma 6.1. $\dashv$

Theorem 6.4 (Subject Reduction Theorem for valences)**.** Consider a free occurrence occ of y in $(\lambda x.M)N$ with valence m, either $+$ or $-$. Also, consider the term that results from (β) reduction, $M[N/x]$. Then the occurrences of y in $M[N/x]$ which correspond to occ also have valence m.

Proof. If the free occurrence of y is in $\lambda x.M$, then our result is easy. So we focus on the case when it is in N. Now $m = m_1 \circ m_2$, where m_1 is such that $\lambda x.M : \sigma \xrightarrow{m_1} \tau$, and m_2 is the valence of occ in N. We are assuming that m_1 is either $+$ or $-$. By the way we type abstractions, all free occurrences of x in M have valence m_1. By Lemma 6.2, the occurrences of y which correspond to occ also have valence $m_1 \circ m_2$. $\dashv$

Definition 6.5. Define $\approx$ to be the least equivalence relation between $\mathcal{L}_\lambda$-terms closed under:

$$(\alpha)\ \frac{}{\lambda x.M \approx \lambda y.M[y/x]}\ [y \notin FV(M) \cup BV(M)]$$

$$(\beta)\ \frac{}{(\lambda x.M)N \approx M[N/x]}\ [BV(M) \cap FV(N) = \emptyset]$$

$$(\eta)\ \frac{}{\lambda x.Mx \approx M} \qquad (\xi)\ \frac{M \approx N}{\lambda x.M \approx \lambda x.N}$$

$$(\text{Cong})\ \frac{M \approx M' \qquad N \approx N'}{M(N) \approx M'(N')}$$

In the (η) rule we also assume $x \notin FV(M)$.

The following proposition guarantees that equivalent terms are assigned the same meaning.

Proposition 6.6. If $M \approx N$, then for all $\mathbb{S}$ and ϕ, $\pi_{\sigma_1,\tau}[\![M]\!]_\phi = \pi_{\sigma_2,\tau}[\![N]\!]_\phi$, where $\tau = \sigma_1 \vee \sigma_2$.

We say that a term M is in *normal form* if it has no β- or η-redexes, those defined in the usual way.

7 Term Structures

In this section, we outline a method to define a prestructure from a preorder on terms of the language. Given a term M, we denote its $\approx$-equivalence class by $\langle M \rangle$. When we define a function ι on the $\approx$-equivalence classes, we generally write $\iota\langle M \rangle$ rather than $\iota(\langle M \rangle)$. Let

$$\begin{aligned} L_\tau &= \{\langle M \rangle : M \text{ is an } \mathcal{L}_\lambda\text{-term of type } \tau\} \\ T_\tau &= \bigcup\{L_\sigma : \sigma \preceq \tau\} \end{aligned}$$

We have *inclusion maps* $i_{\sigma,\tau} : T_\sigma \to T_\tau$.

Proposition 7.1. The family $i_{\sigma,\tau}$ has the following functoriality properties: $i_{\sigma,\sigma}$ is the identity on T_σ, and if $\sigma \preceq \tau \preceq \mu$, then $i_{\sigma,\mu} = i_{\tau,\mu} \circ i_{\sigma,\tau}$.

Definition 7.2. A *term structure* $\mathbb{T}$ is a family $\{T_\tau, \sqsubseteq_\tau\}$ of preorders, subject to the following:

1. If $\langle M\rangle \in T_{\sigma \xrightarrow{+} \tau}$ and $\langle N\rangle \sqsubseteq_\sigma \langle O\rangle$, then $\langle M(N)\rangle \sqsubseteq_\tau \langle M(O)\rangle$.

2. If $\langle M\rangle \in T_{\sigma \xrightarrow{-} \tau}$ and $\langle N\rangle \sqsubseteq_\sigma \langle O\rangle$, then $\langle M(O)\rangle \sqsubseteq_\tau \langle M(N)\rangle$.

3. $\langle M\rangle \sqsubseteq_{\sigma \xrightarrow{m} \tau} \langle N\rangle$ iff $\langle M(O)\rangle \sqsubseteq_\tau \langle N(O)\rangle$ for all $\langle O\rangle \in T_\sigma$.

Lemma 7.3. For any term structure and any type σ, if $\langle M\rangle \sqsubseteq_\sigma \langle N\rangle$ and $\sigma \preceq \tau$, then also $\langle M\rangle \sqsubseteq_\tau \langle N\rangle$. In other words, the inclusion maps $i_{\sigma,\tau}$ are order-preserving.

