Natural Logic for Natural Language

Jan van Eijck[1]

CWI, Amsterdam and Uil-OTS, Utrecht University
jve@cwi.nl,
WWW home page: http://www.cwi.nl/~jve

Abstract. For a cognitive account of reasoning it is useful to factor out the syntactic aspect — the aspect that has to do with pattern matching and simple substitution — from the rest. The calculus of monotonicity, alias the calculus of natural logic, does precisely this, for it is a calculus of appropriate substitutions at marked positions in syntactic structures. We first introduce the semantic and the syntactic sides of monotonicity reasoning or 'natural logic', and propose an improvement to the syntactic monotonicity calculus, in the form of an improved algorithm for monotonicity marking. Next, we focus on the role of monotonicity in syllogistic reasoning. In particular, we show how the syllogistic inference rules (for traditional syllogistics, but also for a broader class of quantifiers) can be decomposed in a monotonicity component, an argument swap component, and an existential import component. Finally, we connect the decomposition of syllogistics to the doctrine of distribution.

1 Introduction

To develop a cognitive account of reasoning, a promising approach is to factor out the syntactic aspect — the aspect that has to do with pattern matching on syntactic structures — from the rest. An obvious candidate for this that has been around for some time now is the so-called calculus of monotonicity. This calculus has a semantic side and a syntactic side. The semantic foundation of monotonicity reasoning is a generalization of the notion of logical consequence to arbitrary types, by defining partial orderings $\Longrightarrow$ on all types (not just the type of sentences, but also those of verb phrases, predicates, adjectives, quantifiers, and so on). In terms of this, one can define what it means to be an order-preserving or an order reversing function from type α to type β. Order preserving functions are the functions f that are such that if $x \Longrightarrow y$ then $f(x) \Longrightarrow f(y)$. Order reversing functions are the functions f that are such that if $x \Longrightarrow y$ then $f(y) \Longrightarrow f(x)$.

The syntactic side of the calculus of monotonicity is the marking of monotonicity of syntactic components in a syntactic structure. Let S be a syntactic structure, and let A be a component of that structure. Suppose that A has type α and S has type β. Consider the syntactic function F that consists of replacing component A by other suitable components of type α. In other words, consider the function $F = \lambda Y.S[Y/A]$. Then the semantic counterpart of F is a function f of type $\alpha \to \beta$. Soundness and completeness of a monotonicity calculus have to do with the relation between F and f.

A monotonicity marking algorithm is *sound* if the following holds: if A is marked $+$ in S, then the function that interprets $\lambda Y.S[Y/A]$ is monotonicity preserving, and if A is marked $-$ in S, then the function that interprets $\lambda Y.S[Y/A]$ is monotonicity reversing.

A monotonicity marking algorithm is *complete* if the following holds: if the function that interprets $\lambda Y.S[Y/A]$ is monotonicity preserving then A is marked $+$ in S, and if the function that interprets $\lambda Y.S[Y/A]$ is monotonicity reversing, then A is marked $-$ in S.

Explanations of aspects of the human reasoning faculty must be based on hypotheses about calculating mechanisms. Monotonicity calculi have been proposed time and again in the literature as candidates for such mechanisms, by philosophers [23, 11], logicians [5, 22], computer scientists [21], linguists [8], and most recently by cognitive scientists [15], with less or more explicit suggestions to use them as a basis for generating hypotheses about processing load in human reasoning. The catch phrases for this enterprise used to be 'natural logic' or 'logic for natural language', for the logic that was meant to provide an account of the way human reasoners actually reason.

The structure of the paper is as follows. First we review the semantic side of monotonicity reasoning. Next, we look at the syntactic side, and propose an improvement of existing algorithms for monotonicity marking. Then, as a first step in developing a cognitive perspective on reasoning, we look at syllogistic reasoning and slight extensions of it, under the aspect of monotonicity. We show how the syllogistic reasoning rules can be decomposed in a monotonicity component (a monotonicity rule), a rule for argument swapping in symmetric quantifiers (a symmetry rule), and a rule for invoking the existential force of the syllogistic quantifiers (an existential import rule). The paper winds up by linking the monotonicity part of syllogistics to the doctrine of distribution.

2 Semantics of monotonicity

Just as we can say that 'Gaia is smiling' logically implies 'Gaia is smiling or Gaia is crying', we would like to say that 'smiling' logically involves 'smiling or crying', or that 'dancing' logically implies 'moving', but also that 'at least three' logically implies 'at least two', and so on.

'Gaia is smiling' is a sentence, 'smiling' is a predicate, 'at least three' is a quantifier. We know what entailment means for sentences. One sentence entails another if whenever the first one is true the second one is. The obvious way to lift this notion to predicates is by stipulating that one predicate entails another if it holds for every subject that the sentence one gets by combining that subject with the first predicate entails the sentence one gets by combining the subject with the second predicate. Similarly for quantifiers. To get a sentence from 'at least three', one has to combine the quantifier with a noun and a verb. Since indeed it holds for every noun N and verb V that 'at least three N V' entails 'at least two N V', we can say that 'at least three' entails 'at least two'.

