Principle-based interpretation of natural language quantifiers

Samuel S. Epstein
Bell Communications Research
445 South Street, 2Q-350
Morristown, NJ 07960-1910

Abstract

This paper describes a working prototype that determines possible relative quantifier scopes and pronoun bindings for natural language sentences, with coverage of a variety of problematic cases. The prototype parses a significant fragment of English, positing empty categories and deriving various relationships among constituents in addition to dominance. It applies cross-linguistically valid principles of Government-Binding theory to compute a set of "Logical Forms" for each sentence it parses, and to derive possible relative quantifier scopes from these Logical Forms. It then translates sentences into an enriched predicate logic. Simple principles apply to these translations to determine possibilities for interpretation of pronouns as bound variables. The prototype's scope and binding modules correspond transparently to elements of a principle-based grammar. Principles apply as filters. All processing is nevertheless highly efficient. The computational techniques employed in the prototype may find wider application in principle-based language processing.

1. Introduction

The interpretation of quantifiers is one of the central problems of natural language understanding. Quantifiers include expressions like *everyone, many students,* and *the professor that skates.* Given a suitably general notion of "quantifier," few natural language sentences contain no quantifiers. On some accounts, all natural language sentences contain quantifiers. This paper describes a working prototype, called "QSB" ("Quantifier Scopes and Bindings"), that determines possible relative quantifier scopes and pronoun bindings for natural language sentences, with coverage of a variety of problematic cases.[1] QSB parses a significant fragment of English and translates it into an enriched predicate logic. The computational techniques that it employs may find wider application.

(1) Every professor expects several students to read many books.

is an example of a sentence with several possibilities for relative quantifier scope. To take one possibility, the "several-every-many" reading, there can be a particular set of several students such that every professor expects each of those students to read many books, where for each choice of student and professor, there may be a different set of many books. The other possibilities for (1) are "every-several-many" and "every-many-several."[2]

(2) Every professor that knows a student that owns a computer covets it.

and

(3) Every professor that knows every student that owns a computer covets it.

illustrate pronoun binding.[3] *A computer* can bind *it* in (2), but not in (3). Studies of relative quantifier scope and of pronoun binding have examined a great variety of examples from a variety of languages and have demonstrated the apparent complexity of these phenomena, but have also made impressive progress toward finding underlying regularities.[4]

QSB follows a principle-based approach to language processing. Principle-based grammars are a recent development in linguistic theory. They are particularly associated with the Government-Binding theory of syntax ("GB").[5] Principle-based grammars characteristically contain a small number of heterogeneous principles, rather than a

1. QSB is implemented in Common Lisp on Symbolics, Release 7.1.

2. QSB does not yet deal explicitly with the possibility of branching quantifiers.

3. "Bind" has distinct technical meanings recognized by different communities of researchers. Its meaning here is fairly close to the standard in logic.

4. Good (although not up-to-date) bibliographies of relevant work are included in [May, 1985] (relative quantifier scope), and in [Heim, 1982], [van Riemsdijk and Williams, 1986], and [Brennan *et al.*, 1987] (pronoun binding).

5. [Chomsky, 1981] is the seminal work on Government-Binding theory. [Van Riemsdijk and Williams, 1986] is a textbook introduction. [Berwick, 1987] discusses the computational exploitation of principle-based grammars.

large number of homogeneous rules. Ideally, principles are uniformly valid for all natural languages. Variation among natural languages is a matter of setting parameters, like "head initial" or "head final," and supplying a lexicon. On the classical conception, principles constrain freely generated linguistic structures. Structures that conform to all the parametrized principles of a grammar belong to the language associated with the grammar. The modularity, simplicity, and substantial shared content of principle-based grammars offer strong advantages for natural language processing. However, it is necessary to confront some apparent problems for principle-based language processing, as discussed in Section 3 below.

For purposes of exposition, QSB may be decomposed into three modules - a parser module, a scope module, and a binding module. The scope and binding modules directly implement aspects of a principle-based grammar. The parser module does not. The next three sections describe these modules in turn. This paper emphasizes computational techniques for efficient implementation of principle-based grammars. Because of space limitations, its discussion of other aspects of the prototype is very brief.

