5

COMPUTING QUANTIFIER SCOPE*

Edward P. Stabler

1 A PUZZLE

There is a quantifier scope ambiguity in (1). In addition to the preferred normal scope reading paraphrased in (1ns), this sentence has the inverse scope reading paraphrased in (1is):

(1) Some linguist speaks every language.

(1ns) There is some linguist x such that x speaks every language

(1is) For every language y, some linguist or other speaks y

Liu (1990) and others point out that certain objects, such as those with decreasing denotations, do not allow an inverse scope reading, as in:

(2) Some linguist speaks at most 2 languages.

(2ns) Some linguist x is such that x speaks at most 2 languages

(2is) There are at most 2 languages y such that some linguist or other speaks those 2 languages y

(2is) is perfectly intelligible: it says that linguists speak at most 2 languages altogether. This does not seem to be available as an interpretation of (2). This is arguably not just a preference; sentence (2) just cannot be interpreted as (2is).

If it is true, as it seems, that a certain semantically identified collection of quantifiers does not allow inverse scope readings, how can this be explained? If there is a mechanism for "raising" quantifiers, how could a semantic property like decreasingness be relevant to whether the mechanism can apply? One might think that the semantic values of the structures computed in any computational

*This work was inspired by stimulating discussions with Anna Szabolcsi, Filippo Beghelli, Fernando Pereira, and especially Dorit Ben-Shalom and Ed Keenan.

model of the language user could have no more relevance than the dollars do when a calculator computes your bank balance.[1] But the puzzle is resolved when we remember the most basic insight about computational systems: formal properties can reflect semantic properties. In a computational model of the language user, to make sense of generalizations like the one mentioned above, we need a theory in which the semantic generalizations are revealed to hold in virtue of the formal representation of quantifiers.

Suppose that we just subcategorize the quantifiers according to their scoping behavior, thereby distinguishing them syntactically. While this allows scoping distinctions to be captured formally in a computational approach, it still misses something in the generalization noted above. That generalization does not merely identify the distinctive scoping behavior of certain classes of quantifiers, but also relates that behavior to the semantic values of the quantifiers in these classes. Putting the matter this way, it is easy to see how the remainder of the puzzle must be resolved. The way to capture this semantic generalization is to elaborate the representational account of scope in such a way that the very representational features that determine the scope of quantifiers also determine fundamental aspects of their inferential role in the language, and hence fundamental aspects of their meaning such as decreasingness. This paper provides a preliminary account of this sort.

Providing such an account has become more challenging in some recent derivational approaches to syntax. If some lexical item has a syntactic requirement which is met in the course of a derivation, there may be no need to assume that the requirement is in some significant way still present in the derived structure. A "checked" or fulfilled requirement that has no further role may be regarded as a deleted syntactic feature. This perspective on syntactic derivation, according to which key features of the structures are deleted in the course of a derivation, apparently conflicts with the idea that derived syntactic structure is the "output" of linguistic analysis and the "input" to later cognitive processes. This tension between what is required for the syntax and what is required for semantic interpretation and later cognitive processes is quite clear in recent debates about the role of chains, for example. Brody (1995) argues that because chains are needed for interpretation, we should assume they are available in the syntax, and hence an additional derivational notion of

[1] This seems to be exactly what Chomsky (1995a) is puzzled about when he insists that semantic properties cannot be relevant to an account of a language user, any more than an intentional relation to the earth is really relevant when we say "the meteorite is aiming for the earth." A certain puzzlement about the role of semantic properties in linguistics is natural, and a resolution of the puzzle is sketched here. It is no surprise that there are no significant generalizations about meteorite trajectories stated in terms of what they are aiming for, and hence no comparable puzzle about semantic generalizations in astronomy. Meteorites do not represent targets to themselves the way language users represent things to themselves and each other.

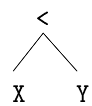
movement becomes unnecessary. But this argument is unsound. If chains are needed for interpretation but not needed to condition the syntactic derivation, the appropriate conclusion is that interpretation depends on aspects of linguistic structure to which the syntax is insensitive. This sort of perspective on semantic values is familiar in categorial grammar and certain other traditions, and is quite natural in transformational grammar as well.

To illustrate how this whole picture could work, a very simple illustrative syntax is formalized in §2, incorporating a simple version of the theory of quantifiers proposed in Beghelli and Stowell (1996) and Szabolcsi (1996). Following Chomsky (1995b), the structures defined in this simple syntax have some properties that are syntactically relevant, and others that are semantically relevant. A preliminary compositional semantics is provided for the structures with semantically relevant properties; and sound inference patterns are defined in which the very features that determine scoping options thereby determine inferential role. That is, for this simple language, we have a purely formal account that conforms to semantic generalizations of the kind mentioned above. In §3 some basic assumptions of this approach are identified and contrasted with alternative approaches.

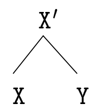
2 A SIMPLE GRAMMAR

A simple example will illustrate how the various pieces of a story about human language could go together to explain semantic generalizations about quantifier scope. This example is not intended to be an adequate representation of English, or even of just those English-like constructions that it generates. But it will be clear in this fragment how syntax, semantics and inference are related in such a way as to allow quantifiers to be used in semantically appropriate ways by a computational system. The proposal is that some such relation is found in human language, and that this is how the semantic generalizations about quantifier scope will ultimately find their explanation. Relevant aspects of the fragment will be discussed in a more general way in §3.

