# HANDLING SCOPE AMBIGUITIES IN ENGLISH

Sven Hurum

Department of Computing Science
615 General Services Building
University of Alberta
Edmonton, Canada T6G 2H1

## ABSTRACT

This paper describes a program for handling "scope ambiguities" in individual English sentences. The program operates on initial logical translations, generated by a parser/translator, in which "unscoped elements" such as quantifiers, coordinators and negation are left in place to be extracted and positioned by the scoping program. The program produces the set of valid scoped readings, omitting logically redundant readings, and places the readings in an approximate order of preference using a set of domain-independent heuristics. The heuristics are based on information about the lexical type of each operator and on "structural relations" between pairs of operators. The need for such domain-independent heuristics is emphasized; in some cases they can be decisive and in general they will serve as a guide to the use of further heuristics based on domain-specific knowledge and on the context of discourse. The emphasis of this paper is on discussing several of the more problematic aspects of the scoping protocol which were encountered during the design of the scoping program.

## INTRODUCTION

Natural languages contain a variety of "logical operators" which interact with each other to give rise to different types of ambiguity. The logical operators recognized by the scoping program include quantifiers, coordinators and negation, which are initially "unscoped" and must therefore be moved into position by the program, and adverbs, predicates and connectives (such as *if-then*). At the moment, other operators such as tense, aspect and modals are left in place and therefore assume innermost scope. There is some evidence that the handling of the scoping of quantifiers relative to such operators may require special treatment (eg. Fodor 1970; Enc 1981; Saarinen 1983).

Three simple examples will illustrate some different types of scope ambiguity and their representation in an informal first order predicate logic, using restrictions on quantifiers and an infix notation for sentential formulas. The meanings of the different interpretations should be clear. For example, (4) may mean that John didn't meet either Jane or Mary (5) or that he didn't meet at least one of them (6). Further examples are given in Hurum & Schubert (1986) and Hurum (1987). Some alternative proposals for representing scope ambiguities are also discussed in the latter.

(1) Someone loves everyone
(2) (∃x:person (∀y:person [x loves y]))
(3) (∀y:person (∃x:person [x loves y]))

(4) John didn't meet Jane or Mary
(5) ¬[[John met Jane] v [John met Mary]]
(6) [¬[John met Jane] v ¬[John met Mary]]

(7) Someone always comes late
(8) (∃x:person (always [x comes late]))
(9) (always (∃x:person [x comes late]))

Until quite recently, designers of natural language understanding systems have given little attention to the problem of dealing with scope ambiguities. Two of the earliest attempts to incorporate quantifier scoping into natural language understanding systems in an integral way are described in Woods (1978) and Dahl (1979). Some more recent scoping algorithms are presented in McCord (1981), Warren & Pereira (1982), Hobbs (1983), Saint-Dizier (1985) and Hobbs & Shieber (1987).

While each of these algorithms introduces some new features, certain problems, such as the scoping of coordinators and the use of heuristics to select preferred readings, have generally been given little or no treatment. Some of the main features of the algorithm being discussed here are: (a) it handles ambiguities created by quantifiers, coordinators, negation and adverbs,[1] (b) it works bottom-up and left-to-right and generates the set of valid scoped readings in one pass, (c) it removes logically redundant readings as they are encountered during the process of scoping and (d) it uses domain-independent heuristics, during the scoping, to arrange the readings in an approximate order of preference.

## LOGICAL REPRESENTATION

The scoping program is designed to be used as an extension to a parser/translator which generates initial translations in a first order modal logic augmented with certain operators (Schubert & Pelletier 1982). The operators being used include a generic kind forming operator, μ, and the

[1] Four types of coordinated expression are currently handled: noun phrases, noun complements, verbs and verb phrases. At the moment adverbs are treated as scoped (unmoved) elements.

operators $\alpha$ and $\tau$ which form functions and terms, respectively, from infix and prefix expressions. For example, the operators $\tau_1$ and $\tau_2$ map infix and prefix expressions, respectively, into terms.

The syntax of the logical translations has been chosen to simplify the mapping from the syntax (using a modified GPSG parser). A mixed infix/prefix notation has been used in order to keep the logical form as close as possible to the surface form. Two examples of the initial logical translations being used are shown below. Unscoped operators, which are to be extracted and positioned by the scoping program, are placed in angled brackets; the square, curly and round brackets signify infix (sentential), prefix (predicative) and functional expressions, respectively. The suffixes which are attached to each word to mark their surface position are not shown here.