2. Parser module

The QSB parser module produces usable surface structure parses for the scope and binding modules. The other QSB modules could be made to work with a parser of different design and functionality, providing that this other parser correctly analyzed certain phenomena. The QSB parser is not among the chief points of interest of this work. It will eventually be replaced by a parser that directly implements grammatical principles. However, the current parser's analyses do include some information that most other parsers fail to derive.

The QSB parser is basically a recursive descent parser with a data-driven component. While it is not principle-based in any strong sense, its analyses conform to Government-Binding theory, particularly to an elaboration of Government-Binding theory proposed in [Aoun and Li, to appear]. It finds only a single constituent structure analysis for each sentence that it parses, hypothetically corresponding to a preferred reading. In addition to finding constituency relationships among overt categories, the parser posits certain empty categories (*wh*-trace, NP-trace, and PRO), and associates these empty categories with the categories that bind them. The parser also sets pointers from determiners to their noun phrase complements, or "restrictions." QSB includes a facility for bit-mapped displays of parse structures, with various links between nodes ("control," and so on) indicated by various kinds of line (for example, "chains" look like chains).

The current parser produces correct results for a subset of English that exhibits the following phenomena, among others: coordination, relativization, raising, obligatory control, and exceptional case marking. For example, it produces an accurate parse for

(4) Every student that admires a dean that every professor seems to respect wants to read many books and some instructor expects many students to read several books that every professor likes and many professors love.

in 0.12 seconds (Symbolics 3645, Release 7.1).

3. Scope module

The scope module is based on an account in [Aoun and Li, to appear], as adapted in [Aoun and Epstein, to appear]. Aoun and Li explain data from several languages concerning relative quantifier scope and relative scope of quantifiers and *wh* operators (such as *who*). Their entirely general and principle-based exposition covers a great variety of syntactic constructions, including, for example, the cases discussed in [Hobbs and Shieber, 1987].

Following [May, 1977], Aoun and Li base their treatment on a rule of "quantifier raising" that is used to derive "Logical Forms" ("LF"'s) from "Surface Structures" ("SS"'s). Aoun and Li formulate alternative accounts of quantifier raising. In the adapted account of [Aoun and Epstein, to appear], LF's are obtained from SS's by raising determiners. Well-formed LF's conform to the following four principles, stated here as they apply in the scope module of QSB:

(I) (Phrasal-node-adjunction) Determiners are raised only to phrasal nodes (such as noun phrase nodes, verb phrase nodes, and sentence nodes).

(II) (Non-theta-adjunction) Determiners are never raised to "theta" positions (argument positions within verb phrases, such as direct object positions).

(III) (Opacity) Determiners are never raised outside their opaque domains. (The "opaque domain" of a determiner is roughly speaking the smallest clause that contains the determiner and either a subject or a tensed verb.)

(IV) (Minimal Binding Requirement, or "MBR") A determiner's "landing site" cannot dominate the "launch site" of another determiner unless it also dominates the landing site of that other determiner.

(I) - (IV) have independent linguistic motivations. Given a well-formed LF, possible relative quantifier scopes are determined by the "Scope Principle," which states in effect that a quantifier Q_1 may have scope over a quantifier Q_2 in case the lowest phrasal node that dominates the landing site of the determiner of Q_1 also dominates the determiner of Q_2 or a trace associated with Q_2. Traces are empty (non-overt) categories. For example, in

(5) Every student seems to admire some professor.

the subject of the infinitive clause *to admire some professor* is a trace associated with *every student*. When *some professor* raises to the top of its opaque domain (the clause *to admire some professor)* it is "higher" than the trace of *every student*, and so by the Scope Principle, *some professor* can have scope over *every student*. Note that LF's do not disambiguate sentences with respect to quantifier scope. The set of possible quantifier scope readings for a sentence is the union of possible scopings over the set of its well-formed LF's.

This is a principle-based account of relative quantifier scope. As with other principle-based accounts, a simple-minded implementation is computationally hopeless. For example, assuming quantifier raising applies without any of the constraints (I) - (IV), (5) has 70 candidate LF's.

(6) Some dean seems to expect several professors to want every student to read many books.

has 50830 candidate LF's. Even for a moderately long sentence like (6), generating each candidate and testing it against (I) - (IV) is absurdly impractical. This absurdity might be compounded by applying the Scope Principle to candidate LF's before filtering them.