Proof. By induction on types. For basic types the order is trivial, so suppose that $\langle M\rangle \sqsubseteq_{\sigma \xrightarrow{m} \tau} \langle N\rangle$, and that $\sigma \xrightarrow{m} \tau \preceq \sigma' \xrightarrow{m'} \tau'$, so that $\sigma' \preceq \sigma$, $\tau \preceq \tau'$, and $m \sqsubseteq m'$. Then:

$$\begin{aligned}
& \langle M\rangle \sqsubseteq_{\sigma \xrightarrow{m} \tau} \langle N\rangle \\
\Leftrightarrow\ & \text{for all } \langle O\rangle \in T_\sigma : \langle M(O)\rangle \sqsubseteq_\tau \langle N(O)\rangle \\
\Rightarrow\ & \text{for all } \langle O\rangle \in T_{\sigma'} : \langle M(O)\rangle \sqsubseteq_\tau \langle N(O)\rangle \\
\Rightarrow\ & \text{for all } \langle O\rangle \in T_{\sigma'} : \langle M(O)\rangle \sqsubseteq_{\tau'} \langle N(O)\rangle \\
\Leftrightarrow\ & \langle M\rangle \sqsubseteq_{\sigma' \xrightarrow{m'} \tau'} \langle N\rangle
\end{aligned}$$

The second implication is because $T_{\sigma'} \subseteq T_\sigma$. The third implication is by induction hypothesis. ⊣

Proposition 7.4. For any term structure $\{\mathbb{T}_\tau\}_{\tau\in\mathcal{T}}$ there is an associated pre-structure $\{\mathbb{D}_\tau\}_{\tau\in\mathcal{T}}$ with order-isomorphisms $\iota_\tau : \mathbb{T}_\tau \to \mathbb{D}_\tau$, such that:

$$\iota_{\sigma \xrightarrow{m} \tau}\langle M\rangle\big(\pi_{\sigma',\sigma}(\iota_{\sigma'}\langle N\rangle)\big) = \iota_\tau\langle M(N)\rangle. \quad (3)$$

Proof. We build preorders $\{\mathbb{D}_\tau\}_{\tau\in\mathcal{T}}$ and order-isomorphisms $\iota_\tau : \mathbb{T}_\tau \to \mathbb{D}_\tau$ using recursion on the set of types. For base types $b \in \mathcal{B}$ we simply take $\mathbb{D}_b = \mathbb{T}_b$, and ι_b is the identity.

Suppose we have already defined $\mathbb{D}_\sigma$ and $\mathbb{D}_\tau$, and we have isomorphisms $\iota_\sigma : \mathbb{T}_\sigma \to \mathbb{D}_\sigma$ and $\iota_\tau : \mathbb{T}_\tau \to \mathbb{D}_\tau$. For $\mathbb{D}_{\sigma \xrightarrow{m} \tau}$, we use

$$D_{\sigma \xrightarrow{m} \tau} \quad = \quad \{M^* : \langle M\rangle \in T_{\sigma \xrightarrow{m} \tau}\}$$

where

$$\begin{aligned}
& M^*\big(\iota_\sigma\langle N\rangle\big) && = && \iota_\tau\langle M(N)\rangle \\
& M^* \leq_{\sigma \xrightarrow{m} \tau} N^* && \text{iff} && M \sqsubseteq_{\sigma \xrightarrow{m} \tau} N \text{ in } \mathbb{T}_{\sigma \xrightarrow{m} \tau}
\end{aligned}$$

In other words, we define M^* exactly so that (3) is satisfied. The map M^* is well-defined because $\approx$ respects term application. The order-embedding $\iota_{\sigma \xrightarrow{m} \tau} : \mathbb{T}_{\sigma \xrightarrow{m} \tau} \to \mathbb{D}_{\sigma \xrightarrow{m} \tau}$ is obviously given by $\iota_{\sigma \xrightarrow{m} \tau}(\langle M\rangle) = M^*$. We show this map is 1-1.

Suppose $\iota_{\sigma \xrightarrow{m} \tau}\langle M\rangle = \iota_{\sigma \xrightarrow{m} \tau}\langle N\rangle$. Choose some variable $v \notin FV(M) \cup FV(N)$. Then by definition of $\iota_{\sigma \xrightarrow{m} \tau}$ we have $\iota_\tau\langle M(v)\rangle = \iota_\tau\langle N(v)\rangle$, which means by induction hypothesis that $\langle M(v)\rangle = \langle N(v)\rangle$, i.e, that $M(v) \approx N(v)$. By rule (ξ) we also have $\lambda v.M(v) \approx \lambda v.N(v)$, and by two applications of (η) and transitivity we have $M \approx N$, whence $\langle M\rangle = \langle N\rangle$.

It remains only to show that $\{\mathbb{D}_\tau\}_{\tau\in\mathcal{T}}$ is a well defined pre-structure with maps $\pi_{\sigma,\tau}$ given by

$$\pi_{\sigma,\tau} = \iota_\tau \circ i_{\sigma,\tau} \circ \iota_\sigma^{-1}.$$

Condition 1 in Definition 3.5 holds trivially. Condition 2 comes from condition 1 on the term structure, condition 3 from point 2, condition 4 from point 3, and condition 7 from Lemma 7.3. The functoriality properties 5 and 6 come from Proposition 7.1. ⊣

Lemma 7.5 is the main construction of semantic models for our calculus besides the full structures which we saw in Definition 3.6 and Example 5.3. In it, note that if ψ is an assignment function (a map from variables to terms), then composing with the natural map (taking terms to $\approx$-classes) gives a map into the term structure. So further composing with ι gives a valuation into a pre-structure. We thus define $\langle\psi\rangle$ to be the valuation function given by $\langle\psi\rangle(x) = \iota\langle\psi(x)\rangle$ for all x. What is more, every valuation function into a model of this type is of the form $\langle\psi\rangle$, and ψ is determined uniquely up to $\approx$.