(10) Many people visit Europe every month
(11) (($\alpha_2$ {during <every month>})
[<many person> (PRES {visit Europe})])

(12) That John didn't arrive surprised Jane and Mary
(13) [($\tau_1$ [John <not (PAST {arrive})>])
(PAST {surprise <and Jane Mary>})]

A sample of the output from the program is shown in the Appendix. The two sentences shown are

1. All men want to marry Peggy or Sue
2. Mary (read or told some story to each child)

The output for each sentence consists of an echo of the input formula followed by a list of scoped readings ordered according to their average scoping weight (see below). In the LISP notation, the prefixes *i, p, f, q* and *c* are used to mark infix, prefix, functional, quantified and coordinated expressions. The first sentence is taken from Schubert & Pelletier (1982) which gives a description of the three interpretations. The second sentence has been parsed as having a verb phrase ambiguity (indicated by the brackets) and the input formula therefore contains two duplicated operators. The two comparisons made are *each/some* and *each/or*. No comparison is made between the commutative operators *some* and *or*.

## COORDINATED EXPRESSIONS

The scoping of coordinated expressions poses several problems. One problem is how to avoid the "vacuous" quantification or coordination which may result whenever a coordinated expression contains an unscoped operator. For example, if the indefinite *some blonde* in (14) is applied to the clause before the coordinator, the subsequent application of the latter will result in vacuous quantification (15).

(14) John met Mary or some blonde
(15) [($\exists$x: blonde [John {met Mary}]) or
($\exists$x: blonde [John {met x}])]

Similarly, "vacuous coordination" may result when nested coordinators are present. This problem could be avoided by scoping in several passes, in each pass scoping only operators which are not embedded inside a coordinator. However, this would considerably complicate the scoping algorithm and would also violate the principle of applying the innermost operator first.

A second problem is how to handle the scoping of multiple copies of the same operator which may occur when the operator is embedded inside a coordinated expression. This problem is unavoidable when it results from the parser; for example, (16) may be parsed and initially translated into (17). The brackets signify that the sentence has been parsed as having a VP coordination.

(16) John (hopes and intends to buy a boat)
(17) [John <and (PRES {hope
($\tau_2$ (INF {buy <$a_1$ boat>})))})
(PRES {intend
($\tau_2$ (INF {buy <$a_1$ boat>})))}) >]

Three constraints on a duplicated operator such as $a_1$ are that (a) it must scope consistently with respect to all other operators, (b) it must only be compared once to each other operator (for the purposes of computing the preferred scope orderings) and (c) if it scopes outside a coordinator which initially embeds it, only one copy of the operator can be carried up. This poses a problem for bottom-up approaches to scoping since some global knowledge is needed to ensure the consistency of the scoping of duplicated operators inside the different expressions in which they occur. It therefore is necessary to use some overhead to keep track of the scope relations of operators which are present in multiple copies, and to store this information separately for each reading.

Duplication of operators may also occur during the scoping process. For example, the application of one of the coordinators in

(18) John and Bill visited Spain or Morocco

will result in the duplication of the other. At present, the scoping program avoids this problem, as well as the problem of vacuous quantification, by using a "branch-trimming" function which removes incorrectly embedded operators from the different branches of a coordinator at the time of applying the coordinator. This function is simple to use but does involve some extra overhead. The problem of duplication resulting from the parser is handled by labelling readings and by storing on the property list of each duplicated operator a list of the operators having been scoped inside and outside the operator.

A third problem is how to treat unscoped operators inside "coordinated predicates". In example (16) it seems evident that the indefinite *a boat* cannot have both opaque and transparent interpretations in the same reading. That is, assuming that the opaque/transparent distinction is to be represented in terms of scope, then both copies of the indefinite must scope consistently relative to the two coordinated predicates *hope* and *intend*. Since the two predicates are distinct, and therefore should be allowed to scope independently with *a boat*, the current version of the program contains a special constraint which forces coordinated predicates to scope consistently relative to all duplicated operators embedded inside them. This rule could be treated as a heuristic rather than as a constraint, but the rule does seem to be absolute.

In contrast, there is a general, but not absolute, preference for "symmetric" interpretations whenever coordinated expressions contain similar but not identical pairs of operators. For example, in (19) one could imagine a context in which it is made clear that Sue, but not Mary, has a particular hat in mind and in (20) it is possible, though very improbable, that the two indefinites have different functional dependencies.

(19) John (knows that Sue wants) and (thinks that Mary hopes) to buy a new hat
(20) Mary read a story to each child or told a story to each child

At present, the program does not adequately handle this preference for symmetric readings, which requires some non-local heuristic knowledge.