There thus may appear at first glance to be a trade-off between the simplicity and modularity of principle-based grammars and the computational expense of running the generate-and-test model that they seem to incorporate. One method of confronting this apparent trade-off is to write a language processor which produces outputs that correspond to well-formed structures according to a principle-based grammar, but which makes no use of principles itself. It is not clear how a processor that isn't itself principle-based can be made to share advantages of principle-based grammars.

According to one ideal, efficient language processors would be compiled from declarative specifications of principle-based grammars. [Berwick, 1987] and [Johnson, 1987] discuss some very preliminary ideas along these lines. This is an ambitious goal with no immediate prospect of achievement. Grammatical principles vary greatly in their forms and in how they interact. Use of general-purpose theorem-proving technology does not (yet) offer a practical solution to this problem.

The quantifier scope module of QSB follows a third broad approach to the implementation of principle-based grammars. The implementation directly mirrors the principle-based grammar. Principles apply as function calls. Effective use of some programming strategies permits highly efficient processing. The implementation retains advantages of a principle-based approach. Extensions and alterations are entirely straightforward.

More specifically, the quantifier scope module of QSB obtains efficiency primarily through six strategies:

(α) **Easier-Earlier Strategy -**
Principles whose applications require less work apply earlier.

(β) **Maximal Filtering Strategy -**
Principles that filter more representations apply earlier.

(γ) **Wholesale Filtering Strategy -**
Filters apply to classes of representations (where possible), rather than to single representations.

(δ) **Schematic Representation -**
Principles apply to schematic representations (where possible).

(ε) **Minimal Construction Strategy -**
Principles apply to components of representations prior to construction of representations (where possible); only representations whose components pass filters are constructed.

(ζ) **Partitioning -**
Representations are partitioned (or quasi-partitioned) to minimize domains of application of principles (where possible).

Accumulation of experience may lead to the formalization and eventual automation of these techniques. The examples that follow illustrate their application in the scope module of QSB.

As an example of the Easier-Earlier strategy, consider Non-theta-adjunction and the MBR. Non-theta-adjunction is a very simple check on landing sites. The MBR must consider interactions among members of sets of (determiner, landing-site) pairs. It is more expensive computationally than Non-theta-adjunction, and should thus apply only after Non-theta-adjunction has reduced its domain of application. If the MBR is ordered before Non-theta-adjunction, time to find quantifier orderings for (6) is 0.49 seconds (Symbolics 3645, Release 7.1). If Non-theta-adjunction is ordered before the MBR, following the Easier-Earlier strategy, time to find quantifier orderings for (6) is 0.26 seconds.

As an example of the Maximal Filtering strategy, consider Opacity and Non-theta-adjunction. In order to make a reasonable comparison of the relative filtering power of these two principles, suppose that both principles apply after Phrasal-node-adjunction and before the MBR.[6] When a sentence contains a single opaque domain, Opacity does little work. The more opaque domains a sentence contains, the more candidate LF's are filtered by Opacity. For (5), with two opaque domains, Non-theta-adjunction applying after Phrasal-node-adjunction passes 15 candidate LF's to Opacity and the MBR. Opacity applying after Phrasal-node-

6. In practice, the Wholesale Filtering strategy stipulates that neither Opacity nor Non-theta-adjunction applies to individual LF's. In addition, the Maximal Filtering strategy requires ordering Phrasal-node-adjunction after Opacity but before Non-theta-adjunction, subject to reservations noted below.

adjunction passes 6 candidate LF's to Non-theta-adjunction and the MBR. For (6), Non-theta-adjunction applying after Phrasal-node-adjunction passes 1701 candidate LF's to Opacity and the MBR. Opacity applying after Phrasal-node-adjunction passes 150 candidate LF's to Non-theta-adjunction and the MBR. Given a policy of optimizing average-case performance, (not to mention a policy of avoiding very bad worst-case performance) the Maximal Filtering Strategy would seem to require ordering Opacity before Non-theta-adjunction.[7]

Applications of Opacity, Non-theta-adjunction, and Phrasal-node-adjunction in the scope module of QSB all illustrate the Wholesale Filtering strategy. For example, for (6), any candidate LF where *many* is raised to its closest dominating phrasal node violates Non-theta-adjunction. It is possible to eliminate all these candidate LF's with a single application of Non-theta-adjunction. With this kind of wholesale filtering, the total number of applications of Non-theta-adjunction necessary to process (6) is 15. With Non-theta-adjunction correctly ordered after Opacity and Phrasal-node-adjunction and before the MBR, but without wholesale filtering, the number of applications of Non-theta-adjunction for (6) is 203.