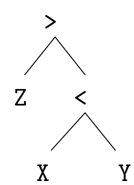
Before presenting the grammar, we informally sketch the basic assumptions. Following Koopman (1994) and Sportiche (1995) we assume that all syntactic requirements must be satisfied either by head movement or by an appropriate relation between a head and a specifier. Following Chomsky (1995b) we assume that all these requirements are encoded as properties of lexical heads. To indicate that a head X has a certain complement Y, rather than "percolating" just some of the features of the head X, we assume that all the features of the head X "project over" those of Y. To depict this in a tree, we will use the notation



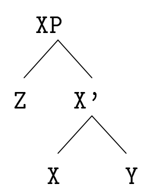
The more traditional X-bar structure here would be something like the following:



A head `X` and complement `Y` may combine with a specifier `Z` to yield a structure like the following:



The traditional X-bar structure here would be something like this:



With these new simpler structures, it is a trivial matter to find the head of any projection; at each internal node one simply goes down the "lesser" branch, the one "pointed to," until reaching the collection of features which is the head. A maximal constituent is either the root of a complete structure or else a node that does not project over its sister. We will assume that all branching is binary, and that the order of heads, complements and specifiers is uniformly the one just depicted: specifier, head, complement.

To implement a simple account of movement, LF movement, and reconstruction, the language will contain, in addition to simple constructions like (3i) for the determiner `some`, constructions in which this element has been split into (3ii) its phonetic features `/some/`, (3iii) its interpretable features (`some`), and (3iv) its bare categorial structure:

(3)

(i)	(ii)	(iii)	(iv)
`d:[=n]`	`d:[=n]`	`d:[=n]`	`d:[=n]`
|	|	|	
`some`	`/some/`	`(some)`	

(The feature `=n` will be explained just below.) Rather than lowering material in reconstruction after having raised it, we aim to achieve the same results by allowing movements to split a category into these various components along its chain. This makes a "one pass" computation of the syntactic structure possible.

To indicate that a head `x` has an unfilled selection requirement, an unfilled "receptor" for some category `y`, we will use the feature `=y`. The head `some` that selects a noun will have the feature `=n`. This feature will be licensed by "incorporating" the categorial feature `n` of the selected constituent by head movement, where this incorporation is possible only in a strictly local configuration. All head movements will be "covert;" that is, no phonetic material will be moved when these relations are established. When `n` is incorporated into a head with the feature category `=n` (together with any other requirements of the selected head), we will indicate that this receptor has been filled by removing the feature `=n` and deleting the category `n`.[2] If the moved verb has any interpretable and phonetic features, they will be left behind.

Turning to quantifier phrases, we will adopt a simplified version of the basic ideas of Beghelli and Stowell (1996) and Szabolcsi (1996).[3] The following four categories of determiners are distinguished:

(4) negative determiners (`no`),
distributive and universal determiners (`each, every`),
group denoting determiners (`the, some, a, one, three,` ...),
counting determiners (`few, fewer than 5, more than 6,` ...).

Furthermore, we assume the special functional categories `ref, dist` and `foc` (or `share`), which provide specifier positions that distinguish among these quantifiers. In simple clauses with transitive verbs, we order these categories with respect to the complementizer `c`, tense `t`, and the verb `v` as follows:

`c (ref) (dist) (foc) t v.`

Beghelli and Stowell (1996) cite work from Szabolcsi, Kinyalolo and others to support the claim that quantifiers of (roughly) these 4 types are restricted to

---

[2]The similar grammars explored in Stabler (1996) distinguish "weak" selection features `=x` from "strong" selection features `=X` which trigger overt head movement. For present purposes, we ignore overt head movement.

[3]The reader is referred to these sources for the motivations and details of the sort of syntactic theory of quantifiers considered here.

different surface positions in other languages; but in English the argument for the various specifier positions is less direct. In any case, this account will be approximated in our simple language fragment, and the syntactic properties of objects formed with negative and counting determiners will accordingly reflect their inability to take inverse scope. More precisely, in the fragment defined below, as in the Beghelli and Stowell (1996) account of human languages, we have the following:

(5) (i) negative quantifiers are interpreted in the specifier of `neg`
(ii) distributive and universal quantifiers are interpreted in the specifier of `dist`
(iii) group denoting quantifiers can interpreted in the specifiers of `ref`, `foc` or in their case positions
(iv) counting determiners are interpreted in their case positions

In the simple language defined here, we assume that the case positions in a simple clause are the specifier of `t` and the specifier of `v`. We will not treat Beghelli and Stowell's (1995) interesting suggestions about quantification over events, or the differences between `each`, `every` and `all`, or the full account of negation. There seems to be no obstacle to elaborating the fragment presented here to encompass these ideas.

2.1 Syntax

We begin as in Keenan and Stabler (1994), letting the syntax comprise four sets: a vocabulary V, a set of categories Cat, a lexicon Lex which is a set of expressions which are structures built up from V and Cat, and a set of structure-building rules $\mathcal{F}$ which map sequences of expressions to other expressions. For any such grammar $G = \langle V, Cat, Lex, \mathcal{F} \rangle$, the language $L(G)$ is the closure of Lex under $\mathcal{F}$. That is, the language is all the structures that can be built from the lexicon using the structure building operations that the grammar provides.