Lemma 7.5. Let $\mathbb{T}$ be a term structure, and let $\mathbb{D}$ be its associated pre-structure from Proposition 7.4. Define an interpretation function in $\mathbb{D}$:

$$[\![M]\!]_{\langle\psi\rangle} \quad = \quad \iota\langle M[\psi]\rangle, \quad (4)$$

for all terms M and term substitutions ψ. Let $\mathbb{S} = (\mathbb{D}, [\![\ ]\!])$. Then $\mathbb{S}$ is a structure.

Proof. We check the two requirements on the interpretation function. The first requirement concerns applications. Let $M : \sigma \xrightarrow{m} \tau$, and $N : \sigma' \preceq \sigma$. Fix a substitution ψ. Then

$$\begin{aligned}
& [\![M(N)]\!]_{\langle\psi\rangle} && \\
=\ & \iota\langle M(N)[\psi]\rangle && \text{by (4)} \\
=\ & \iota\langle M[\psi](N[\psi])\rangle && \text{by def. of } [\psi] \\
=\ & \iota\langle M[\psi]\rangle(\pi_{\sigma',\sigma}\langle\iota(N[\psi])\rangle) && \text{by (3)} \\
=\ & [\![M]\!]_\psi\big(\pi_{\sigma',\sigma}[\![N]\!]_\psi\big) && \text{by (4), twice}
\end{aligned}$$

Finally, consider a term $\lambda x.M$, and fix a substitution ψ. Let $a \in D_\sigma$, and let A be a term such that $\iota\langle A\rangle = a$. Let y be a variable which is not free in A, and also not free in M or any $\psi(z)$ for z a free variable of M. Then

$$\begin{array}{lll} & (\iota\langle(\lambda x.M)[\psi]\rangle)(a) & \\ = & (\iota\langle\lambda y.M[\psi^y_x]\rangle)(\iota\langle A\rangle) & \text{by def. of } A \text{ and } y \\ = & (\lambda y.M[\psi^y_x])^*(\iota\langle A\rangle) & \text{by definition of } \iota \\ = & \iota((\lambda y.M[\psi^y_x])(A)) & \text{by definition of } {}^* \\ = & \iota(M[\psi^y_x][\mathrm{id}^A_y]) & \text{by } \beta\text{-equivalence} \\ = & \iota\langle M[\psi^A_x]\rangle & \text{by choice of } y \\ = & [\![M]\!]_{\langle\psi^A_x\rangle} & \text{by the def. in (4)} \\ = & [\![M]\!]_{\langle\psi\rangle^a_x} & \text{because } \iota\langle A\rangle = a \\ = & [\![\lambda x.M]\!]_{\langle\psi\rangle}(a) & \text{semantics of } \lambda x.M \end{array}$$

This completes the proof. ⊣

8 Monotonicity Entails Positivity; Antitonicity Entails Negativity

The main result of this section is Theorem 8.2, a converse (of sorts) to Lemma 5.2.

8.1 A Term Structure Built "Freely" from an Inequality

The proof of Theorem 8.2 employs a specific term structure. Let σ be a type, and let x, y, and z be distinct variables of type σ. We take $\mathbb{T} = \mathbb{T}(x, y, z)$ to be the term structure obtained by defining for each type ρ, $\langle P\rangle \sqsubseteq_\rho \langle Q\rangle$ if and only if the following holds: P and Q are in normal form, there is a term S with no occurrences of y or z, and there are pairwise disjoint sets of occurrences A, B, Y, and Z of x in S such that all occurrences in A are positive, all occurrences in B are negative, and

$$\begin{array}{rcl} P & = & S[y/x_A, z/x_B, y/x_Y, z/x_Z] \\ & = & S[y/x_{A\cup Y}, z_{B\cup Z}] \\ Q & = & S[z/x_A, y/x_B, y/x_Y, z/x_Z] \\ & = & S[y/x_{B\cup Y}, z_{A\cup Z}] \end{array} \tag{5}$$

In other words, if $\langle P\rangle \sqsubseteq_\rho \langle Q\rangle$, then we can obtain Q from P, assuming these are in normal form, by "increasing" some positive occurrences y to z (those occurrences in A) and "decreasing" some negative occurrences of z to y (those in B). The sets Y and Z are needed in order to make the whole construction work. More specifically, it follows from (5) that

$$P[x/y, x/z] \;=\; S \;=\; Q[x/y, x/z].$$

Lemma 8.1. $\mathbb{T}$ is a term structure.

Theorem 8.2 (Lyndon Theorem)**.** Suppose $M : \tau$ is a typed term in normal form, and let $x : \sigma$ be a variable. Then the following are equivalent:

a. All free occurrences of x in M are $+$ ($-$).

b. For all structures $\mathbb{S} = (\mathbb{D}, [\![\ ]\!])$ and assignments ϕ, $a \mapsto [\![M]\!]_{\phi^a_x}$ is monotone (antitone).