Schematic linguistic representations abstract away what is irrelevant to the purposes at hand. Their use corresponds to a radical sort of structure-sharing. For example, given a full representation of the Surface Structure of a sentence, each candidate LF for the sentence can be represented as a set of (determiner, landing-site) pairs, with one pair for each determiner in the sentence. Properties of candidate LF's can be read off their schematic representations in association with the SS. It is thus possible to apply (I) - (IV) and the Scope Principle without ever computing full LF's. The notion of schematic representation is related to the notion of "use of knowledge" of structures in [Johnson, 1987].

The Minimal Construction strategy reduces the number of representations that get constructed, and thus reduces the amount of time and space expended on the construction of representations. Minimal construction is similar to lazy evaluation. For example, constructing a set of schematic representations of LF's for a sentence requires constructing for each determiner d in the sentence a set of pairs of the form (d, landing-site), and then taking the Cartesian product of these sets of pairs. Opacity, Phrasal-node-adjunction, and Non-theta-adjunction apply directly to landing sites. Following the Minimal Construction strategy, these three principles apply to reduce the size of the set of candidate landing sites for each determiner prior to the construction of schematic representations of LF's. For (6), the number of candidate LF's constructed is thereby reduced from 50830 to 64.

The technique of partitioning linguistic representations applies readily to the problem of computing relative quantifier scopes. It follows from Opacity (and may be observed independently) that relative quantifier scope relationships never arise across coordination boundaries. It is therefore possible to compute relative quantifier scopes one coordinate at a time. For example, in

(7) Every dean read few books and many students read several reports.

the question of relative scope for *few books* and *many students* does not arise. In order to analyze (7), it is sufficient to analyze *every dean read few books* and *many students read several reports,* and then "multiply" the analyses. Thus rather than considering 4! = 24 possible relative quantifier scopings, it is necessary only to consider 2 possible scopings in the first conjunct, and 2 in the second. Similarly, quantifiers in a relative clause (for example) can only enter into direct relative quantifier scope relationships inside the relative clause or with its head. In

(8) Every dean that many professors admire reads few books.

the question of relative scope for *few books* and *many professors* does not arise. In order to analyze (8), it is sufficient to consider ordering possibilities for *every dean* and *few books. Many professors* inside the relative clause must have narrower scope than *every dean.*

Examples like (1) require "quasi-partitioning." Rather than analyze (1) as a single structure it is possible to divide this sentence into the slightly overlapping quasi-partitions *every professor expects several students* and *several students to read many books.* Quasi-partitioning may proceed top-down as follows: (i) Find all quantifiers that lie within the clause in question but no lower clause. (ii) Find the lowest clause that contains a member of the chain of one of these quantifiers. This lowest clause, with all intermediate clauses, is included in the quasi-partition. ((5) in its entirety is thus included in a single quasi-partition.) (iii) If the next lower clause is an infinitive and has a subject, also include this subject in the quasi-partition. Given possible relative quantifier scope orderings within quasi-partitions for a sentence, the possible orderings for the entire sentence are those orderings which are consistent with possible orderings within quasi-partitions.

7. As optimal ordering for application of principles varies from sentence to sentence, orderings might be adjusted based on simplified preliminary analyses of sentences. For the principles implemented in the scope module of QSB, such case by case adjustment does not appear to save computational resources overall.

The Easier-Earlier strategy and the Maximal Filtering strategy may conflict. For example, Opacity is a more complex principle than Phrasal-node-adjunction, but for long sentences it filters more LF's. I am not aware of any general method that resolves conflicts between ordering strategies. In this case, it seems advantageous to order Opacity first.