## REDUNDANT READINGS

A test is made for logically redundant readings whenever an unscoped operator is about to be positioned (applied to a clausal expression). A reading is considered to be redundant if two commutative operators are applied consecutively and the suffix of the outer operator is greater than that of the inner one. (Suffixes are attached to words by the parser to mark their position in the original sentence). If one of the operators is a coordinator the criterion used is that the quantifier should scope inside the coordinator. Readings will also be removed if they contain an ordering of a pair of operators which has a scoping weight less than a preset parameter.

## SCOPING WEIGHTS

In order to quantify scoping preferences, we associate a "scoping weight", a value between 0 and 1, with each pair of interacting operators. The weight indicates the preference for the reading in which the second operator (in surface order) scopes outside the first one. For example, the value 0.9 indicates a strong preference for the reading in which the second operator takes wide scope, a preference which might, on occasion, be overridden by pragmatics. The weight associated with the reverse ordering will automatically be 0.1. The value 0.5 indicates an equal preference for both scope orderings in a pragmatically neutral context. The following examples illustrate how the scoping weights are used.

| | |
|---|---|
| (21) Some person on each team was injured | .9 |
| (22) Some person playing on each team was injured | .5 |
| (23) Some person who plays on each team was injured | .02 |

As the scoping weights indicate, the ability of the embedded quantifier *each team* to widen scope over *some person* decreases as the embedding phrase changes from a prepositional phrase (21) to a verb phrase (22) to a full clause (23). This "embedding hierarchy" was pointed out by van Lehn (1978) and also holds for phrases serving as adverbials or as terms.

The scoping weights used by the program have been derived from the examination of a large number of sentences such as these. An attempt was made to keep the sentences as pragmatically neutral as possible and to try to obtain a domain-independent weight for pairs of operators in a given "pattern", where a pattern is a combination of two operators of given types and in a given structural relation to one another. Although the data reflect the intuitve judgements of the author, it is likely that there would be a good general agreement in cases in which there is a strong preference for one ordering. In other cases, the need to include pragmatic knowldege would be more important. Some consideration was also given to the empirical data on scoping preferences described previously (eg. Ioup 1975, van Lehn 1978, Gil 1982).

Given that we can determine scoping weights for pairs of operators, it is still necessary to combine these to arrive at an overall rating of a reading. This involves two separate problems: how to select pairs of operators for comparison and then how to combine the weights obtained. There appears to be no obvious solution to either of these problems. There are at least three different choices which need to be made when picking a strategy for selecting pairs of operators for comparison, none of which is clearcut. For example, if a sentence contains three quantifiers at the same level, such as a subject and two objects, should all three pairs of quantifiers be compared or should the results of each comparison made be used to reduce the number of further comparisons needed?

There is also no obvious way to combine the scoping weights obtained. A probabilistic treatment is not feasible, in part because different readings of a sentence may involve different numbers of comparisons. The simplest method is to order the readings according to their average scoping weight and this appears to give quite good results. The major drawback to this method is that it tends to smooth out the effect of very low individual weights. However, there are ways to minimize this problem. At present, a parameter is used to specify the minimal acceptable scoping weight so that readings with very low pairwise orderings can be removed. Alternatively, readings could be tagged with their lowest weights and some readings later be set aside or some more complex function could be used for combining the scoping weights. These problems are discussed in Hurum (1987).

## HEURISTICS BASED ON LEXICAL TYPES

The domain-independent heuristics are based on two types of information: the lexical type of each operator and structural relations between pairs of operators. Some heuristics are defined for individual lexical types, such as *each, some* and *or*, and others for classes of individual types, such as universal or existential quantifiers. Most of the heuristics used by the program are stored in a table of scoping weights. To minimize the amount of data, universal and existential quantifiers are sometimes represented by the "standards" *each* and *some* and other members of these classes are then related to the standards by ratios. Most, but not all, of the heuristics described here are currently being used by the program.

The universal quantifiers may be arranged in the hierarchy *each* > *every* > *all* in terms of the tendency to take wide scope. This hierarchy has been mentioned by both Ioup and van Lehn and a number of people have commented that the function of *each* in English may partly be to indicate the distributive (ie. wide scope) reading. Universal quantifiers have a surpringly marked tendency to scope inside a negation (24-26) given their usual tendency, with the exception of *both*, to take wide scope:

| | |
|---|---|
| (24) All people aren't happy | .6 |
| (25) John didn't win every race | .2 |
| (26) John didn't win both races | .1 |

Non-universal quantifiers in the subject position of a negated sentence seldom scope inside the negation.

