Polarity sensitivity and evaluation order in type-logical grammar

Chung-chieh Shan
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
ccshan@post.harvard.edu

Abstract
We present a novel, type-logical analysis of polarity sensitivity: how negative polarity items (like any and ever) or positive ones (like some) are licensed or prohibited. It takes not just scopal relations but also linear order into account, using the programming-language notions of delimited continuations and evaluation order, respectively. It thus achieves greater empirical coverage than previous proposals.

1 Introduction
Polarity sensitivity (Ladusaw, 1979) has been a popular linguistic phenomenon to analyze in the categorial (Dowty, 1994), lexical-functional (Fry, 1997, 1999), and type-logical (Bernardi, 2002; Bernardi and Moot, 2001) approaches to grammar. The multitude of these analyses is in part due to the more explicit emphasis that these traditions place on the syntax-semantics interface—be it in the form of Montague-style semantic rules, the Curry-Howard isomorphism, or linear logic as glue—and the fact that polarity sensitivity is a phenomenon that spans syntax and semantics.

On one hand, which polarity items are licensed or prohibited in a given linguistic environment depends, by and large, on semantic properties of that environment (Ladusaw, 1979; Krifka, 1995, inter alia). For example, to a first approximation, negative polarity items can occur only in downward-entailing contexts, such as under the scope of a monotonically decreasing quantifier. A quantifier q, of type (e → t) → t where e is the type of individuals and t is the type of truth values, is monotonically decreasing just in case

\[ \forall s_1. \forall s_2. (\forall x. s_2(x) \Rightarrow s_1(x)) \Rightarrow q(s_1) \Rightarrow q(s_2). \]

Thus (2a) is acceptable because the scope of nobody is downward-entailing, whereas (2b–c) are unacceptable.

(2) a. Nobody saw anybody.
   b. *Everybody saw anybody.
   c. *Alice saw anybody.

On the other hand, a restriction on surface syntactic form, such as that imposed by polarity sensitivity, is by definition a matter of syntax. Besides, there are syntactic restrictions on the configuration relating the licensor to the licensee. For example, (2a) above is acceptable—nobody manages to license anybody—but (3) below is not. As the contrast in (4) further illustrates, the licensor usually needs to precede the licensee.

(3) *Anybody saw nobody.

(4) a. Nobody’s mother saw anybody’s father.
   b. *Anybody’s mother saw nobody’s father.

The syntactic relations allowed between licensor and licensee for polarity sensitivity purposes are similar to those allowed between antecedent and pronoun for variable binding purposes. To take one example, just as an antecedent’s ability to bind a (c-commanded) pronoun percolates up to a containing phrase (such as in (5), what Büring (2001) calls “binding out of DP”), a licensor’s ability to license a (c-commanded) polarity item percolates up to a containing phrase (such as in (4a)).

(5) [Every boy’s mother] saw his father.

Moreover, just as a bindee can precede a binder in a sentence when the bindee sits in a clause that excludes the binder (as in (6); see Williams, 1997, §2.1), a licensee can precede a licensor in a sentence when the licensee sits in a clause that excludes the licensor (as in (7); see Ladusaw, 1979, page 112).

(6) That he would be arrested for speeding came as a surprise to every motorist.

(7) That anybody would be arrested for speeding came as a surprise to the district attorney.

This paper presents a new, type-logical account of polarity sensitivity that encompasses the semantic properties exemplified in (2) and the syntactic properties exemplified in (3–4). Taking advantage of the Curry-Howard isomorphism, it is the first account of polarity sensitivity in the grammatical frameworks mentioned above to correctly fault (3) for the failure of nobody to appear before
We use two binary modes: the default mode (blank) for surface syntactic composition, and the continuation mode $c$. As usual, a formula of the form $A \circ B$ can be read as “$A$ followed by $B$”. By contrast, a formula of the form $A \circ c B$ can be read as “$A$ in the context $B$”. In programming-language terms, the formula $A \circ c B$ plugs a subexpression $A$ into a delimited continuation $B$. The Root rule creates a trivial continuation: it says that 1 is a right identity for the $c$ mode, where 1 can be thought of as a nullary connective, effectively enabling empty antecedents for the $c$ mode. The binary modes, along with the first three postulates in Figure 2, provide a new way to encode Moortgat’s ternary connective $q$ (1996b) for in-situ quantification. For intuition, it may help to draw logical formulas as binary trees, distinguishing graphically between the two modes.