This idea of lifting entailment from the category of sentences to arbitrary categories was made fully precise by Van Benthem in [5]. The semantics of monotonicity from [5], given in terms of partial orders on arbitrary semantic domains (supposed to correspond to various syntactic categories), effectively extends the notion of logical entailment from the level of sentences to that of verb phrases, quantifiers, noun phrases, adjectives, and so on.

Van Benthem starts out from the basic types t (truth values, the type of sentences), and e (entities, the type of proper names). Complex types are defined by recursion, as follows: (i) e and t are types. (ii) if α and β are types, then $\alpha \to \beta$ is a type. The entailment relation is defined as follows (we use $E :: \alpha$ for "syntactic expression E has semantic type α):

- If $E, E' :: e$ then $I(E) \Longrightarrow I(E') := I(E) = I(E')$.
- If $E, E' :: t$ then $I(E) \Longrightarrow I(E') := I(E) \leq I(E')$.
- If $E, E' :: \alpha \to \beta$ then $I(E) \Longrightarrow I(E')$ iff

$$\text{for all } x \in D_\alpha, I(E)(x) \Longrightarrow I(E')(x).$$

Here $I(E)$ denotes the interpretation of E, and D_α is used for the domain of objects of type α. If $E :: \alpha$ then $I(E) \in D_\alpha$, i.e., the interpretation of E is an object in D_α, the domain of objects of type α.

This definition yields results like the following:

$$\text{beautiful and intelligent} \Longrightarrow \text{beautiful}$$

$$\text{cry} \Longrightarrow \text{cry or sulk}$$

$$\text{Mary} \Longrightarrow \text{some woman}$$

$$\text{at most 1} \Longrightarrow \text{at most 2}$$

The 'order calculus' implied by this definition got reinvented in [13].

Theorem 1. *If the domain D_e is finite, then for any type α the relation $\Longrightarrow_\alpha$ is decidable, and the monotonicity properties of any $F : \alpha \to \beta$ are decidable.*

Proof. If D_e is finite, then D_α will be finite for any type α.

To decide whether $f : D_\alpha \to D_\beta$ is order preserving (monotone increasing), check whether $f(x) \Longrightarrow f(y)$ for the finite number of pairs (x, y) with $x \Longrightarrow y$.

To decide whether f is order reversing (monotone decreasing), check whether $f(y) \Longrightarrow f(x)$ for the finite number of pairs (x, y) with $x \Longrightarrow y$. □

The decidability result was for fixed finite sizes of the domain. But note that the result still holds if you put an arbitrary finite treshold on the domain size:

Theorem 2. *For any finite threshold k on the domain size, and for any type α the relation $\Longrightarrow_\alpha^{\leq k}$ ($\Longrightarrow_\alpha$ for all domains up to size k) is decidable, and the monotonicity properties of any $F : \alpha \to \beta$ are decidable.*

Proof. Just check the $\Longrightarrow_\alpha^0, \ldots, \Longrightarrow_\alpha^k$ in turn. □

Below we will study the generalized syllogistic quantifiers based on 'At most K'. If we evaluate these in universes up to some fixed size, all inferences expressed in terms of them are decidable.

3 General Structure of Rules for Monotonicity Reasoning

A monotonicity preserving function F can be represented as a kind of 'mental model' [17], as follows:

$$\frac{X \Longrightarrow Y}{F(X) \Longrightarrow F(Y)}\ F\uparrow$$

Here it is assumed that X, Y are expressions of a logical type α that is partially ordered by $\Longrightarrow$, that $F(X), F(Y)$ are expressions of logical type β that is partially ordered by $\Longrightarrow$, and that F is an order-preserving function of type $\alpha \rightarrow \beta$. One way of reading the rule is as an explication of the fact that F is order preserving (or monotone increasing). Another way of reading the rule is as an inference rule triggered by a function F that is known to be order preserving. $F\uparrow$ expresses that F is order preserving.

The mental model somehow represents the 'transfer' by F of the growth of X to the growth of $F(X)$, with details largely irrelevant. Indeed, the lack of formal detail of the publications in the mental models school seems to indicate that mental models are meant to provide a suggestive *metaphor* of cognitive processing rather than a formal mechanism. The metaphor suggests that when the mental picture of 'uniform growth' is put in reverse, processing load increases:

$$\frac{X \Longrightarrow Y}{F(X) \Longleftarrow F(Y)}\ F\downarrow$$

Again, there are various ways to read this rule. $F\downarrow$ expresses that F is order reversing (or: monotone decreasing).