We define the **vocabulary** V as the union of three sets:

(6) a. $PI = \{$ `some, less, than, one, no, every, linguist, sentence, speaks, believes, is` $\}$
b. $P = \{/\mathtt{x}/|\ \mathtt{x} \in \mathtt{PI}\}$ (phonetic features)
c. $I = \{(\mathtt{x})|\ \mathtt{x} \in \mathtt{PI}\}$ (interpreted features)
d. $V = (PI \cup P \cup I)$

A lexical item may include some element of PI in a structure together with syntactic features, where we distinguish among these, the basic categories and the others:

(7) a. $base = \{\texttt{v,d,n,c,case,t,neg,ref,dist,foc}\}$. (basic categories)

b. $features = \{\texttt{=x} \mid \texttt{x} \in base\} \cup$ (selection requirement)
$\{\texttt{+x} \mid \texttt{x} \in base\} \cup$ (weak assignment of x to specifier)
$\{\texttt{+X} \mid \texttt{x} \in base\} \cup$ (strong assignment of x to specifier)
$\{\texttt{-x} \mid \texttt{x} \in base\}$ (requirement of x)

Finally, projections of any head are labeled with either $>$ or $<$ as indicated above.

The lexicon is a finite set of trees, lexical heads. We can think of each lexical head as a sequence of features, but for convenience we will separate the categorial feature, the phonetic and interpretable features, and the other features, depicting them in the following way:

```
          category:[other features]
                       |
  phonetic and interpretable features
```

If the category feature is absent, we will put a 0 in that position. If the phonetic or interpretable or both are absent, we may have just `(interpretable material)` or `/phonetic material/` or 0. If there are no other syntactic features, we have the empty list `[]` of other features.

We now present some elements of the **lexicon**. Here are four determiners and two nouns, but we allow the determiner `some` to be syntactically 3 ways ambiguous:

```
 d:[=n]      d:[=n,-foc]      d:[=n,-ref]      d:[=n]
   |              |                |              |
  some          some             some       more than one


d:[=n,-neg]     d:[=n,-dist]     n:[-case]      n:[-case]
     |               |               |              |
     no            every          student        sentence
```

We have transitive verbs like `speaks`. To introduce a simple recursion we allow `believes` to select a `c` object and a `d` subject. To facilitate the discussion of semantic properties in §2.3, we include `is` as a transitive verb, ignoring the special syntactic properties of this verb.

```
v:[+case,=d,=d]     v:[+case,=c,=d]     v:[+case,=d,=d]
       |                   |                   |
    speaks             believes                is
```

Finally, functional categories provide the glue that ties the lexical elements together. These elements have no phonetic features:

```
c:[=ref]      c:[=dist]      c:[=foc]      c:[=t]
   |              |              |            |
   0              0              0            0

ref:[=dist,+ref]     ref:[=foc,+ref]     ref:[=t,+ref]
       |                   |                   |
     (ref)               (ref)               (ref)

dist:[=foc,+dist]     dist:[=t,+dist]     foc:[=t,+foc]
       |                    |                  |
    (dist)               (dist)              (foc)

t:[=neg,+CASE]     t:[=v,+CASE]     neg:[=v,+neg]
       |                 |                |
      (t)               (t)             (neg)
```

As will become clear, these lexical entries do not provide for sentences with more than one universal or distributive quantifier. In a language where all universal quantifiers are covertly fronted, we could allow a lexical entries for `dist` to recursively select another `dist`. In a language where only one quantifier is fronted overtly, we could let `agr` select `dist` with a strong `+DIST` feature , while letting the recursive form select `dist` with a weak `+dist` feature. We leave this and other similar elaborations aside for the present.

Roughly following Chomsky (1995b), we have two basic structure building relations, *merge* and *move*, but we break *move* up into 3 different functions, according to whether there is movement of a complete category with its phonetic features and interpretable structure, a movement of just the interpretable structure (leaving phonetic properties behind), or a movement of just phonetic features. These operations are called *movepi, movei, and movep*, respectively. So our set of **structure building functions** is

$$\mathcal{F} = \{merge, movepi, movei, movep\}.$$

We now define each of these structure building operations in turn. Example applications will be presented in a derivation below.

(8) The function *merge* combines two expressions, in response to a selection feature `=x`. It is restricted to two cases: (i) a head with `=x` merges with

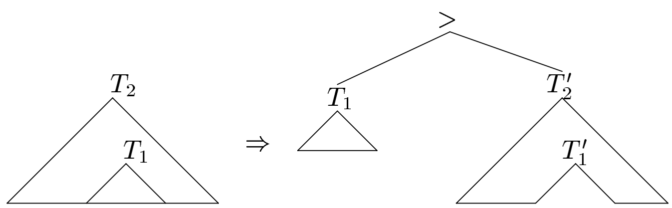


Figure 1 *movepi* raises subtree T_1, replacing it by empty subtree T_1'

a constituent of category `x` on its right (a complement), deleting both of these features; (ii) a nonhead with `=x` merges with a category `x` on its left (a specifier), deleting both features and checking any other `+y`, `-y` pairs. For the moment we will assume that in both cases the `x` head incorporates all the features of the other category, except when `v` incorporates `d` it leaves behind all features of the form `-y`.