Proof. The (a) $\Rightarrow$ (b) directions follow from Lemma 5.2. We show (b) $\Rightarrow$ (a). We only argue that "monotone implies positive", as the argument that "antitone implies negative" is similar. Let $M : \tau$ be a term, and suppose $y : \sigma$ and $z : \sigma$ are distinct variables not appearing in M.

Fix a normal form M and a variable $x : \sigma$ that occurs freely in it. Take $\mathbb{T}$ to be the term structure $\mathbb{T}(x, y, z)$ studied in Lemma 8.1. Take ϕ to be the assignment generated by the identity substitution, $\langle\phi\rangle(w) = \langle \mathrm{id}(w)\rangle = \langle w\rangle$.

Let $(\mathbb{D}, [\![\ ]\!])$ be the structure obtained from $\mathbb{T}$ using Lemma 7.5. We apply (1b) to this structure. By monotonicity of ι, $\iota\langle y\rangle \leq_\sigma \iota\langle z\rangle$ in $\mathbb{D}$. We thus see that in $\mathbb{D}_\tau$,

$$\begin{array}{rll} & \iota\langle M[y/x]\rangle & \\ = & [\![M]\!]_{\langle\phi^y_x\rangle} & \text{by (4)} \\ = & [\![M]\!]_{\langle\phi\rangle^{\iota\langle y\rangle}_x} & \text{since } \langle\phi^y_x\rangle(x) = \iota\langle y\rangle \\ \leq_\tau & [\![M]\!]_{\langle\phi\rangle^{\iota\langle z\rangle}_x} & \text{by hypothesis on } M \\ = & [\![M]\!]_{\langle\phi^z_x\rangle} & \text{since } \langle\phi^z_x\rangle(x) = \iota\langle z\rangle \\ = & \iota\langle M[z/x]\rangle & \text{by (4)} \end{array}$$

Since ι reflects order, $\langle M[y/x]\rangle \sqsubseteq_\tau \langle M[z/x]\rangle$.

Notice that $M[y/x]$ and $M[z/x]$ are β-normal forms, since M is a β-normal form. By definition of the order in $\mathbb{T}$, there is a term S and sets of free occurrences of x in S, say A, B, Y, and Z, such that all occurrences in A are positive, all occurrences in B are negative, and

$$\begin{array}{rcll} S & = & M[y/x][x/y, x/z] & \\ & = & M[z/x][x/y, x/z] & \\ M[y/x] & = & S[y/x_{A\cup Y}, z/x_{B\cup Z}] & (6) \\ M[z/x] & = & S[y/x_{B\cup Y}, z/x_{A\cup Z}] & (7) \end{array}$$

However, z does not occur in the term on the left of (6), since z does not occur in M. And so $B = Z = \emptyset$. Similarly, y does not occur in the term on the right of (7), so $Y = \emptyset$ also. Thus $M[y/x] = S[y/x_A]$. And as x is not free in $M[y/x]$, it is not free in $S[y/x_A]$. This means that A is the set of

$$\text{(Ref)}\ \frac{}{M \precsim_\sigma M} \qquad \text{(Trans)}\ \frac{M \precsim_\sigma N \qquad N \precsim_\sigma O}{M \precsim_\sigma O} \qquad \text{(Point)}\ \frac{M \precsim_{\sigma \overset{m}{\to} \tau} N}{M(O) \precsim_\tau N(O)}$$

$$\text{(Mono)}\ \frac{N \precsim_\sigma O}{M(N) \precsim_\tau M(O)}\ [M : \sigma \overset{+}{\to} \tau] \qquad \text{(Anti)}\ \frac{N \precsim_\sigma O}{M(O) \precsim_\tau M(N)}\ [M : \sigma \overset{-}{\to} \tau]$$

$$\text{(Pres)}\ \frac{M \precsim_\sigma N}{M \precsim_\tau N}\ \sigma \preceq \tau \qquad \text{(Equiv)}\ \frac{}{M \precsim_\tau N}\ M \approx N \qquad \text{(Func)}\ \frac{M \precsim_\tau N}{\lambda x.M \precsim_{\sigma \overset{m}{\to} \tau} \lambda x.N}$$

Figure 3: Rules of the Monotonicity Calculus. We omit the types on the terms, except for the side conditions in the (Mono) and (Anti) rules.

all free occurrences of x in S. As a result, all free occurrences of x in S are $+$. Furthermore,

$$S \quad = \quad M[y/x][x/y, x/z] \quad = \quad M.$$

(The last equality holds because $M[y/x][x/y, x/z]$ takes M, then changes all free occurrences of x in M to y, and then changes y and z back to x. Since y and z do not occur free in M, this is M itself.) Again, $S = M$. Thus all free occurrences of x in M are $+$, as desired. $\dashv$

9 A Complete Proof System

We come to the centerpiece of this work, the Monotonicity Calculus given by the rules of inference in Figure 3.

Syntax Our setting is similar to equational reasoning in simply typed lambda calculus (Friedman, 1975); however, our calculus deals with *inequality assertions* $M \precsim_\sigma N$. We make such assertions when the types of M and N are both $\preceq \sigma$. We use Γ for a set of inequality assertions. We write $\Gamma \vdash M \precsim_\sigma N$ if there is a proof of $M \precsim_\sigma N$ from Γ, that is, if there is a finite tree with root $M \precsim_\sigma N$, and each node either a leaf from Γ, or an application of one of the rules in Figure 3.