For an appreciation of the generality of the monotonicity rule, it is illuminating to look at some special cases. If $X, Y, F(X), F(Y)$ all have type t, then $\Longrightarrow$ is logical consequence (or logical implication), and $F(X)$, $F(Y)$ are statements, and we get:

$$\frac{X \Longrightarrow Y \qquad F(X)}{F(Y)}\ F\uparrow$$

An example of this would be: infer from 'Mary dances implies Mary moves' (with 'Mary dances' for X and 'Mary moves' for Y), and 'Mary dances gracefully' (with 'gracefully' for F) that 'Mary moves gracefully'.

$$\frac{X \Longrightarrow Y \qquad F(Y)}{F(X)}\ F\downarrow$$

Reading X and Y as above, and reading F as negation, we get the following example of this rule: infer from An example of this would be: infer from 'Mary dances implies Mary moves' and 'Mary does not move' (with 'does not' for F) that 'Mary does not dance'.

In the case where X, Y are sets (type $e \rightarrow t$) and $F(X), (F(Y)$ are truth values, F has type $(e \rightarrow t) \rightarrow t$ (the type of quantifiers), and we get:

$$\frac{\mathrm{Q}(X) \qquad X \subseteq Y}{\mathrm{Q}(Y)}\ \mathrm{Q}\uparrow$$

For an example, let X stand for 'dancing', Y for 'moving', and Q for 'everyone'. Then the rule says that one may conclude from 'everyone is dancing' and 'dancing involves moving' that 'everyone is moving'.

$$\frac{\mathrm{Q}(Y) \qquad X \subseteq Y}{\mathrm{Q}(X)}\ \mathrm{Q}\downarrow$$

To get an example of this, read X and Y as above, and interpret Q as 'nobody'.

In fact, F may have further internal structure, i.e., $F(X)$ may have the form of binary generalized quantifier $\mathrm{Quant}(X, P)$ or $\mathrm{Quant}(P, X)$. This gives us four possible monotonicity rules for binary quantifiers. Examples of binary quantifiers are *all*, with monotonicity properties $(\downarrow, \uparrow)$, *some*, with $(\uparrow, \uparrow)$, *no*, with $(\downarrow, \downarrow)$, and *most*, with $(\_, \uparrow)$.

$$\frac{\mathrm{Quant}(X, P) \qquad X \subseteq Y}{\mathrm{Quant}(Y, P)}\ \mathrm{Quant}(\uparrow, \_)$$

Example: infer from 'some philosophers are mortal' and 'philosophers are humans' that 'some humans are mortal'.

$$\frac{\mathrm{Quant}(P, X) \qquad X \subseteq Y}{\mathrm{Quant}(P, Y)}\ \mathrm{Quant}(\_, \uparrow)$$

Example: infer from 'most philosophers are human' and 'humans are mortal' that 'most philosophers are mortal'.

$$\frac{\mathrm{Quant}(Y, P) \qquad X \subseteq Y}{\mathrm{Quant}(X, P)}\ \mathrm{Quant}(\downarrow, \_)$$

Example: infer from 'all humans are mortal' and 'philosophers are human' that 'all philosophers are mortal'.

$$\frac{\mathrm{Quant}(P, Y) \qquad X \subseteq Y}{\mathrm{Quant}(P, X)}\ \mathrm{Quant}(\_, \downarrow)$$

Example: infer from 'no philosophers are mortal' and 'humans are mortal' that 'no philosophers are human'.

4 Polarity Marking Revisited

If we can manage to parse a syntactic structure S in some way or other as a monotonicity preserving function F taking an argument A, we can make an inference step, given a suitable trigger. If we can parse S as a monotonicity reversing function F taking an argument A, we can make an inference step,

given a suitable trigger. In an application of a monotonicity rule, one of the premisses is of the form $X \Rightarrow Y$, for some X, Y of the same type. We call this premisse the trigger of the rule. In fact, polarity marking is an enrichment of syntax that can be viewed as a 'shallow' alternative for a translation into logical form.

Existing polarity marking calculi [22, 8, 6] are all based in one way or another on Sanchez's [22] bottom-up algorithm for polarity marking, which needs a separate pass for determining polarity in context. We propose to replace this by a top-down polarity marking algorithm, with the advantage that it takes context into account and computes polarity in a single pass. Here are some comparisons:

- Our approach assigns marking maps as part of the (bottom-up) syntax structure building process and next computes markings top-down.
- Sanchez' algorithm [22] works bottom-up and has three stages: (i) marking argument positions in lexical entries, (ii) propagating the markings to other categories, and (iii) polarity determination of nodes C by counting the number of plusses and minuses on a path from the root to C.
- The approach of Dowty [8] is constraint-based and bottom up. This necessitates multiplication of syntactic categories for items that can occur in both positive and negative positions.
- The approach of [6] is also bottom up. It uses the machinery of multimodal categorial grammar, for which the issue of parsing complexity is still open (no polynomial parsing algorithm is known). Our approach to monotonicity marking avoids the machinery of multimodal categorial grammar.
- The 'order calculus' of [13] is a proof system for $\Longrightarrow$ for a particular natural language fragment. The system does not separate out polarity marking from $\Longrightarrow$ calculation. Because of the fact that for all but the simplest natural language fragments the $\Longrightarrow$ relation has much higher complexity than polarity marking (which can always be done in polynomial time), this is a design flaw.