(9) The function *movepi* is similar to what is usually called XP substitution. Following Chomsky (1995b) in conceiving of it now as a structure building operation, it applies to a single constituent whose head has a "strong "`+X` feature, moving the closest `-x` proper subconstituent to its specifier position as shown in Figure 1. We will not allow this operation to move a constituent out of a position where it must be interpreted. That is, to get the results in (5i)-(5iv), this function can move a `-case` subtree T_1 unless it is the specifier of `dist, foc, ref` or `neg`. So let's say that a subtree T_1 is **movable**, $\mu(T_1)$, if, and only if it is not the specifier of `dist, foc, ref` or `neg`, and it has at least one of the features `-case, -dist, -foc, -ref, -neg`. Furthermore, whenever *movepi* or *movei* move the interpretable part of a constituent that has no `-case` feature, moving it to check some other feature `-x`, the movement leaves behind a special interpreted feature $\blacktriangleleft_x$.

(10) The second type of movement, *movei*, is triggered by a "weak" feature `+x` rather than a strong feature. This is covert XP movement, similar to *movepi* except that the phonetic features of the moved constituent are left behind. Only the syntactic and interpreted features move.

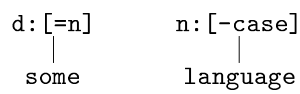
(11) What happens when movement is triggered by a strong feature `+X` but the only available `-x` constituent is not movable? As discussed just above, this happens for example in the case of a `-neg` subject which ends up in the specifier of `neg` with an unchecked `+CASE` feature. In this case, *movep* can apply, moving just the phonetic structure to the position of the strong feature where it must be pronounced.

This completes the specification of the entire grammar $G = \langle V, Cat, Lex, \mathcal{F} \rangle$. Formally, the language $L(G)$ is everything that can be derived from the lexical elements *Lex* using any of the five structure building functions in $\mathcal{F}$. Of course, we will typically be interested in derivations of well-formed clauses, that is, expressions with root `c` in which all structural requirements have been satisfied. Let , (G) be this set of all expressions with root `c` in which all structural requirements have been satisfied.

2.1.1 A simple clause

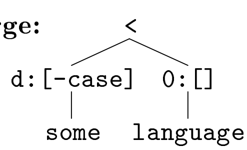
The grammar allows a number of derivations of well-formed clauses that would be pronounced as `some linguist speaks some language`. One of them interprets both arguments of the verb in their case positions. We step through the derivation of this simple structure first. We begin by drawing the components of the object from the lexicon:

step 0 lexicon:



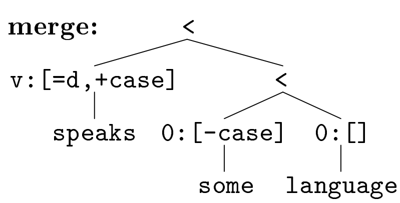
These components can be merged, deleting `=n` and the categorial feature `n`:

step 1 merge:



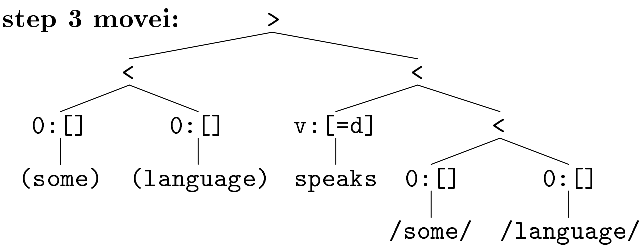
Then we merge the result with the lexical item `speaks`, deleting `=d` and the categorial feature `d`:

step 2 merge:



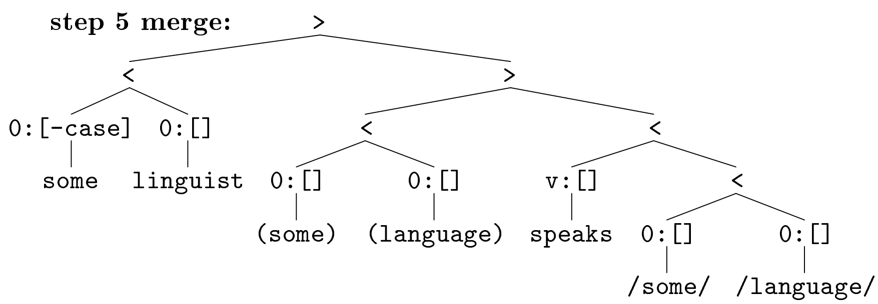
The result has `+case` and the derivation cannot proceed until this feature is discharged, so we covertly move the object to the specifier position, deleting `+case` and `-case`. This has the effect of splitting the interpretable and categorial features of the object from the phonetic features, the latter being left behind:

step 3 movei:



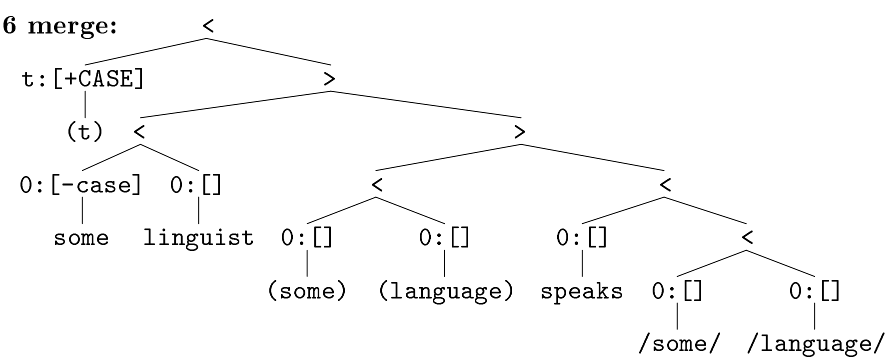
Step 4 of the derivation merges `some` with `linguist` to form the subject, using *merge* as was done in Step 1. The subject can then be merged with the result of Step 3 to give us a verbal projection that contains both the subject and the object:

step 5 merge:



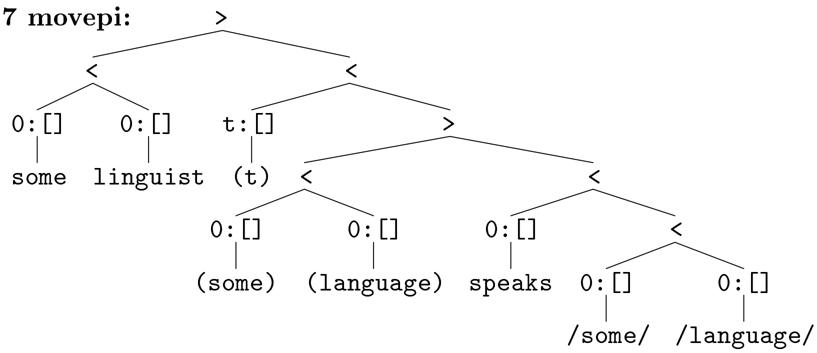
This structure can then be merged as the complement of the lexical `t` that selects it, deleting `=v` and `v`:

step 6 merge:



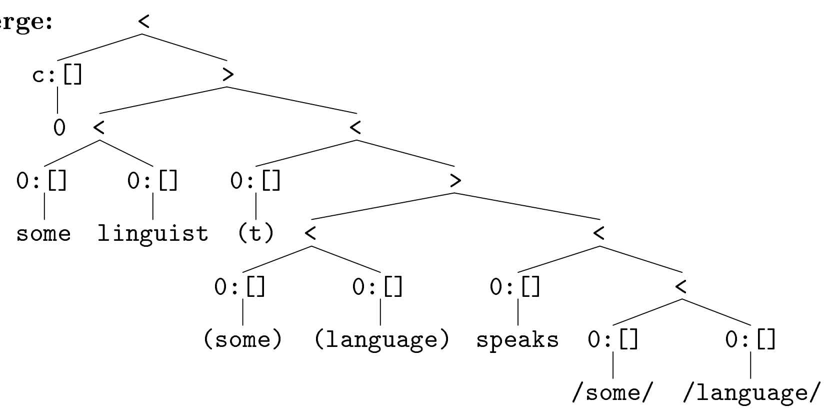
The head of this structure has a strong `+CASE` feature which must be assigned, so we overtly move the subject, deleting `+CASE` and `-case`:

step 7 movepi:

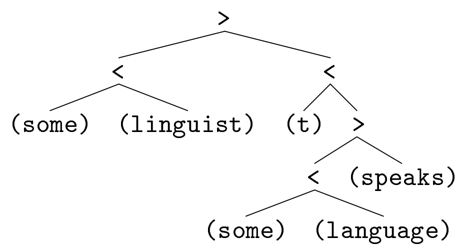


Finally, this whole structure can be taken as the complement of the lexical item `c` that selects it, deleting `=t` and `t`:

step 8 merge:



Notice that in this derived structure, there are no outstanding syntactic features except for the categorial feature `c`. Reading the phonetic material in order across the leaves of the tree we find `/some linguist speaks some language/`. Reading the interpretable constituents across the leaves of the tree we find: `(some linguist some language speaks)`. Actually, for PF and for LF, we require more structure than just the string. If we strip out just the parts of this last structure that are semantically relevant, we have the structure:



Leaving off the parentheses, the semantically relevant structure of the verb and its arguments is just

`[`$_{\mathtt{t}}$`[some linguist] [[some language] speaks]].`

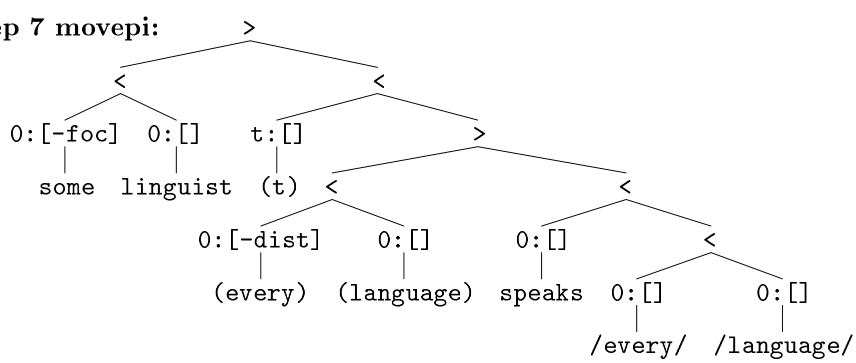
In the next section we consider a derivation in which this arrangement of semantic constituents is altered.