To further capture the interaction between scope inversion and polarity sensitivity exemplified in (3–4), we use three unary modes: the value mode (blank), the unquotation mode $u$, and the polarity mode $p$. The value mode marks when an expression is devoid of in-situ quantification, or, in programming-language terms, when it is a pure value rather than a computation with control effects. As a special case, any formula can be turned pure by embedding it under a diamond using the $T$ postulate, analogous to *quotation* or *staging* in programming languages. Quotations can be concatenated using the $K'$ postulate. The unquotation mode $u$ marks when a diamond can be canceled using the Unquote postulate. Unquotation is also known as *eval* or *run* in programming languages. The polarity mode $p$, and the empirical utility of these unary modes, are explained in §3.

A derivation is considered complete if it culminates in a sequent whose antecedent is built using the default binary mode $\circ$ only, and whose conclusion is a type of the form $\diamond u A$. Below is a derivation of Alice saw Bob.

\[
\frac{\text{saw } \circ (\text{saw } \circ \text{Bob}) \oplus (\text{Bob } \circ \text{saw })}{\text{Alice saw Bob}}
\]

Note that clauses take the type $\diamond u s$ rather than the usual $s$, so the Unquote rule can operate on clauses. We abbreviate $\diamond u s$ to $s'$ below.

To illustrate in-situ quantification, Figure 3 on the following page shows a derivation of Alice saw a man’s mother. For brevity, we treat a man as a single lexical item. It is a quantificational noun phrase whose polarity is *neutral* in a sense that contrasts with other quantifiers considered below. The crucial part of this derivation is the use of the structural postulates Root, Left, and Right to divide the sentence into two parts: the subexpression a man and its context Alice saw’s mother. The type of a man, $s' \circ (np) \circ s'$, can be read as “a subexpression that produces a clause when placed in a context that can enclose an np to make a clause”.

Figure 1 shows natural deduction rules for multimodal categorial grammar, a member of the type-logical family of grammar formalisms (Moortgat, 1996a; Bernardi, 2002). Figure 2 lists our structural postulates. These two figures form the logic underlying our account.
3 Polarity sensitivity and evaluation order

The $p$ mode mediates polarity sensitivity. For every unary mode $\alpha$, we can derive $A \vdash \Diamond_p \ominus_p A$ from the rules in Figure 1. This fact is particularly useful when $\alpha = p$, because we assign the types $\Diamond_p \ominus_p p$ and $\Box_p \ominus_p q$ to positive and negative clauses, respectively, and can derive

\begin{equation}
\begin{align*}
&\Delta p \vdash (np \wedge s) / np \wedge s, \\
&\Delta p \vdash (np \wedge s) / np \wedge s
\end{align*}
\end{equation}

In words, a neutral clause can be silently converted into a positive or negative one. We henceforth write $s^+$ and $s^-$ for $\Diamond_p \ominus_p p$ and $\Box_p \ominus_p q$. By (9), both types are “subtypes” of $s^\ominus$ (that is to say, entailed by $s^\ominus$).

The $p$ mode is used in Figure 5 on the next page to derive Nobody saw anybody. Unlike a man, the quantifier anybody has the type $s^\ominus \vdash (np \wedge s^\ominus)$, showing that it takes scope over a negative clause to make another negative clause. Meanwhile, the quantifier nobody has the type $s^\ominus \vdash (np \wedge s^\ominus)$, showing that it takes scope over a negative clause to make a neutral clause. Thus nobody can take scope over the negative clause returned by anybody to make a neutral clause, which is complete.

The contrast between (2a) and (3) boils down to the Right (but not Left) postulate’s requirement that the leftmost constituent be of the form $\Diamond B$. (In programming-language terms, a subexpression can be evaluated only if all other subexpressions to its left are pure.) For nobody to take scope over (and license) anybody in (3) requires the context *Anybody saw _. In other words, the sequent

\begin{equation}
\begin{align*}
&np \vdash ((1 \Diamond \text{anybody}) \Diamond \text{saw}) \wedge s^-
\end{align*}
\end{equation}

must be derived, in which the Right rule forces the constituents anybody and saw to be embedded under diamonds. Figure 6 shows an attempt at deriving (10), which fails because the type $s^-$ for negative clauses cannot be Unquoted (shown with question marks). The sequent in (10) cannot be derived, and the sentence *Anybody saw nobody is not admitted. Nevertheless, Somebody saw everybody is correctly predicted to have ambiguous scope, because neutral and positive clauses can be Unquoted.