The three maps on polarity markings that our algorithm employs are (i) preservation, (ii) reversal, and (iii) breaking of polarity. Polarity marking is fully determined by the polarity preserving and reversing properties of the semantic functions involved. Let polarity markings m range over $\{+, -, 0\}$. Instead of explicitly giving the function, in a monotonicity calculus it is enough to give the mappings on polarity markings: preservation (the mapping i), reversal (the mapping r), or breaking of monotonicity (the mapping b), with i the identity map, r the map given by $r(+) = -, r(-) = +, r(0) = r(0)$, and b given by $b(x) = 0$. Using m for the domain $\{+, -, 0\}$, we see that maps on polarity markings have the type $m \to m$.

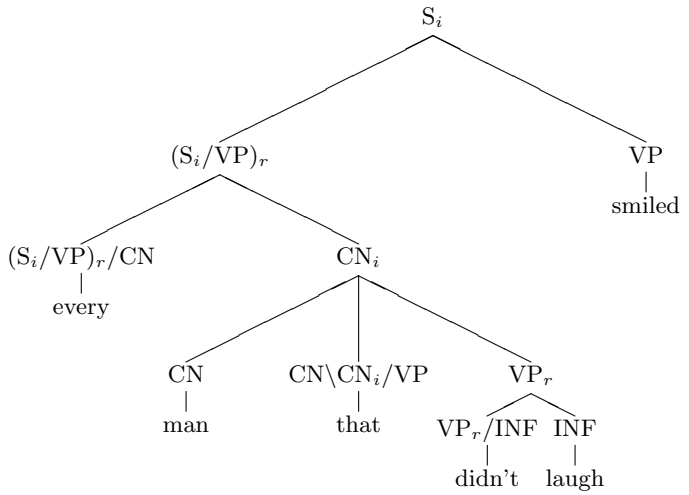

Fig. 1. 'Every man that didn't laugh smiled.'

Polarity Map Assignment

Leaf Marking Functional lexical categories have all their result categories labelled with marker transformers. Lexical argument categories get an unlabelled basic category.

Tree Marking If C consists of a function C_f/A and an argument A, where f is a marker transformer, (or of an argument A and a function $A\backslash C_f$, or of an argument A, a function $A\backslash C_f/B$ and an argument B), then C gets marker transformer f.

In this algorithm, polarity maps are annotations on result categories, as in the following example lexicon.

C(every)	$= (S_i/VP)_r/CN$	C(did)	$= VP_i/INF$
C(some)	$= (S_i/VP)_i/CN$	C(didn't)	$= VP_r/INF$
C(no)	$= (S_r/VP)_r/CN$	C(man)	$= CN$
C(any)	$= (S_i/VP)_i/CN$	C(that)	$= CN\backslash CN_i/VP$
C(the)	$= (S_i/VP)_b/CN$	C(laugh)	$= INF$
C(most)	$= (S_i/VP)_b/CN$	C(laughed)	$= VP$
C(Ann)	$= S_i/VP$	C(kissed)	$= VP_i/(S/VP)$

The category $(S_i/VP)_r/CN$ for 'every' reflects the fact that the semantic function for this quantifier reverses monotonicity direction for its first argument, and keeps the monotonicity direction the same for its second argument.

Syntax trees are built using the familiar categorial grammar construction process, where A/B combines with B to form $[_A A/B\ B]$, $B\backslash A$ combines with B

to form $[_A B\ B\backslash A]$, and $B\backslash A/C$ combines with B and C to form $[_A B\ B\backslash A/C\ C]$. An example is in Figure 1.

The polarity marking algorithm works top-down, using the polarity marking maps at the nodes as guidance for determining the polarity markings of the argument nodes (the function nodes always inherit the marking of their parents, for reasons explained in [5]).

Polarity Marking Algorithm

Root Marking The main structure C to be marked has positive polarity, so it is marked with +.

Component Marking If a structure C has polarity marking k, then:

Leaf Marking If C is a leaf, then done.

Composite Marking If C consists of a function (C/A) and argument A (or an argument A and a function A\C, or an argument A, a function A\C/B and an argument B), then the function gets polarity marking k, and the argument(s) get polarity marking $f(k)$, where f is the polarity marking map at node C.

This algorithm in fact defines a function from syntax trees with polarity marking maps to syntax trees with polarity marking maps and markings. The result of running the algorithm on the example tree is in Figure 2.

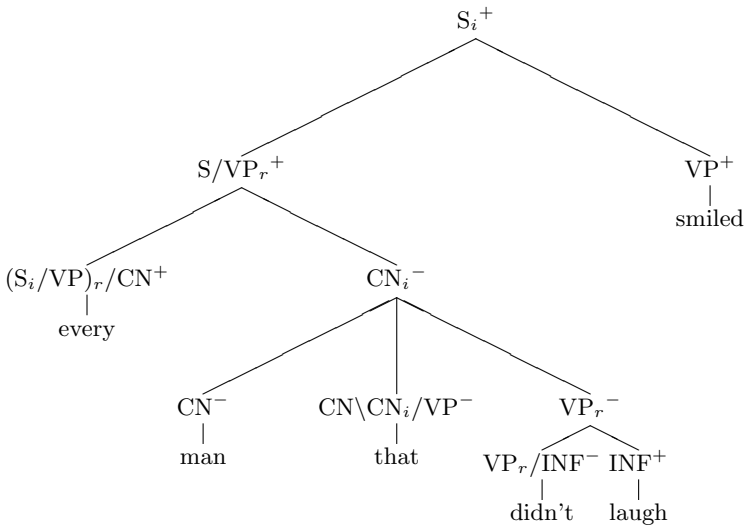

Fig. 2. Marked version of Figure 1.