Example 9.1. In Figure 4 we give a derivation using our ongoing example of real functions. The derivation is similar to the one depicted in Figure 1; we only include this one in full for reasons of space. The proof of "$1 - 1 \leq 2 - 0$" uses two basic assumptions: "$0 \leq 1$" and "$x \leq x + 0$ for any x." Note in particular the use of Lemma 9.5. (We could alternatively have assumed $\lambda x.x \precsim \lambda x. + (x)(0)$ in the same way we used the assumption $\texttt{deftly} \precsim \lambda x.x$ in Figure 1.)

Semantics Given a structure $\mathbb{S} = (\mathbb{D}, [\![\ ]\!])$, we write $\mathbb{S} \vDash_\phi M \precsim_\sigma N$ if the following hold:

1. The types of M and N are both $\preceq \sigma$.
2. $\pi_{\sigma_1,\sigma}([\![M]\!]_\phi) \leq \pi_{\sigma_2,\sigma}([\![N]\!]_\phi)$ in $\mathbb{D}_\sigma$.

Frequently we leave off the type σ in assertions $\mathbb{S} \vDash_\phi M \precsim_\sigma N$. We also write $\Gamma \vDash M \precsim N$ if for all structures $\mathbb{S}$ such that $\mathbb{S} \vDash_\phi G \precsim H$ for all $G \precsim H \in \Gamma$ and all assignments ϕ, we also have $\mathbb{S} \vDash_\phi M \precsim N$ for all assignments ϕ.

9.1 Basic Properties of the System

Proposition 9.2. If $\Gamma \vdash M \preceq_\sigma N$, and also $M \approx M'$ and $N \approx N'$, then $\Gamma \vdash M' \preceq_\sigma N'$

The next result is a key fact about our system. It emphasizes the fact that we take open assertions in hypothesis sets Γ to be "universally quantified."

Proposition 9.3. Let $M : \sigma_1 \to \tau_1$ and $N : \sigma_2 \to \tau_2$ be terms, let $m_1, m_2 \sqsubseteq m$, and let σ and τ be types with $\sigma \preceq \sigma_1, \sigma_2$ and $\tau_1, \tau_2 \preceq \tau$. Thus, $\sigma_1 \overset{m_1}{\to} \tau_1, \sigma_2 \overset{m_2}{\to} \tau_2 \preceq \sigma \overset{m}{\to} \tau$. Suppose that for all terms $O : \sigma$, $\Gamma \vdash M(O) \preceq_\tau N(O)$. Then $\Gamma \vdash M \preceq_{\sigma \overset{m}{\to} \tau} N$.

Proof. Let x be a variable of type σ which does not occur in M or N. Our hypotheses tell us that $\Gamma \vdash M(x) \preceq_\tau N(x)$. By (Func), $\Gamma \vdash \lambda x.M(x) \preceq_{\sigma \overset{m}{\to} \tau} \lambda x.N(x)$. By ($\eta$), (Equiv), and (Trans), $\Gamma \vdash M \preceq_\tau N$. $\dashv$

Remark 9.4. We do not know whether Proposition 9.3 holds without (Func). This would be important if one were to revise our meaning of the calculus. Currently $\Gamma \vDash M \precsim N$ means that for all structures $\mathbb{S}$ such that $\mathbb{S} \vDash_\phi G \precsim H$ for all $G \precsim H \in \Gamma$ and all assignments ϕ, we also have $\mathbb{S} \vDash_\phi M \precsim N$ for all assignments ϕ. Suppose we wish to change this to mean: for all $\mathbb{S}$ and ϕ such that $\mathbb{S} \vDash_\phi G \precsim H$ for all $G \precsim H \in \Gamma$, $\mathbb{S} \vDash_\phi M \precsim N$ for the same ϕ. Then our rules (ξ) and (Func) are no longer sound. We conjecture that dropping (ξ) and (Func) results in a complete system with the revised semantic interpretation.

$$\dfrac{\dfrac{\dfrac{x \precsim +(x)(0)}{1 \precsim +(1)(0)}\ \text{Lemma 9.5} \quad \dfrac{0 \precsim 1}{+(1)(0) \precsim +(1)(1)}\ \text{(Mono)}}{1 \precsim +(1)(0)}\ \text{(Trans)} \quad \dfrac{\dfrac{1 \precsim 2}{-(1) \precsim -(2)}\ \text{(Mono)}}{-(1)(1) \precsim -(2)(1)}\ \text{(Point)} \quad \dfrac{0 \precsim 1}{-(2)(1) \precsim -(2)(0)}\ \text{(Anti)}}{-(1)(1) \precsim -(2)(0)}\ \text{(Trans)}$$

Figure 4: Example of elementary monotonicity reasoning with real numbers and functions.