*Few* and *no* have very little ability to widen scope over a preceding operator but, in contrast, have a strong tendency to trap subsequent operators. Therefore, a distinction needs to be made between the ability to widen scope over a preceding operator and to trap subsequent operators. The following examples show the scoping of *few* and *no* relative to quantifiers (27,28), the negation operator (29) and temporal adverbs (30). By comparison, *some* does not create a strong trap for *always* (31).

| | |
|---|---|
| (27) Nobody read every article | .02 |
| (28) Someone read no articles | .02 |
| (29) Few people weren't surprised | .01 |
| (30) Few people always come late | .01 |
| (31) Someone always comes late | .5 |

There appear to be some sentences, typically containing two *no* or *few* quantifiers, which are used in a sense which does not appear to correspond to any straightforward ordering of the quantifiers. Instead, the total quantity of predications being made seems to be emphasized. An example is given in (35). One possible way of representing such sentences might be to use branching quantification (Hintikka 1974).

(35) Few boys kissed few girls

Sentences containing operators which create negated contexts (eg. *few, no, not, never*) are often disambiguated by the presence of *any, ever* ("at any time") or *neither-nor*. For example, after *few* or *no* the adverb *sometimes* is usually replaced by *ever* (32,33) and the wide-scope reading of *never* in (34) is best obtained by replacing the *or* with *and* or by using *neither-nor* and *ever* (35).

| | |
|---|---|
| (32) ?Few people sometimes come late | .1 |
| (33) Few people ever come late | .0 |
| (34) (Either) John or Bill never comes late | .05 |
| (35) Neither John nor Bill ever comes late | .0 |

The singular indefinite *a* is quite consistently more likely than *some* to take narrow scope. For example, it would be more natural to use (36) and (38) than (37) and (39) to indicate the narrow scope existential reading. Also, (40) is acceptable but (41) is not. (The scoping weights given for (40) and (41) have not been adjusted to take into account the effect of the modifier *different*).

| | |
|---|---|
| (36) Each person grabbed a chair | .3 |
| (37) Each person grabbed some chair | .5 |
| (38) John didn't find a chair | .3 |
| (39) John didn't find some chair | .6 |
| (40) A different person brought each chair | (.7) |
| (41) *Some different person brought each chair | (.5) |

The scoping of sentences containing the determiner *a* may be complicated by the presence of generic interpretations. For example, in (42) the non-specific reading could be obtained either by giving *never* wide scope or by treating *a guest* as a quasi-universal quantifier (derived from the generic interpretation via meaning postulates). Assuming that the generic reading is present, the standard interpretation in which *never* has wide scope must be treated as being either absent or logically redundant. (This is an oversimplified view; some attempts are currently being made to give a uniform interpretation to indefinites which would avoid this problem of redundancy). A somewhat similar problem arises when indefinites which may have generic interpretations are present inside the antecedent clause of an *if-then* sentence (as in certain donkey sentences). Note that there is no comparable reading when *a* is replaced by *some*, which does not receive a generic interpretation (43).

| | |
|---|---|
| (42) An old sailor never gets seasick | .5? |
| (43) Some old sailor never gets seasick | .01 |

Plural indefinites can be placed in an approximate hierarchy in terms of their ability to receive collective interpretations: *some* > *three* > *several* > *many*. This correlates with their ability to be given "specific" interpretations and therefore with their ability to widen scope from strong clausal scope traps (44,45) and perhaps also, to a lesser extent, relative to the negation operator (46,47). The scoping weights shown are associated with the scoping of the existentially quantified collections.

| | |
|---|---|
| (44) If three people show up then I will come | .3 |
| (45) If many people show up then I will come | .05 |
| (46) John didn't find three chairs | .4 |
| (47) John didn't find many chairs | .2 |

Plural indefinites may have implicit universal partitives associated with them (see Hurum & Schubert 1986) and, when present, these must be scoped separately. While the existential quantifiers associated with indefinites are free to scope to any position, in the absence of pragmatic information, there are considerable restrictions on the ability of plural indefinites to distribute over preceding operators. For example, plural indefinites in the object position almost never distribute over quantifiers in the subject position unless preceded by an explicit partitive.

## HEURISTICS BASED ON STRUCTURAL RELATIONS

Scoping preferences are strongly influenced by "structural relations", that is, the relations between pairs of operators in the initial logical translations (or, approximately, in the parse tree). Structural relations may be loosely classified as "horizontal", an example being the *subject-object* relation, or "vertical", an example being the relation between a noun phrase determiner and an operator inside the noun complement. Although this distinction is not always clearcut, the scoping program makes considerable use of it and separate heuristics are used for horizontal and vertical relations.