A monotonicity calculus can be based on the polarity marking algorithm plus information about the mappings on polarity markings of functional lexical

elements and information about the $\Longrightarrow$ ordering, by means of the following rules:

$$\frac{[_S \cdots X^+ \cdots] \quad X \Longrightarrow Y}{[_S \cdots Y^+ \cdots]}$$

$$\frac{[_S \cdots X^- \cdots] \quad Y \Longrightarrow X}{[_S \cdots Y^- \cdots]}$$

Determination of $[_S \cdots X^+ \cdots]$ and $[_S \cdots X^- \cdots]$ is done by an algorithm for parsing plus monotonicity marking; for any reasonable grammar formalism it may be assumed that this can be done in polynomial time. Determination of $\Longrightarrow$ for the category of X and Y is another manner. As is explained in [19], this relation may have a high complexity, even for fairly simple fragments.

But even if $\Longrightarrow$ is hard to compute, partial information about this relation (say, for basic categories), is enough for drawing interesting sound conclusions. Information about the $\Longrightarrow$ ordering for basic categories is supposedly available as basic semantic knowledge of language users. In an implementation, this kind of knowledge can be extracted from semantic databases like Wordnet. The Wordnet [12] *hyperonym* relation encodes the $\Longrightarrow$ relation on the logical type $e \to t$ of nouns. If Wordnet gives the information that 'cat' has hyperonym 'feline', that 'feline' has hyperonym 'mammal', that 'mammal' has hyperonym 'animal', then we can translate this as:

$$\text{cat} \Longrightarrow \text{feline}$$

$$\text{feline} \Longrightarrow \text{mammal}$$

$$\text{mammal} \Longrightarrow \text{animal}$$

This combined with shallow text processing and monotonicity marking allows us to use Wordnet to draw shallow inferences from texts about cats.

5 Monotonicity in Syllogistics

Monotonicity calculi can be viewed as the logical mechanics of syllogistic theory [11, 4, 9]. The cornerstone of syllogistics is the following well-known Square of Opposition:

$$\begin{array}{ccc} \text{All}(\downarrow,\uparrow) & ==== & \text{No}(\downarrow,\downarrow) \\ \big\downarrow & & \big\downarrow \\ \text{Some}(\uparrow,\uparrow) & ==== & \text{NotAll}(\uparrow,\downarrow) \end{array}$$

In set-theoretic notation:

$$\begin{array}{ccc} P \subseteq Q & ==== & P \subseteq \overline{Q} \\ \big\downarrow & & \big\downarrow \\ P \not\subseteq \overline{Q} & ==== & P \not\subseteq Q \end{array}$$

The inferencing that goes on in syllogistics reduces to applications of SYM, EI, and MON, where MON is the monotonicity rule, while SYM and EI are the following rules:

- **SYM or Symmetry** is the rule that infers Quant(Q, P) from Quant(P, Q) for symmetric quantifiers. *Some* and *No* are symmetric.
- **EI or Existential Import** is the principle that every term has a non-empty extension.

$P \subseteq Q$ together with $P \neq \emptyset$ yields $P \cap Q \neq \emptyset$, i.e.: From *All P are Q* it follows by EI that *Some P are Q* $P \subseteq \overline{Q}$ together with $P \neq \emptyset$ yields $P \cap \overline{Q} \neq \emptyset$, i.e.: From *No P are Q* it follows by EI that *Not all P are Q*.

In the context of syllogistics, the **Monotonicity triggers** are the following quantifiers:

- All P Q: $P \subseteq Q$,
- No P Q: $P \subseteq \overline{Q}$, $Q \subseteq \overline{P}$.

Since there is only a finite number of valid syllogistic patterns, completeness of the rules MON, SYM, EI for syllogistics can be proved by checking that every valid syllogistic pattern can be 'decomposed' into applications of MON, SYM and EI (see Section 6 below for examples). A computer program for this is given in [10]. Modulo the correctness of the program, this establishes:

Theorem 3. *The calculus consisting of the rules MON, SYM and EI is complete for syllogistics.*

It is well-known that a generalization of syllogistics can be based on the following parametrized version of Square of Opposition ([4, 9]):

$$\begin{array}{ccc} \text{AllExceptAtMost } N\ (\downarrow,\uparrow) & = & \text{AtMost } N\ (\downarrow,\downarrow) \\ \downarrow & & \downarrow \\ \text{AtLeast } (N+1)\ (\uparrow,\uparrow) & = & \text{AtLeast } (N+1) \text{ Not } (\uparrow,\downarrow) \end{array}$$

The traditional square is the special case of this with N set to 0.