2.1.2 Inverse scope

The grammar allows a number of derivations of well-formed clauses that would be pronounced as `some linguist speaks every language`. In one of them, `every language` in `dist` scopes over `some linguist` in `foc`. This structure has the inverse scope reading (1is) discussed in the introduction. (We will discuss the interpretation of this structure in more detail in §2.3, below.)

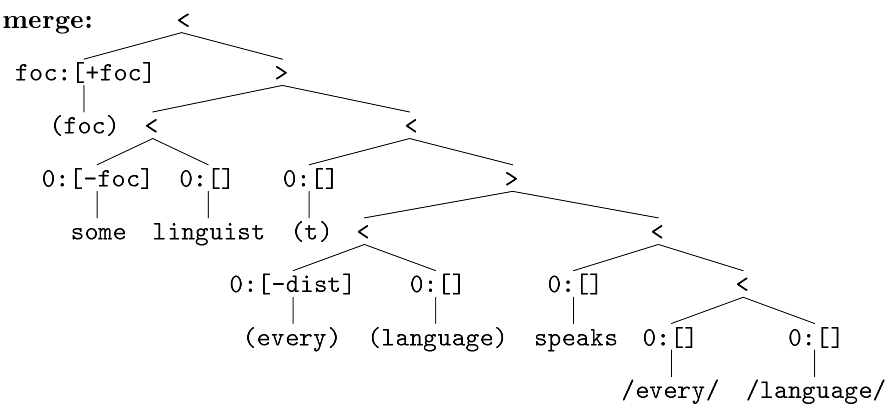
We begin the derivation with the same first seven steps as in the previous case, combining the `v` phrase with `t` and raising the subject to the specifier position. The resulting structure is just like the one we have at the same stage of the previous derivation except we have the additional features `-dist` and `-foc` which must be discharged.

step 7 movepi:

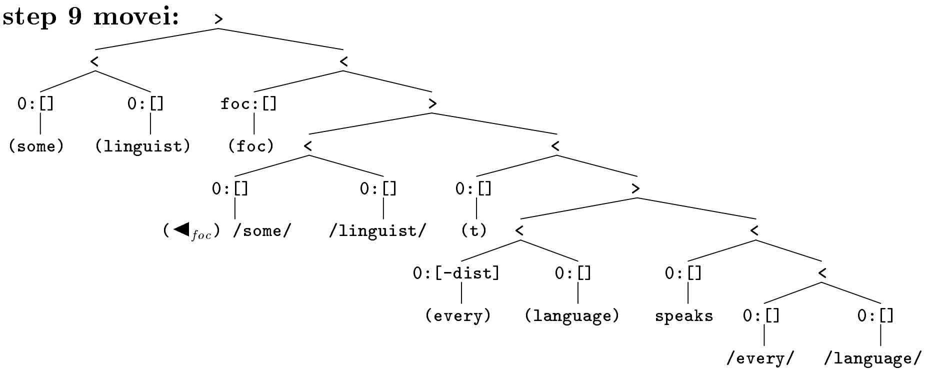


To discharge these additional features, a few more derivational steps are needed. First, we merge with `foc` in Step 8, and then raise the focused argument to specifier position in Step 9. Notice that Step 9 leaves behind the semantic feature $\blacktriangleleft_{foc}$, since the constituent being moved has already had its `-case` feature checked:

step 8 merge:

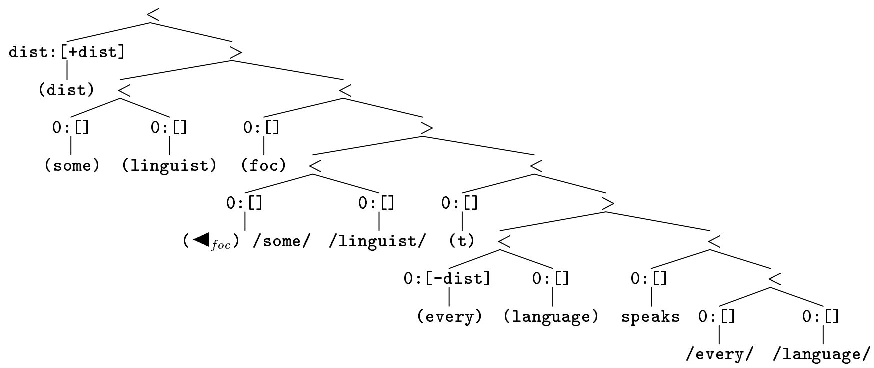


step 9 movei:

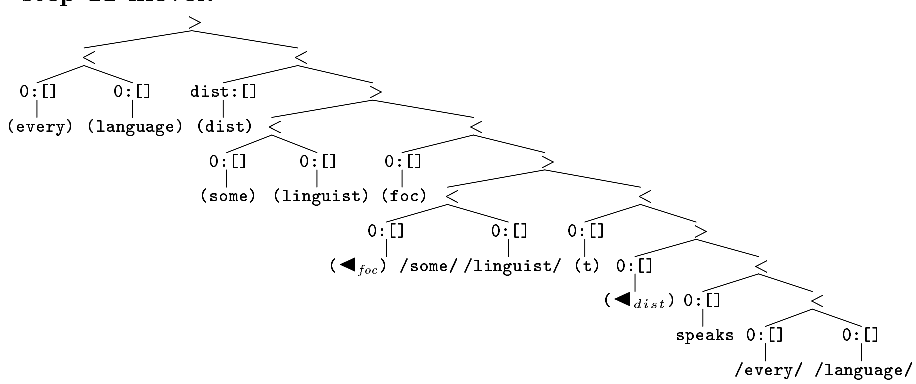


The structure that results at step 9 has only one syntactic feature left, namely the category `foc`. This feature can be deleted by merging it with the lexical item `dist` which selects it, in step 10. Then in step 11, the `+dist` feature of this new head pulls the interpreted features of the `-dist` object up to its specifier position to cancel `+dist`:

step 10 merge:

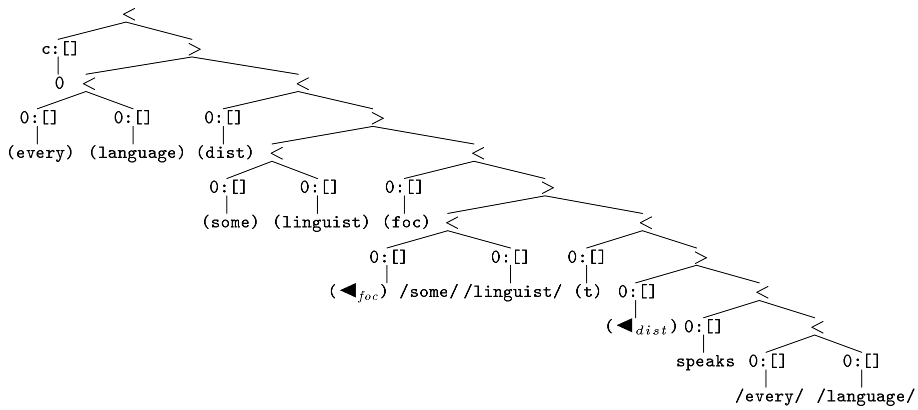


step 11 movei:



The phrase that results at step 11 again has just one syntactic feature, `dist`, which allows it to be the complement of a lexical `c` head in step 12, completing the derivation of a clause in which all syntactic features except `c` have been canceled:

step 12 merge:



The semantically relevant structure here, the LF, is

[$_{dist}$every language [$_{foc}$some linguist [$_{t}$◀$_{foc}$ ◀$_{dist}$ speaks]]].

2.2 Basic properties

By derivations of the sort presented in the previous sections we can establish that , (G) contains a range of structures of the sort we are interested in:

PROPOSITION 1

(a) [$_{\texttt{dist}}$(every language)[$_{\texttt{foc}}$(some linguist)[◀$_{foc}$◀$_{dist}$ speaks]]] ∈ , (G)

(b) [$_{\texttt{ref}}$(some linguist)[$_{\texttt{dist}}$(every language)[◀$_{ref}$◀$_{dist}$ speaks]]] ∈ , (G).

(c) [$_{\texttt{dist}}$(every language)[(some linguist) ◀$_{dist}$ speaks]] ∈ , (G).

Other, more general properties of , (G) can be established from the definition of the grammar:

PROPOSITION 2

(a) There is no derivation of a c∈ , (G) in which the interpretable structure of the object no language appears higher than the interpretable structure of the subject some linguist.

(b) There is no derivation of a c∈ , (G) in which the interpretable structure of the object less than one language appears higher than the interpretable structure of the subject some linguist.

Proof: (a) follows immediately from the movability requirement μ that restricts the domains of the functions *movepi* and *movei*. These are the only functions that move interpretable structure, and they are blocked from moving any specifier of `neg`, but `no language` will have a `-neg` feature which can only be checked in that position.
(b) follows again from the movability requirements. Since `less than one language` will have its case feature checked immediately in object position, and since it has no `-dist, -foc, -ref` or `-neg` feature, its interpretable structure cannot be moved at all. □

In sum, we have captured in this fragment the basic structural distinctions set out in (5).

Although chains, sequences of constituents coindexed by numbers or variables, are not present at any point in our derivations, we can still talk about series of movements which would have been regarded as constituting a chain in earlier theories. It is clear that the following properties fall out of our definition of the grammar:

PROPOSITION 3 Trace is immobile.

Proof: This follows immediately from the fact that the movements defined here leave no features on the trace which could trigger any later syntactic operation. □

PROPOSITION 4 Chains are uniform.

Proof: There are no chains in the syntax, and no indexing of categories with numbers or elements of a model. However, we can look at the possible sequences of movements and see that various desired sorts of "uniformity" are guaranteed. For example, no constituent can move to more than one case position; no head can adjoin to another head and then move to a specifier position; and so on. These follow from our definitions of the movement operations, so that no additional uniformity requirement need be stipulated. □

Finally, it is a simple matter to compute the syntactic structures generated by the grammar G, as we see stepping through the derivation of the example structure above.[4] These structures determine quantifier scope, so what we have seen is how the computation of quantifier scope can proceed without reference to semantic values of any expressions. This leads us back to the original puzzle: haven't we missed something important? Before tackling this question, let's sketch a semantics for the fragment.

[4]It is not quite so simple to provide a reasonable account of how people might compute these structures incrementally, as they hear a sentence from beginning to end. This problem is explored in Stabler (1996).