As a general rule in English, scope order tends to follow surface order, although there are some exceptions such as in the case of postposed adverbials. The effect of surface order is strengthened considerably by "shifting", where shifting is used here in a general sense to include the preposing of adverbials, topicalization and perhaps the dative shift. For example, it is much more likely that (48) refers to a different set of people each year than (49) and the distributive reading is more likely in (50) than in (51).

| | |
|---|---|
| (48) Every year many people visit Europe | .02 |
| (49) Many people visit Europe every year | .5 |

| | |
|---|---|
| (50) Every sailor gave flowers to two girls | .3 |
| (51) To two girls, every sailor gave flowers | .3 |

It should be pointed out that Ioup (1975) has presented evidence that in a wide range of languages "grammatical function" (eg. subject, direct object, ...) may be a more important determiner of scope than surface order. (Ioup considers "topic" to be a grammatical category rather than a result of shifting.) It happens that in English there is a close correlation between surface order and scope order. However, it would always be possible, if necessary, to reinterpret some of the heuristics shown here in terms of grammatical relations rather than in terms of surface order.

The effect of surface order and shifting also appears to hold for temporal adverbs, although the interaction of quantifiers with such adverbs can sometimes be quite complex. In the case of negated quantifiers (eg. *no, few*) and *not* the effect of surface order is again quite decisive, with the exception of certain postposed adverbs (see below):

| | |
|---|---|
| (52) Often, nobody is late for lunch | .01 |
| (53) Nobody is often late for lunch | .0 |

The effect of shifting can also be seen with existential quantifiers. The following examples show the scoping of the existential quantifier associated with *many* relative to *often* in preposed, medial and postposed positions. The effect of adverb placement is clear, although the scoping of postposed adverbs will be radically different depending on such factors as the pronounciation or the presence or absence of a comma (56,57).

| | |
|---|---|
| (54) Often, many people are late for lunch | .02 |
| (55) Many people are often late for lunch | .5 |
| (56) Many people are late for lunch often | .1 |
| (57) Many people are late for lunch, often | .98 |

The principal ambiguity in these sentences is related to whether or not the same group of people is being referred to in each situation (we may loosely interpret *often* as quantifying over instances of a type of situation, in this case a lunch setting). This ambiguity can be represented by scoping the existential quantifier associated with *many* relative to *often*. It is very unlikely that we would give *many* wide scope in (54) although this would be more likely with indefinites which can more easily receive specific interpretations, such as *some, three* and *several*.

There is also an optional universal partitive associated with plural indefinites such as *many* and this must also be scoped. The interaction of universal quantifiers with temporal adverbs involves some quite subtle ambiguities which are related to whether or not all members of some collection are involved in the same situation. However, the effect of surface position is still notable:

| | |
|---|---|
| (58) Often, everyone is late for lunch | .02 |
| (59) Everyone is often late for lunch | .5 |

Different types of embedding construct form quite consistent traps for quantifiers and other unscoped operators. Operators inside prepositional phrases generally widen scope over the head quantifier, those inside full clauses almost never do (with the exception of specific indefinites) and those inside bare verb phrases have an intermediate tendency to do so (see (21)-(23)). Verb phrases serving as noun complements form considerably weaker traps than do those serving as nominalized arguments. Preposed antecedent clauses of connective sentences such as *if-then* sentences appear to form absolute traps for distributive quantifiers, in contrast to consequent or postposed antecedent clauses, and for connective clauses in general the ordering of the antecedent and consequent clauses needs to be considered.

The effect of structural relations on the scoping of quantifiers generally holds for coordinators as well. Some examples will illustrate the effect of the surface position of NP coordinators relative to negation (60,61) and to quantifiers (62,63). The presence of *either*, by emphasizing the disjunction, tends to widen the scope of *or* somewhat.

| | |
|---|---|
| (60) (Either) Sue or Mary didn't dance with John | .2 |
| (61) John didn't dance with Sue or Mary | .2 |

| | |
|---|---|
| (62) Few people danced with Sue or Mary | .2 |
| (63) (Either) Sue or Mary danced with few people | .2 |

Verb coordinators usually scope inside quantifiers in the subject and object positions. For quantifiers in the subject position this is clearly a structural constraint; in both (64) and (65) the subject presumably scopes outside the coordinator and it is difficult to reverse this ordering by passivization or by replacing the subject with *someone different*. By contrast, the examples show that the scoping of a direct object relative to a verb coordinator is largely dependent on pragmatics.

(64) Someone wrote and mailed a letter
(65) Someone wrote and received a letter

However, there is probably some bias, which might be considered structural, for scoping an object outside a verb coordinator, and this bias is stronger for prepositional objects and for *some* (66). It is always possible for *or* to take wide scope, both relative to subject and object quantifiers, although the latter is more likely (67,68). This is the "speaker's uncertainty" reading. Although always present, it is particularly difficult to get this reading with *few* or *no* in the object position.