Using $P \subseteq_n Q$ for

$$\exists P' \subseteq P(|P - P'| \leq n \wedge P' \subseteq Q),$$

we see that generalized monotonicity triggers now appear in the following guises:

- AllExceptAtMost n P Q: $P \subseteq_n Q$,
- AtMost n P Q: $P \subseteq_n \overline{Q}$, $Q \subseteq_n \overline{P}$.

Here is the generalized square in set-theoretic notation:

$$\begin{array}{ccc} P \subseteq_n Q & = & P \subseteq_n \overline{Q} \\ \downarrow & & \downarrow \\ P \not\subseteq_n \overline{Q} & = & P \not\subseteq_n Q \end{array}$$

Applications of monotonicity reasoning in this generalized setting look like this:

$$\frac{P \subseteq_n Q \qquad Q \subseteq_m R}{P \subseteq_{n+m} R}$$

Using $P \cap_n Q$ for $|P \cap Q| \geq n$, we see that $P \cap_{n+1} Q$ is equivalent to $P \not\subseteq_n \overline{Q}$. We get:

$$\frac{P \cap_n Q \qquad Q \subseteq_m R}{P \cap_{n \dot{-} m} R}$$

where $n \dot{-} m$ denotes cut-off subtraction (if $n \leq m$ then $n \dot{-} m = 0$).

Note that existential import yields nothing new for the new quantifiers. For let $n \geq 1$. Then from $P \neq \emptyset$ and $P \subseteq_n Q$ it does *not follow* that $P \cap Q \neq \emptyset$.

The situation changes when we adopt the following natural generalization of existential import, to keep in step with the new situation:

> Generalized existential import (GEI) for predicate P is the requirement that $|P| > n$.

Existential import for standard syllogistics is the special case of this where $n = 0$. Note that GEI does have an effect: from $|P| > n$ and $P \subseteq_n Q$ it *does follow* that $P \cap Q \neq \emptyset$.

Again, by a careful case by case analysis, we can establish:

Theorem 4. *The calculus consisting of the rules MON, SYM and GEI is complete for generalized syllogistics.*

The monotonicity behaviour of the function $\lambda n \lambda P \lambda Q.P \subseteq_n Q$ is given by:

$$\frac{P \subseteq_n Q \qquad n \leq m}{P \subseteq_m Q}$$

This yields the monotonicity marker map $((\mathrm{S}_i/\mathrm{VP})_r/\mathrm{CN})_i/\mathrm{NUM}$ for 'all except at most'. To build natural language fragments for generalized syllogistics, one can use a lexicon that has entries like the following:

C(all)	=	$(\mathrm{S}_i/\mathrm{VP})_r/\mathrm{CN}$	C(Greeks)	=	CN
C(some)	=	$(\mathrm{S}_i/\mathrm{VP})_i/\mathrm{CN}$	C(Atheneans)	=	CN
C(no)	=	$(\mathrm{S}_r/\mathrm{VP})_r/\mathrm{CN}$	C(barbarians)	=	CN
C(not all)	=	$(\mathrm{S}_r/\mathrm{VP})_i/\mathrm{CN}$	C(philosophers)	=	CN
$C(1), C(2), \ldots$	=	NUM	C(sophists)	=	CN
C(all except at most)	=	$((\mathrm{S}_i/\mathrm{VP})_r/\mathrm{CN})_i/\mathrm{NUM}$	C(cynics)	=	CN
C(at least)	=	$((\mathrm{S}_i/\mathrm{VP})_r/\mathrm{CN})_r/\mathrm{NUM}$	C(are)	=	$\mathrm{VP}_i/\mathrm{CN}$
C(at most)	=	$((\mathrm{S}_i/\mathrm{VP})_r/\mathrm{CN})_i/\mathrm{NUM}$	C(are not)	=	$\mathrm{VP}_r/\mathrm{CN}$

For such fragments, one can state and prove completeness results for monotonicity reasoning, by comparing the calculus with rules MON, SYM and GEI to the semantic consequence relation for first order models that result from interpreting the fragment: domains of discourse, plus interpretations for the common nouns. [19] gives an assessment of the complexity of the satisfiability problem for a variety of fragments starting from syllogistic theory. Syllogistic satisfiability is decidable in polynomial time. If relative clauses are added the complexity becomes NP, further addition of transitive verbs moves the complexity to EXPTIME, and so on. One can look at these findings in various ways. In [19] the conclusion is drawn that the programme of natural logic is hopeless:

> [...] from a complexity-theoretic point of view, there is every reason to believe that, for all but the most impoverished fragments, reasoning using schemata based on the syntax of natural language will confer no advantage whatever.

One may also draw the conclusion that natural logic is perhaps more complex and more interesting and more challenging than people used to believe.