Lemma 9.5. For all terms M and N, and all term substitutions ψ,

$$M \precsim N \vdash M[\psi] \precsim N[\psi].$$

Proof. Let $x_1, \ldots x_n$ be the free variables of M and N. By repeated use of (Func), we have

$$M \precsim N \vdash \lambda x_1 \ldots \lambda x_n.M \precsim \lambda x_1 \ldots \lambda x_n.N$$

Now, because of (Equiv) and α equivalence, we may change each x_i into a variable y_i with $y_i \notin FV(\psi(x_j))$ for each $i, j = 1, \ldots n$. Then by repeated use of the (Point), we have

$$\begin{array}{rcl} M \precsim N & \vdash & (\lambda y_1 \ldots \lambda y_n.M')\psi(x_1) \ldots \psi(x_n) \\ & \precsim & (\lambda y_1 \ldots \lambda y_n.N')\psi(x_1) \ldots \psi(x_n) \end{array}$$

where

$$\begin{array}{rcl} M' & = & M[y_1/x_1] \ldots [y_n/x_n] \\ N' & = & N[y_1/x_1] \ldots [y_n/x_n] \end{array}$$

Then by repeated use of (Equiv) and β reductions:

$$\begin{array}{rcl} M \precsim N & \vdash & M'[\psi(x_1)/y_1] \ldots [\psi(x_n)/y_n] \precsim \\ & & N'[\psi(x_1)/y_1] \ldots [\psi(x_n)/y_n] \end{array}$$

Now because each y_i does not appear in $\psi(x_j)$ for $j \leq i$, then these substitutions can be done simultaneously, i.e.,

$$\begin{array}{l} M'[\psi(x_1)/y_1] \ldots [\psi(x_n)/y_n] \\ \qquad = M'[\psi(x_1)/y_1, \ldots, \psi(x_n)/y_n] \\ N'[\psi(x_1)/y_1] \ldots [\psi(x_n)/y_n] \\ \qquad = N'[\psi(x_1)/y_1, \ldots, \psi(x_n)/y_n] \end{array}$$

And since

$$\begin{array}{l} M'[\psi(x_1)/y_1, \ldots, \psi(x_n)/y_n] \\ \qquad = M[\psi(x_1)/x_1, \ldots, \psi(x_n)/x_n] \\ N'[\psi(x_1)/y_1, \ldots, \psi(x_n)/y_n] \\ \qquad = N[\psi(x_1)/x_1, \ldots, \psi(x_n)/x_n] \end{array}$$

then $M \precsim N \vdash M[\psi(x_1)/x_1, \ldots, \psi(x_n)/x_n] \precsim N[\psi(x_1)/x_1, \ldots \psi(x_n)/x_n]$, or in other words, we have $M \precsim N \vdash M[\psi] \precsim N[\psi]$. ⊣

Lemma 9.6. If $\Gamma \vdash M \precsim N$, then for all term substitutions ψ, $\Gamma \vdash M[\psi] \precsim N[\psi]$.

Proof. This is an easy induction on the proof system whose base case is Lemma 9.5. ⊣

Theorem 9.7 (Completeness of the Monotonicity Calculus)**.** $\Gamma \vdash M \precsim_\sigma N$ iff $\Gamma \vDash M \precsim_\sigma N$.

9.2 Soundness

We fix a structure $\mathbb{S}$ and a valuation ϕ making all sentences in Γ true in $\mathbb{S}$. We show by induction on derivations from Γ that if $\Gamma \vdash M \precsim_\sigma N$, with the types of M and N being $\sigma_1 \preceq \sigma$ and $\sigma_2 \preceq \sigma$, then $\pi_{\sigma_1,\sigma}([\![M]\!]_\phi) \leq \pi_{\sigma_2,\sigma}([\![N]\!]_\phi)$ in $\mathbb{S}_\sigma$. We use the properties of pre-structures in Definition 3.5.

The most basic derivations from Γ are the elements of Γ itself. This case is trivial.

We begin with (Pres). Assume we have $\Gamma \vdash M \precsim_\tau N$ via proof whose last line is an application of (Pres), with $\sigma \preceq \tau$. So $\Gamma \vdash M \precsim_\sigma N$. Our induction hypothesis tells us, $\mathbb{S} \models_\phi M \precsim_\sigma N_\phi$. Suppose that the types of M and N are σ_1 and σ_2, respectively. Let us write m for $[\![M]\!]_\phi$ and n for $[\![N]\!]_\phi$. Then we have $\pi_{\sigma_1,\sigma}m \leq \pi_{\sigma_2,\sigma}n$ in $\mathbb{D}_\sigma$. And now we calculate easily that $\pi_{\sigma_1,\tau}(m) \leq \pi_{\sigma_1,\tau}(n)$. As a result, $\mathbb{S} \models_\phi M \precsim_\tau N$.

For (Ref), we use the fact that each relation $\sqsubseteq_\sigma$ is reflexive. We omit the easy details on (Trans).

In the rest of this proof, we shall deal with assertions $\Gamma \vdash M \precsim N$ without any notation for the overall type; that is, we shall assume that the types of M and N are exactly σ.