(66) John drove and flew to some resort .8

(67) John flew or drove to each resort .7
(68) Each person drove or flew to the resort .1

The interaction of plural quantifiers with verb conjunction is more complex and we make a distinction between primary and secondary scope dependencies: the former involves the scoping of the collection formed from the plural quantifier and the latter the details of the predications of individual members of the collection. For example, in (69) there is presumably only one set of two people, meaning that the collection formed from the subject scopes outside the coordinator. The details of the individual predications can be specified later. In general, some members of the set might be involved in both predications and some in just one. This type of interaction between sets is similar to that between two plural quantifiers. The conjoined subject in (70) could also intitially be treated as a collection, or the ambiguity might in this case be handled directly by the parser.

(69) Two different people painted and redecorated the apartment
(70) John and Fred, respectively, fixed and upholstered the chair

The scoping of noun coordinators is somewhat similar to that of verb coordinators, although there is evidence that the wide scope *and* reading is elliptical for a NP coordination and should therfore be handled by the parser. Therefore, the scoping weights for (71)-(74) have been placed in brackets. The "scoping" of *and* relative to a singular indefinite is again largely dependent on pragmatics (71,72), although there is probably a statistical bias in favour of the wide scope (elliptical) *and* reading. Again, this reading is less likely when *a* is replaced with (singular) *some*.

(71) A man and woman came to help (.5)
(72) A friend and colleague came to help (.5)

Plural indefinites also display two levels of interaction with coordinated nouns. The initial scope (or syntactic) ambiguities of (73) and (74) again are related to whether there is one collection or two: the latter (meaning a wide scope *and*) is pragmatically more likely in (73) simply because we wouldn't use this wording to refer to a man and a woman. The details of applying the predicates to members of the collection can again be postponed until later. (A scoping program clearly needs to have some specialized knowledge for handling interactions between sets of objects and predicates.)

(73) Two men and women arrived (.5)
(74) Twenty men and women arrived (.5)

Again, *or* tends to be trapped (75) and the trap is especially strong with *no* and *few* (76).

(75) Every freshman or sophomore finished the course .2
(76) Few freshmen or sophomores finished the course .05

Coordinators are treated as forming complete scope traps except for existential quantifiers and *or*. This useful rule removes the three unwanted readings of (77) in which either or both of the universal quantifiers take wide scope. It may prevent the anaphoric binding of pronouns, as in (78), but this is part of a more general problem for which there is no satisfactory theory at the moment (see Lepore & Garson 1983; Schubert & Pelletier 1987a,1987b; Hobbs & Shieber 1987).

(77) Every man or every woman arrived late
(78) Every man or some friend of *his* arrived late

Like universal quantifiers, *and* tends to be trapped by clausal embedding whereas *or*, though less easily than existential quantifiers, can generally widen scope from strong scope traps such as "scope islands". For example, there is no reading of (79) in which *and* scopes outside *someone*, that is in which there is a different person for Sue and Mary, but in (80) there is a reading, probably the preferred one, in which *or* must scope outside the nested clause, meaning that each person has either heard that his aunt or that his uncle is arriving. There is also a reading of (80), perhaps not obvious at first, in which *or* has maximally wide scope, meaning that the speaker is not sure whether it was his aunt or his uncle that each person heard was arriving.

(79) Someone heard the news that Sue and Mary were arriving
(80) Each person heard the news that his aunt or his uncle was arriving

## IMPLEMENTATION OF HEURISTICS

Each formula (given an input list of sentential formulas) is traversed in a bottom-up left-to-right order with different types of expression, such as infix, prefix and coordinated expressions, being scoped by separate procedures. As each unscoped operator is encountered, its structural category is stored on its property list and this information is later used to determine the structural relation between a given pair of operators. Vertical relations are passed as parameters to subordinate procedures; in the case of *if-then* sentences the parameters also contain information about the position and type of clause being scoped (eg. "preposed antecedent clause").

The scoping weight for a pair of operators can then be determined from a table of weights which is indexed according to structural relations and operator types. The table has been kept as small as possible by the use of default values and "standard" operator types (such as *some* and *each* for existential and universal quantifiers). Although the use of a table of weights does not in itself have much psychological plausibility, the rules on which the table is based, such as those described above, are generally quite simple and it is hoped that rules such as these can eventually be incorporated into a more comprehensive model of the grammatical biases which underly scope preferences.