6 Fine Structure of Syllogisms

Every valid syllogism involves exactly one application of the monotonicity rule, either triggered by 'All' or by 'No' (or in the generalized case, by 'All except at most N' or by 'At most N'). Arguably, the syllogisms that just involve monotonicity are the simplest ones. A syllogism may or may not involve an application of the following rules:

1. symmetry of a premise,
2. symmetry of the conclusion,
3. existential import of a premise
4. existential import of the conclusion.

As an example of decomposition of a syllogism in terms of the rules MON, SYM and EI, here are two possible decompositions of the syllogism *fesapo*:

$$\dfrac{\dfrac{\dfrac{\text{No } C\ B}{\text{No } B\ C}\ \text{Sn} \qquad \dfrac{\dfrac{\text{All } B\ A}{\text{Some } B\ A}\ \text{Ea}}{\text{Some } A\ B}\ \text{Ss}}{\text{Some } A\ \overline{C}}\ \text{Mn}$$

$$\dfrac{\dfrac{\dfrac{\text{No } C\ B}{\text{No } B\ C}\ \text{Sn} \qquad \dfrac{\text{All } B\ A}{\text{Some } B\ A}\ \text{Ea}}{\text{Some } \overline{C}\ A}\ \text{Mn}}{\text{Some } A\ \overline{C}}\ \text{Ss}$$

Here Mn denotes an application of MON with *No* as trigger, Ea denotes EI for *All*, Ss denotes SYM for *Some*, and so on.

Measured in terms of decomposition complexity, *fesapo* is the most complex valid syllogism. In an empirical set-up of [7], the inference from 'No B C, All B A' to 'NotAll A C' (the *fesapo* pattern) is only recognized as valid in 8 percent of the cases, while in a staggering 61 percent of the cases, subjects think, erroneously, that the conclusion *No A C* follows from the premises. The only cases where the scores are still lower for endorsement of a valid conclusion are cases where the conclusion follows by existential import from a universal negative conclusion that is *also* valid, and that is recognized in a majority of cases as being valid.

7 Monotonicity and Distribution

An important heuristics in traditional logic is the doctrine of distribution, consisting of the following two rules:

1. the middle term of a valid syllogism has to be distributed in at least one of the premises,
2. if a term of a valid syllogism is distributed in the conclusion it has to be distributed in one of the premises.

Prior [20] gives the following explanation of what 'distributed' means in these rules:

> It is often said [...] that a distibuted term refers to all, and an undistributed term to only a part, of its extension. But in what way does "Some men are mortal", for example, refer to only a part of the class of men? Any man whatever will do to verify it: if any man whatever turns out to be mortal, "Some men are mortal" is true. What the traditional writers were trying to express seems to be something of the following sort: a term t is distributed in a proposition $f(t)$ if and only if it is replaceable in $f(t)$, without loss of truth, by any term "falling under it" in the way that a species falls under a genus.

Interpreting 'being distributed' like this, we can see that

> From 'All A B' and 'All B C' infer 'All A C'

has the middle term B distributed in 'All B C', in agreement with the first rule of distribution, while B violates the first rule of distribution in the following invalid pattern:

> From 'All A B' and 'Some B C' infer 'All A C'.

An invalid pattern that violates the second rule of distribution is:

> From 'Some A B' and 'All B C' infer 'All A C'.

Here A is distributed in the conclusion, but not in the premise where it occurs.

Prior's suggestion of a modern version of the doctrine of distribution is taken up in Van Benthem [4]. In Van Eijck [9] the relations between traditional logic

(syllogistic theory) and generalized quantifier theory [18, 1, 3] are worked out further, with due attention to the role of monotonicity in syllogistic reasoning, and with the observation that the square of opposition generalizes to quantifiers defined from *At least n*.

Hodges [16] relates the doctrine of distribution to monotonicity (just as [20, 4, 9] had done before), and gives a semantic argument to show that the correctness of the two rules of distribution follows from the interpretation of 'distributed term' as 'term in a downward monotone position'. The doctrine of distribution also follows from our completeness result. Consider the first rule of distribution, saying that the middle term has to be distributed in at least one of the premises. If the trigger of the monotonicity rule is 'No P Q', then this condition is always fulfilled, for both P and Q are in downward position. If the trigger of the monotonicity rule is 'All P Q', then the condition is fulfilled if P is the middle term, for P is in a downward position in 'All P Q', and it is also fulfilled if Q is the middle term, for the monotonicity rule allows substitution of Q by P in the other premise only if Q is in downward position in that premise. Hodges shows that the second rule of distribution follows from the first rule, as follows. Let ϕ and ψ be the two premises, and assume P is in downward position in $\chi(P)$, where

ϕ, ψ, therefore $\chi(P)$

is a valid syllogism. Assume, without loss of generality, that P is a term in ϕ, and suppose that P is in upward position in $\phi(P)$. Then

$\phi(P)$, $\overline{\chi}(P)$, therefore $\overline{\psi}$

is also a valid syllogism. But in this syllogism P is the middle term. Moreover, the effect of wide scope negation is that P is in upward position in $\overline{\chi}(P)$, and we have a contradiction with the first rule of distribution.