The quantifiers a man, nobody, and anybody in Figures 3 and 5 exemplify a general pattern of analysis: every polarity-sensitive item, be it traditionally considered a licenser or a licensee, specifies in its type an input polarity (of the clause it takes scope over) and an output polarity (of the clause it produces). Figure 4 lists more quantifiers and their input and output polarities. As shown there, these type assignments can be visualized as a finite-state machine. The states are the three clause types. The $\epsilon$-transitions are the two derivability relations in (9). The non-$\epsilon$ transitions are the quantifiers. The start states are the clausal types that can be Unquoted. The final state is the clausal type returned by verbs, namely neutral.

The precise pattern of predictions made by this theory can be stated in two parts. First, due to the lexical types in Figure 4 and the “subtyping” relations in (9), the quantifiers in a sentence must form a valid transition sequence, from widest to narrowest scope. This constraint is standard in type-logical accounts of polarity sensitivity. Second, thanks to the unary modes in the structural
postulates in Figure 2, whenever two quantifiers take inverse rather than linear scope with respect to each other, the transitions must pass through a start state (that is, a clause type that can be Unquoted) in between. This constraint is an empirical advance over previous accounts, which are oblivious to linear order.

The input and output polarities of quantifiers are highly mutually constrained. Take everybody for example. If we hold the polarity assignments of the other quantifiers fixed, then the existence of a linear-scope reading for A man introduced everybody to somebody forces everybody to be input-positive and output-neutral. But then our account predicts that Nobody introduced everybody to somebody has a linear-scope reading, unlike the simpler sentence Nobody introduced Alice to somebody. This prediction is borne out, as observed by Kroch (1974, pages 121–122) and discussed by Szabolcsi (2004).

Acknowledgments

Thanks to Chris Barker, Raffaella Bernardi, William Ladusaw, Richard Moot, Chris Potts, Stuart Shieber, Dylan Thurston, and three anonymous referees. This work is supported by the United States National Science Foundation Grant BCS-0236592.

References

Raffaella Bernardi and Richard Moot. 2001. Generalized quantifiers in declarative and interrogative sentences. Journal of Language and Computation, 1(3):1–19.

Raffaella Bernardi. 2002. Reasoning with Polarity in Categorial Type Logic. Ph.D. thesis, Utrecht Institute of Linguistics (OTS), Utrecht University.

Daniel Béring. 2001. A situation semantics for binding out of DP. In Rachel Hastings, Brendan Jackson, and Zsofia Zvolensky, editors, SALT XI: Semantics and Linguistic Theory, pages 56–75, Ithaca. Cornell University Press.

Olivier Danvy and Andrzej Filinski. 1990. Abstracting control. In Proceedings of the 1990 ACM Conference on Lisp and Functional Programming, pages 151–160, New York, March. ACM Press.

David R. Dowty. 1994. The role of negative polarity and concord marking in natural language reasoning. In Mandy Harvey and Lynn Santelmann, editors, SALT IV: Semantics and Linguistic Theory, Ithaca. Cornell University Press.

Matthias Felleisen. 1988. The theory and practice of first-class prompts. In POPL ’88: Conference Record of the Annual ACM Symposium on Principles of Programming Languages, pages 180–190, New York. ACM Press.

John Fry. 1997. Negative polarity licensing at the syntax-semantics interface. In Philip R. Cohen and Wolfgang Wahlster, editors, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, pages 144–150, San Francisco. Morgan Kaufmann.

John Fry. 1999. Proof nets and negative polarity licensing. In Mary Dalrymple, editor, Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach, chapter 3, pages 91–116. MIT Press, Cambridge.

Manfred Krifka. 1995. The semantics and pragmatics of polarity items. Linguistic Analysis, 25:209–257.

Anthony S. Kroch. 1974. The Semantics of Scope in English. Ph.D. thesis, Massachusetts Institute of Technology. Reprinted by New York: Garland, 1979.

William A. Ladusaw. 1979. Polarity Sensitivity as Inherent Scope Relations. Ph.D. thesis, Department of Linguistics, University of Massachusetts, August. Reprinted by New York: Garland, 1980.

Michael Moortgat. 1996a. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, Handbook of Logic and Language, chapter 2. Elsevier Science, Amsterdam.

Michael Moortgat. 1996b. Generalized quantification and discontinuous type constructors. In Harry C. Bunt and Arthur van Horck, editors, Discontinuous Constituency, pages 181–207. de Gruyter, Berlin.

Richard Moot. 2002. Proof Nets for Linguistic Analysis. Ph.D. thesis, Utrecht Institute of Linguistics (OTS), Utrecht University.

Anna Szabolcsi. 2004. Positive polarity—negative polarity. Natural Language and Linguistic Theory, 22(2):409–452, May.

Edwin Williams. 1997. Blocking and anaphora. Linguistic Inquiry, 28(4):577–628.