## PRAGMATICS

The most obvious place to try to combine pragmatic and domain-independent information is at the level of determining the pairwise scoping weights. The problem of how these weights should then be combined still remains but this approach does seem to be worth pursuing. Since properly applied pragmatic knowledge will often result in strong, if not absolute, preferences for certain scope orderings, the chances of selecting the best overall reading of a sentence will be improved when pragmatic heuristics are added. The ability of pragmatic knowledge to veto certain scope orderings can quite easily be implemented by setting the appropriate scoping weight below the value of the *min-weight* parameter which will automatically disallow any readings containing such orderings.

## CONCLUSION

This paper has described some features of a program designed to handle scope ambiguities in English. Some of the more problematic issues which were encountered during the designing of the program were selected for discussion: the choice of logical representation, the scoping of coordinated expressions, the choice of a strategy for selecting preferred scope orderings and the determination of a set of domain-independent heuristics. The program is currently being extended to include a wider range of lexical types and input expressions and the heuristics are being improved. Following this, it is hoped to incorporate some simple types of domain- and discourse-dependent knowledge into the program, in particular knowledge about expected relations among objects in a given domain and a simple discourse focus structure.

The selection of preferred scope orderings depends on the complex interaction of linguistic and context-dependent knowldege. It would be a considerable advantage to be able to factor out the contributions of different types of knowledge required and then at some later time to combine them. One conclusion of this work is that there is a body of largely domain-independent knowledge which can play an important, and at times decisive, role in the disambiguation of scope. Such knowledge is most useful when it indicates a very strong or absolute preference for one reading.

Absolute preferences typically occur with operators such as *any* or *both* and with distributive quantifiers or *and* inside strong clausal trap or inside a coordinator. Very strong preferences may occur with operators such as *few*, *no* or *each*, with preposed or topicalized operators and with operators inside prepositional phrases. When the domain-independent heuristics do not provide a strong preference for one reading, they may still serve as a useful guide guide for the later application of pragmatic knowledge. This is commonly the case when indefinites are present, as the "specificity" of indefinites is mainly context-dependent.

A number of problems have not been discussed here because they remain unresolved. These include: the scoping of quantifiers relative to tense and opaque operators, the logical representation and scoping of generics, the treatment of pronouns not embedded within their quantifier antecedents, non-local problems such as the preference for "symmetric" readings, the use of stray words, such as *together* and *both* (as an adverb), which provide important clues for preferred scope relations and the difficult problem of combining linguistic and context-dependent heuristic knowledge.

## ACKNOWLEDGEMENTS

I would like to thank Dr. Len Schubert for originally suggesting this project and for his advice and many helpful comments throughout the course of this work. I would also like to thank the members of my thesis committee and the members of the Logical Grammar Study Group at the University of Alberta for their comments on parts of this work. This work was supported in part by NSERC Operating Grant A8818.

## REFERENCES

Dahl, V. (1979), "Quantification in a Three-Valued Logic for Natural Language Question-Answering Systems", *Proceedings of the Sixth International Joint Conference for Artificial Intelligence*, 182-187.

Enc, M. (1981), *Tense Without Scope: An Analysis of Nouns as Indexicals*, unpublished Ph.D. Dissertation (University of Wisconsin, Madison).

Fodor, J.D. (1976), *The Linguistic Description of Opaque Contexts*, Ph.D. Dissertation, available from Indiana University Linguistics Club.

Gil, D. (1982), "Quantifier Scope, Linguistic Variation, and Natural Language Semantics", *Linguistics and Philosophy* 5, 421-472.

Hintikka, J. (1974), "Quantification vs. Quantification Theory", *Linguistic Inquiry* 5, 153-177.

Hobbs, J.R. (1983), "An Improper Treatment of Quantification in Ordinary English", *Proceedings of the Twenty-First Annual Meeting of the Association for Computaional Linguistics*, 57-63.

Hobbs, J.R. & S.M. Shieber (1987), "An Algorithm for Generating Quantifier Scopings", (In preparation).

Hurum, S. (1987), *Quantifier Scoping in Initial Logical Translations of English Sentences*, M.Sc. thesis, University of Alberta, 1-242.

Hurum, S. & L.K. Schubert (1986), "Two Types of Quantifier Scoping", *Proceedings of the Sixth Canadian Conference on Artificial Intelligence"*, 39-43.

Ioup, G. (1975), "Some Universals for Quantifier Scope", in J.P. Kimball (ed.), *Syntax and Semantics*, Vol. 4, (New York: Academic Press), 37-58.

Lepore, E. & J. Garson (1983), "Pronouns and Quantifier-Scope in English", *Journal of Philosophical Logic* **12**, 327-358.