Quasi-partitioning can yield dramatic performance advantages. Consider

(9) $[_0$ Every professor expects $[_1$ several students $]_0$ to want $[_2$ few deans$]_1$ to expect $[_3$ some freshman $]_2$ to read many books $]_3$.

which quasi-partitions as indicated. (9) has 5 quantifiers, with 8 possible relative quantifier scope orderings. Without quasi-partitioning, it is necessary to consider 5! = 120 possible orderings. If processing is set up to follow strategies (α) - (ζ) but not (quasi-)partitioning, 50 seconds are required to compute relative quantifier scope orderings for (9). With (quasi-)partitioning, 0.45 seconds are required, an improvement of two orders of magnitude. It seems likely that an analog of Partitioning plays a role in human language processing.

Strategies (α) - (ζ), working in concert with application of some additional programming practices, permit highly efficient computation of relative quantifier scope possibilities. Given the output of the parser module, the scope module computes the 3 relative scope possibilities for sentence (4) (which has 9! = 362880 candidate orderings) in 0.16 seconds (Symbolics 3645, Release 7.1).

4. Binding module

I describe the binding module in a forthcoming paper. Space limitations permit only a brief summary here.

The binding module computes possible quantifier antecedents for pronouns. For example, it determines that *a donkey* can bind *it* in both

(10) Every man that owns a donkey that loves every child that feeds it is content.

and

(11) Every man that owns a donkey beats it.

(10) exhibits top-down propagation of binding scope, while (11), a prototypical "donkey" sentence, exhibits both top-down and bottom-up propagation of binding scope. [Chomsky, 1981] and [Reinhart, 1983] discuss top-down propagation of binding scope, using other terminology. [Hintikka and Carlson, 1979], [Kamp, 1981], [Heim, 1982], and [Barwise, 1986] discuss examples like (11). [Johnson and Klein, 1986] discusses an implementation of aspects of Kamp's account.

The binding module of QSB is based on a new account of pronominal bound variables that recognizes bottom-up propagation of binding scope, subject to localized requirements of existence and uniqueness. For example, the binding scope of *a donkey* in

(12) Pat owns a donkey, and Terry covets it.

can propagate up to the main clause and then down to *it*. However, such propagation is blocked by the negation operator in

(13) Pat doesn't own a donkey, and Terry covets it.

because of the localized existence requirement on bottom-up propagation. On a reading of (13) where the negation operator has higher scope than *a donkey*, the assertion of the existence of a donkey is not in force for the second conjunct. [Karttunen, 1969] discusses a variety of examples that illustrate the localized existence requirement.

(14) Pat owns every donkey, and Terry covets it.

where binding is impossible, illustrates the localized uniqueness requirement on bottom-up propagation of binding scope. No singled-out donkey is available for association with *it* in (14). Note that binding is possible in

(15) Pat owns every donkey, and Terry covets them.

but not in

(16) Many men own several donkeys, and Terry covets them.

on a reading where there can be different sets of several donkeys for different men, and where *them* is intended to identify a particular set of several donkeys owned by one man. *Every* in (15) in effect introduces a single level of multiplicity that is accommodated by the plural pronoun *them*. *Many* in (16) introduces a second level of multiplicity beyond the level introduced by *several*, and blocks the binding of the plural pronoun *them*. Similarly, binding of *it* by *a computer* is possible in (2) above, but not in (3). Note that bottom-up propagation of binding scope also works intersententially, as in

(17) Pat owns a donkey. Terry covets it.

Determination of possibilities for pronominal bound variables requires prior determination of possible relative quantifier scopes. For this and other reasons, the QSB binding module works on logical translations of natural language sentences. The current target language for translation is an enriched predicate logic. The next prototype will use a target language that more adequately captures meanings of natural language expressions. Binding scopes propagate bottom-up and top-down, from left to right. Binding is subject to agreement constraints, and to the following constraint, discussed in varying forms in [Keenan, 1974], [Chomsky, 1981] and [Hintikka and Kulas, 1983]: a quantifier cannot bind a pronoun and another variable within the minimal complete functional complex of the pronoun. This constraint disallows binding in such examples as

(18) Every man admires him.