For (Point), assume that $[\![M]\!]_\phi \leq_\sigma [\![N]\!]_\phi$ in $\mathbb{D}_\sigma$. Then by the "pointwise property" (Definition 3.5 part 4) of $\mathbb{S}$,

$$\begin{array}{rcl} & & [\![M(O)]\!]_\phi \\ & = & [\![M]\!]_\phi([\![O]\!]_\phi) \\ & \leq_\sigma & [\![N]\!]_\phi([\![O]\!]_\phi) \\ & = & [\![N(O)]\!]_\phi. \end{array}$$

For (Mono), let $M : \sigma \xrightarrow{+} \tau$. Assume that $[\![N]\!]_\phi \leq_\sigma [\![O]\!]_\phi$ in $\mathbb{D}_\sigma$. By Lemma 5.2,

$[\![M]\!]$ is a monotone function $\mathbb{D}_\sigma \to \mathbb{D}_\tau$. Hence $[\![M(N)]\!]_\phi \leq_\sigma [\![M(O)]\!]_\phi$. The (Anti) rule is treated similarly. The soundness of the (Equiv) rule follows easily from Proposition 6.6.

9.3 Completeness

We next show that the monotonicity calculus is complete. Assume that $\Gamma \models M^\star \precsim_\sigma N^\star$, with the types of $M^\star$ and $N^\star$ being σ_1 and σ_2. We shall show that $\Gamma \vdash M^\star \precsim_\sigma N^\star$ using a term structure $\mathbb{T}$ whose associated structure $\mathbb{S}$ from Lemma 7.5 is called the *canonical model of* Γ. We recall Definition 7.2 and the notation there, especially the fact that each T_σ is the set of $\approx$-equivalence classes of terms of type $\preceq \sigma$. We order T_σ by

$$\langle M\rangle \sqsubseteq_\sigma \langle N\rangle \quad \text{iff} \quad \Gamma \vdash M \precsim_\sigma N.$$

This relation is well-defined by Proposition 9.2.

Claim 9.8. $\mathbb{T}$ is a term structure.

Proof. Let $\langle M\rangle \in T_{\sigma \overset{+}{\to} \tau}$ and $\langle N\rangle \sqsubseteq_\sigma \langle O\rangle$. Then $M : \sigma' \overset{+}{\to} \tau'$, for some types $\sigma \preceq \sigma'$ and $\tau' \preceq \tau$. We thus have a derivation from Γ:

$$\dfrac{\dfrac{\dfrac{\vdots}{N \precsim_\sigma O}\,(\text{Prec})}{N \precsim_{\sigma'} O}\,(\text{Mono})}{M(N) \precsim_{\tau'} M(O)}\,(\text{Prec})}{M(N) \precsim_\tau M(O)}$$

Thus, $\langle M(N)\rangle \sqsubseteq_\tau \langle M(O)\rangle$. For a similar reason, Property 2 also holds, only (Anti) is used instead. One direction of Property 3 is easy, using (Point), and so we omit it. For the reverse direction, suppose $\langle M\rangle(\langle O\rangle) \sqsubseteq_\tau \langle N\rangle(\langle O\rangle)$ for all $\langle O\rangle \in T_\sigma$. Using Proposition 9.3, we see that $\Gamma \vdash M \precsim_{\sigma \overset{m}{\to} \tau} N$. ⊣

Claim 9.8 proved, we appeal to Proposition 7.4 and Lemma 7.5 to obtain an associated structure which we call $\mathbb{S}$. As we know, for any assignment ϕ on $\mathbb{D}$, there is a substitution ψ such that $\langle\psi\rangle = \phi$. To find such a ψ, note that for every variable x, $\phi(x) = \iota(\langle A\rangle)$ for some term A, so we can define $\psi(x) = A$. Then $\langle\psi\rangle(x) = \iota(\langle\psi(x)\rangle) = \iota(\langle A\rangle)$.

Lemma 9.9. Let $G : \sigma_1$ and $H : \sigma_2$, with $\sigma_1, \sigma_2 \preceq \sigma$. If $\Gamma \vdash G \precsim_\sigma H$, then $\mathbb{S} \models G \precsim_\sigma H$. In other words, for every assignment ϕ,

$$\pi_{\sigma_1,\sigma}([\![G]\!]_\phi) \leq_\sigma \pi_{\sigma_2,\sigma}([\![H]\!]_\phi).$$

In particular, for every assertion $G \precsim_\sigma H$ in Γ, $\mathbb{S} \models G \precsim_\sigma H$.

Proof. Suppose $\Gamma \vdash G \precsim_\sigma H$, and fix an assignment ϕ, and let ψ be a substitution such that $\langle\psi\rangle = \phi$. By Lemma 7.5,

$$[\![G]\!]_\phi = [\![G]\!]_{\langle\psi\rangle} = \iota(\langle G[\psi]\rangle), \qquad \text{and}$$
$$[\![H]\!]_\phi = [\![H]\!]_{\langle\psi\rangle} = \iota(\langle H[\psi]\rangle).$$

As $\pi_{\sigma_1,\sigma}$, $\pi_{\sigma_2,\sigma}$, and ι are all order preserving, to prove our lemma we only need to show that $\langle G[\psi]\rangle \sqsubseteq_\sigma \langle H[\psi]\rangle$: in other words, $\Gamma \vdash G[\psi] \precsim_\sigma H[\psi]$. And this was shown in Lemma 9.6. ⊣