McCord, M.C. (1981), "Focalizers, the Scoping Problem and Semantic Interpretation Rules in Logic Grammars", *Proceedings of the International Workshop on Logic Programming for Expert Systems*, Logicon, (Woodland Hills).

Saarinen, E. (1980), "Quantifier Phrases are (at Least) Five Ways Ambiguous in Intensional Contexts", in F. Heny (ed.), *Ambiguities in Intensional Contexts*, (Dordrecht: Reidel), 1-45.

Saint-Dizier, P. (1985), "Handling Quantifier Scope Ambiguities in a Semantic Representation of Natural Language Sentences", in V. Dahl & P. Saint-Dizier (eds.), *Natural Language Understanding and Logic Programming*, (North-holland), 49-63.

Schubert, L.K. & F.J. Pelletier (1982), "From English to Logic: Context-Free Computation of 'Conventional' Logical Translation", *American Journal of Computational Linguistics* **8**, 26-44. Reprinted (with corrections) in B.J. Grosz, K. Sparck-Jones & B.L. Webber (eds.), *Readings in Natural Language Processing*, (Los Altos: Morgan Kaufman), 1986.

Schubert, L.K. & F.J. Pelletier (1987a), "Problems in the Representation of the Logical Form of Generics, Plurals and Mass Nouns", in E. Lepore (ed.), *New Directions in Semantics*, (Academic Press).

Schubert, L.K. & F.J. Pelletier (1987b), "Generically Speaking, With Remarks on the Interpretation of Pronouns and Tenses". To appear in G. Chierchia, B. Partee & R. Turner (eds.), *Property Theory, Type Theory, and Semantics*, (Dordrecht: Reidel).

Van Lehn, K. (1978), *Determining the Scope of English Quantifiers*, MIT Artificial Intelligence Laboratory Technical Report AI-TR-483, 1-123.

Warren, D.H.D. & F.C.N. Pereira (1982), "An Efficient Easily Adaptable System for Interpreting Natural Language Queries", *American Journal of Computational Linguistics* **8**, 110-119.

Woods, W.A. (1978), "Semantics and Quantification in Natural Language Question Answering", *Advances in Computers*, vol. 17, (New York: Academic Press), 1-87.

## APPENDIX: OUTPUT FROM SCOPING PROGRAM

**Sentence 1**

```
(i (q all1 man2)
  (f PRES (p want3 (TAU2 (f INF (p marry4 (c or6 Peggy5 Sue7)))))))
```

1. The average weight is 0.7 based on 1 comparison

```
(q all1
  y5
  (i y5 man2)
  (i y5
    (f PRES
      (p want3
        (TAU2
          (l y10
            (i (i y10 (f INF (p marry4 Peggy5)))
              or6
              (i y10 (f INF (p marry4 Sue7))))))))))
```

2. The average weight is 0.5 based on 2 comparisons

```
(q all1
  y5
  (i y5 man2)
  (i (i y5 (f PRES (p want3 (TAU2 (f INF (p marry4 Peggy5))))))
    or6
    (i y5 (f PRES (p want3 (TAU2 (f INF (p marry4 Sue7))))))))
```

3. The average weight is 0.3 based on 2 comparisons

```
(i (q all1
    y5
    (i y5 man2)
    (i y5 (f PRES (p want3 (TAU2 (f INF (p marry4 Peggy5)))))))
  or6
  (q all1
    y5
    (i y5 man2)
    (i y5 (f PRES (p want3 (TAU2 (f INF (p marry4 Sue7))))))))
```

time used = 308 msecs.

---

**Sentence 2**

```
(i Mary1
  (c or4
    (f PAST (p read3 (q some6 story7) (q each8 child9)))
    (f PAST (p tell5 (q some6 story7) (q each8 child9)))))
```

1. The average weight is 0.82 based on 2 comparisons

```
(i (q each8
    y17
    (i y17 child9)
    (q some6 y15 (i y15 story7) (i Mary1 (f PAST (p read3 y15 y17)))))
  or4
  (q each8
    y17
    (i y17 child9)
    (q some6 y15 (i y15 story7) (i Mary1 (f PAST (p tell5 y15 y17))))))
```

2. The average weight is 0.67 based on 2 comparisons

```
(i (q some6
    y15
    (i y15 story7)
    (q each8 y17 (i y17 child9) (i Mary1 (f PAST (p read3 y15 y17)))))
  or4
  (q some6
    y15
    (i y15 story7)
    (q each8 y17 (i y17 child9) (i Mary1 (f PAST (p tell5 y15 y17))))))
```

time used = 386 msecs.

---