The current binding module handles intrasentential binding of singular pronouns by universal and existential quantifiers. It finds binding possibilities with one pre-order pass through each logical translation. Total elapsed time for parsing and computation of normalized logical translations for

(19) Some pony expects every child to pet it and every man that knows every woman that owns a donkey covets it or some horse loves every child that feeds it.

with all possible pronoun bindings indicated, is 0.47 seconds (Symbolics 3645 Release 7.1). Of this time, 0.09 seconds is attributable to the parser module, 0.05 seconds is attributable to the scope module, and 0.04 seconds is attributable to the computation of binding possibilities.

Acknowledgments

This paper grew out of my ongoing collaboration with Joseph Aoun. I have also benefited greatly from interactions with colleagues at Bellcore. I received valuable comments on aspects of this work at the CUNY Human Sentence Processing Conference and during visits to MIT, Stanford, SRI, XEROX PARC, USC, and UCSD. Responsibility for errors that remain is entirely mine.

References

[Aoun and Li, to appear] Joseph Aoun and Audrey Li. *The Syntax of Scope*. MIT Press, Cambridge, Massachusetts, to appear.

[Aoun and Epstein, to appear] Joseph Aoun and Samuel S. Epstein. Relative quantifier scope. CUNY Conference on Human Sentence Processing, March 1988. Proceedings to appear.

[Barwise, 1986] Jon Barwise. Noun phrases, generalized quantifiers, and anaphora. Report number CSLI-86-52, CSLI, Stanford University, 1986.

[Berwick, 1987] Robert C. Berwick. Principle-based parsing. Technical Report Number 972, MIT Artificial Intelligence Laboratory, June 1987.

[Brennan *et al.*, 1987] Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics,* pages 155-162. July 1987.

[Chomsky, 1981] Noam Chomsky. *Lectures on Government and Binding*. Foris Publications, Dordrecht, 1981.

[Heim, 1982] Irene Heim. *The Semantics of Indefinite and Definite Noun Phrases.* Doctoral dissertation, University of Massachusetts, Amherst, 1982.

[Hintikka and Carlson, 1979] Jaakko Hintikka and Lauri Carlson. Conditionals, generic quantifiers, and other applications of subgames. In Franz Guenthner and S. J. Schmidt (editors), *Formal Semantics and Pragmatics for Natural Languages,* pages 1-36. D. Reidel, Dordrecht, 1979.

[Hintikka and Kulas, 1983] Jaakko Hintikka and Jack Kulas. Definite descriptions in game-theoretical semantics. In Jaakko Hintikka with Jack Kulas, *The Game of Language,* pages 137-160. D. Reidel, Dordrecht, 1983.

[Hobbs and Shieber, 1987] Jerry R. Hobbs and Stuart M. Shieber. An algorithm for generating quantifier scopings. *Computational Linguistics,* 13(1-2):47-63, January-June 1987.

[Johnson, 1987] Mark Johnson. The use of knowledge of language. MIT manuscript, 1987.

[Johnson and Klein, 1986] Mark Johnson and Ewan Klein. Discourse, anaphora, and parsing. In *Proceedings of the 11th International Conference on Computational Linguistics,* pages 669-675. International Committee on Computational Linguistics, August 1986.

[Kamp, 1981] Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof (editors), *Formal Methods in the Study of Language,* Part 1, pages 277-322. Mathematisch Centrum, Amsterdam, 1981.

[Karttunen, 1969] Lauri Karttunen. Discourse referents. International Conference on Computational Linguistics, 1969. Reprinted in in J. McCawley (editor), *Notes from the linguistic underground,* pages 363-385. Academic Press, New York, 1976.

[Keenan, 1974] Edward Keenan. The functional principle: generalizing the notion "subject of." In M. LaGaly, R. Fox and A. Bruck (editors), *Papers from the Tenth Regional Meeting of the Chicago Linguistic Society,* pages 298-309. Chicago Linguistic Society, Chicago, 1974.

[May 77] Robert May. *The Grammar of Quantification.* Doctoral dissertation, MIT, 1977.

[May, 1985] Robert May. *Logical form: its structure and derivation.* MIT Press, Cambridge, Massachusetts, 1985.

[van Riemsdijk and Williams, 1986] Henk van Riemsdijk and Edwin Williams. *Introduction to the Theory of Grammar.* MIT Press, Cambridge, Massachusetts, 1986.