We conclude the proof of completeness. We started with $\Gamma \models M^\star \precsim N^\star$, and now we show that $\Gamma \vdash M^\star \precsim N^\star$. By Lemma 9.9, the canonical model $\mathbb{S}$ satisfies Γ under all assignments. We apply this with the assignment ϕ given by $\phi(x) = \langle x\rangle$. Therefore $\mathbb{S} \models_\phi M^\star \preceq_\sigma N^\star$. So $\langle M^\star[\phi]\rangle \sqsubseteq_\sigma \langle N^\star[\phi]\rangle$. However, because $\langle M^\star[\phi]\rangle = \langle M^\star\rangle$ and $\langle N^\star[\phi]\rangle = \langle N^\star\rangle$, we see from the ordering on $\mathbb{S}$ that $\Gamma \vdash M^\star \precsim_\sigma N^\star$.

10 Conclusion

We have presented a calculus extending the simply typed lambda calculus with enough order-theoretic infrastructure to represent arguments about increasing and decreasing functions. The calculus provides a mathematical foundation for a style of monotonicity reasoning that is often implicit in practical NLP work (e.g., MacCartney and Manning 2007; Angeli and Manning 2014; Angeli et al. 2016). Typically this research draws upon lexical resources such as WordNet and entailment relations learned from data, and uses these assumptions as input for proof search over derivations similar to those we considered here. Functional expressions will be marked as monotone or antitone either "by hand" or in an automated way.

In addition to formalizing the proof procedures used in existing work, we believe the present study also suggests further possibilities for applied work. For instance, we have shown how more complex entailment assumptions can be stated and used in a flexible way, e.g., allowing comparison between functions of different polarity types (recall the example in Section 2).

From a technical point of view, our completeness result is analogous to a standard result for simply typed lambda calculus due to Friedman (1975). While our setting is considerably more complex, there still remain open issues here that were settled in the simpler setting: e.g., how to obtain completeness for "full" structures (Ex. 5.3). We leave such questions for future work.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *EMNLP*. pages 2492–2502.

Gabor Angeli and Christopher Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *EMNLP*. pages 534–545.

Gabor Angeli, Neha Nayak, and Christopher Manning. 2016. Combining natural logic and shallow reasoning for question answering. In *ACL*. pages 442–452.

Raffaella Bernardi. 2002. *Reasoning with Polarity in Categorial Type Logic*. Ph.D. thesis, University of Utrecht.

Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. 2015. Learning distributed word representations for natural logic reasoning. In *AAAI Spring Symposium on Knowledge Representation and Reasoning*. pages 10–13.

Harvey Friedman. 1975. Equality between functionals. In Rohit Parikh, editor, *Proceedings of Logic Colloquium '73*. volume 53 of *Lecture Notes in Mathematics*, pages 22–37.

Bart Geurts. 2003. Reasoning with quantifiers. *Cognition* 86(3):223–251.

Bart Geurts and Frans van der Slik. 2005. Monotonicity and processing load. *Journal of Semantics* 22:97–117.

Thomas F. Icard. 2012. Inclusion and exclusion in natural language. *Studia Logica* 100(4):705–725.

Thomas F. Icard and Lawrence S. Moss. 2013. A complete calculus of monotone and antitone higher-order functions. In *Proceedings, Topology, Algebra, and Categories in Logic 2013*, Vanderbilt University, volume 23 of *EPiC Series*, pages 96–99.

Thomas F. Icard and Lawrence S. Moss. 2014. Recent progress on monotonicity. *Linguistic Issues in Language Technology* 9(7):167–194.

Hans Kamp and Barbara Hall Partee. 1995. Prototype theory and compositionality. *Cognition* 57(2):129–191.

Roger C. Lyndon. 1959. An interpolation theorem in the predicate calculus. *Pacific Journal of Mathematics* 9(1):129–142.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *ACL Workshop on Textual Entailment and Paraphrasing*. pages 193–200.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *IWCS-8, Proceedings of the Eighth International Conference on Computational Semantics*. pages 140–156.

Lawrence S. Moss. 2012. The soundness of internalized polarity marking. *Studia Logica* 100(4):683–704.

Reinhard Muskens. 2010. An analytic tableau system for natural logic. In Maria Aloni, Haarold Bastiaanse, Tikitu de Jager, and Katrin Schulz, editors, *Logic, Language and Meaning*, Lecture Notes in Computer Science, volume 6042, pages 104–113.

Victor Sánchez-Valencia. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, Universiteit van Amsterdam.

Allen Stoughton. 1988. Substitution revisited. *Theoretical Computer Science* 59(3):317–325.

William Tune. 2016. *A Lambda Calculus for Monotonicity Reasoning*. Ph.D. thesis, Indiana University.

Johan van Benthem. 1986. *Essays in Logical Semantics*. Reidel, Dordrecht.

Jan van Eijck. 2007. Natural logic for natural language. In Balder ten Cate and Henk Zeevat, editors, *6th International Tbilisi Symposium on Logic, Language, and Computation*. Springer, pages 216–230.

A. Zamansky, N. Francez, and Y. Winter. 2006. A 'natural logic' inference system using the Lambek calculus. *Journal of Logic, Language, and Information* 15(3):273–295.