8 Related and Future Work

Sanchez [22] is an extensive study of the role of monotonicity in 'natural reasoning', with as main contribution an algorithm for monotonicity marking, and a system for monotonicity reasoning in terms of monotonicity markings. This work is based on [5], and is in turn the basis of almost all later proposals for monotonicity calculi.

In Geurts [14] a monotonicity based system of reasoning for syllogistics is sketched, in terms of Sanchez-style monotonicity markings. The claim is made that monotonicity, symmetry and existential import account for all syllogistic inference, but the presentation of the rules is too informal to admit a proof of this. Geurts' intention is to explain empirical findings about accomplishment in syllogistic reasoning tasks in terms of complexity of inference in his reasoning system. It seems clear that the interest of syllogistics for cognitive science lies in the mechanism of monotonicity. Connecting the logical exploration of this mechanism with empirical findings in the psychology of reasoning is an obvious next goal. The hypothesis of [15] that reversal of monotonicity increases human

processing load can be linked to the mental models metaphor. Interestingly, from a logical point of view the reversed monotonicity pattern is no more complex than the pattern of preserved monotonicity.

Monotonicity calculi can be specified in a fully precise manner, by presenting them as proof calculi, consisting of axioms and inference rules. Such calculi are meant to capture standard notions of logical consequence: they are not calculi of logical falsehoods. If they can be used to explain where human reasoners err, it should be in an indirect manner, by making clear what the added complexity of a particular task is in comparison with tasks where human reasoners tend not to err. This suggests that, given a suitably precise version of a monotonicity calculus, it should be possible to flesh out the mental models metaphor as a formally precise extension of the monotonicity calculus, a kind of add-on tool that allows us to classify reasoning tasks with respect to their claims on the human processing faculty [2].

Acknowledgement Thanks to Fabian Battaglini for getting me interested in the topic of natural logic again, and for inspiring discussions. Thanks to Ian Pratt and two anonymous referees for spotting errors in and providing illuminating comments to an earlier version of this paper. All remaining errors are my own.

References

1. J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219, 1981.
2. Fabian Battaglini. Monotonicity and cognition, Manuscript, Uil-OTS, Utrecht, 2006.
3. J. van Benthem. Questions about quantifiers. *Journal of Symbolic Logic*, 49:443–466, 1984.
4. J. van Benthem. *Essays in Logical Semantics*. Reidel, Dordrecht, 1986.
5. J. van Benthem. *Language in Action: categories, lambdas and dynamic logic*. Studies in Logic 130. Elsevier, Amsterdam, 1991.
6. R. Bernardi. *Reasoning with Polarity in Categorial Type Logic*. PhD thesis, Uil-OTS, Utrecht University, 2002.
7. N. Chater and M. Oaksford. The probability heuristics model of syllogistic reasoning. *Cognitive Psychology*, 38:191–258, 1999.
8. D. Dowty. Negative polarity and concord marking in natural language reasoning. In *SALT Proceedings*, 1994.
9. J. van Eijck. Generalized quantifiers and traditional logic. In J. van Benthem and A. ter Meulen, editors, *Generalized Quantifiers, Theory and Applications*. Foris, Dordrecht, 1985.
10. Jan van Eijck. Syllogistics = monotonicity + symmetry + existential import. Technical Report SEN-R0512, CWI, Amsterdam, July 2005. Available from `http://db.cwi.nl/rapporten/`.
11. G. Englebretsen. Notes on the new syllogistic. *Logique et Analyse*, 85–86:111–120, 1979.
12. C. Fellbaum. *Wordnet, an electronic lexical database*. MIT Press, 1998.
13. Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. Order-based inference in natural logic. *Logic Journal of the IGPL*, 11:385–416, July 2003.

14. B. Geurts. Reasoning with quantifiers. *Cognition*, 86:223–251, 2003.
15. B. Geurts and F. van der Slik. Monotonicity and processing load. *Journal of Semantics*, 22(97–117), 2005.
16. W. Hodges. The laws of distribution for syllogisms. *Notre Dame Journal of Formal Logic*, 39:221–230, 1998.
17. P.N. Johnson-Laird. *Mental Models; towards a cognitive science of language, inference and consciousness*. Cambridge University Press, 1983.
18. A. Mostowski. On a generalization of quantifiers. *Fundamenta Mathematica*, 44:12–36, 1957.
19. I. Pratt-Hartmann. Fragments of language. *Journal of Logic, Language and Information*, 13(2):207–223, 2004.
20. A.N. Prior. Traditional logic. In P. Edwards, editor, *The Encyclopedia of Philosophy*, volume 5, pages 34–45. Macmillan, 1967.
21. W.C. Purdy. A logic for natural language. *Notre Dame Journal of Formal Logic*, 32:409–425, 1991.
22. V. Sánchez. *Studies on Natural Logic and Categorial Grammar*. PhD thesis, University of Amsterdam, 1991.
23. F. Sommers. *The Logic of Natural Language*. Cambridge University Press, 1982.