2.3 Semantics

We will interpret the simple clauses in , (G) extensionally, providing no account of sentences involving the verb `believe`. For simplicity we will also ignore the contribution of tense etc. However, it will be relevant whether an argument appears in the a projection of a head with a tense feature `(t)`, the tense phrase, or not, as we will see. The language , (G) contains predicates which take various numbers of arguments. We use the notation q^j to indicate that q is a predicate that takes j arguments. As in standard extensional semantics, a predicate q^j is denotes a j-ary relation Q^j on the (nonempty) domain A. And as usual, we let nouns denote 1-ary relations.

We adapt to our purposes a formulation of semantic theory from Ben-Shalom (1996).[5] The basic elements of our models are infinite sequences s of objects from some nonempty domain, together with three special registers which will hold focused, referentially identified, or distributed elements. Since the contents of these registers are always in the sequences they are associated with, we can depict them as pointers to particular elements in the sequence, as for example in

$$\begin{array}{l} \quad\quad\;\; \mathit{ref} \quad\; \mathit{dist} \quad\; \mathit{foc} \\ \quad\quad\;\;\; \downarrow \quad\;\;\; \downarrow \quad\;\;\; \downarrow \\ s = (\, s_0, s_1,\, s_2,\, s_3, s_4, s_5 \ldots) \end{array}$$

Here, the element in the first position is also the content of the *ref* register, and this element will be denoted $\mathit{ref}(s)$. The contents of the second and fourth registers are similarly $\mathit{dist}(s)$ and $\mathit{foc}(s)$, respectively. We call a sequence together with these registers an r-sequence.

The result of extending r-sequence s by adding some $a \in A$ to the first position is denoted by as as usual. The result of extending s by adding some $a \in A$ to the first position and also marking the new element as the contents of the *ref* register will be denoted

$$\begin{array}{l} ref \\ \downarrow \\ a \;\; s \end{array}$$

Similarly for extensions with elements that are marked *dist* and *foc*.

[5]Ben-Shalom (1996) notices that the standard model theory of propositional modal logic (as in, e.g. Goldblatt 1992) can be adapted to a language of predicates and generalized quantifiers by letting the points of the model be sequences and letting the generalized quantifiers correspond to modal operators. Compare Alechina (1995). The resulting semantic theory is remarkably simple and natural. For brevity, in defining the models below we mention just one accessibility relation R, namely extension, but the definition of $\models_s$ shows that we actually use different extensions according to the restrictors of the quantifiers. See Ben-Shalom (1996) for a more careful treatment.

With these conventions, we can let a model be a triple $\mathcal{M} = (S, R, V)$ where S is the set of infinite r-sequences of objects from domain A; R is a binary relation on S such that

$$sRs' \text{ iff } s' = as \text{ for some } a \in A;$$

and V is a function from predicates q to subsets of S defined as follows:

$$s \in V(q) \text{ iff } \begin{cases} s_0 = s_1 & \text{when } q = \texttt{is}^2, \\ (s_0, \ldots, s_{j-1}) \in \mathrm{Q}^j & \text{for any other } q = q^j \end{cases}$$

Then an r-sequence $s \in S$ in a model $\mathcal{M} = (S, R, V)$ satisfies an expression as follows:

$\mathcal{M} \models_s q$	iff	$s \in V(q)$
$\mathcal{M} \models_s [_f\texttt{every}\ q\ \phi]$	iff	for every $a \in \mathrm{Q}$, $\mathcal{M} \models_{\overset{f}{\underset{as}{\downarrow}}} \phi$
$\mathcal{M} \models_s [_f\texttt{some}\ q\ \phi]$	iff	for some $a \in \mathrm{Q}$, $\mathcal{M} \models_{\overset{f}{\underset{as}{\downarrow}}} \phi$
$\mathcal{M} \models_s [_f\texttt{no}\ q\ \phi]$	iff	for no $a \in \mathrm{Q}$, $\mathcal{M} \models_{\overset{f}{\underset{as}{\downarrow}}} \phi$
$\mathcal{M} \models_s [_f\texttt{less than one}\ q\ \phi]$	iff	for less than one $a \in \mathrm{Q}$, $\mathcal{M} \models_{\overset{f}{\underset{as}{\downarrow}}} \phi$
$\mathcal{M} \models_s [\blacktriangleleft_f\ \phi]$	iff	for $a = f(s)$, $\mathcal{M} \models_{as} \phi$

As usual, we say that an expression ϕ is verified by a model, $\mathcal{M} \models \phi$ if and only if for all $s \in S$, $\mathcal{M} \models_s \phi$. And an expression ϕ is valid in frame (S, R) just in case for all valuations V, $(S, R, V) \models \phi$.

Consider for example the expression

$$\phi = [_{dist}\texttt{every language}[_{foc}\texttt{some linguist}[_t \blacktriangleleft_{foc} \blacktriangleleft_{dist}\texttt{speaks}]]].$$

We stepped through the syntactic derivation of this expression in §2.1.2. Let's show that this sentence ϕ is true in a model $\mathcal{M}$ where there are two linguists $l0, l1$ and two languages $a0, a1$ where $l0$ speaks $a0$ and $l1$ speaks $a1$. That is, the universe A is $\{l0, l1, a0, a1\}$, so S is the set of infinite r-sequences of these elements. We have the relations

$\text{LINGUIST} = \{l0, l1\}$
$\text{LANGUAGE} = \{a0, a1\}$
$\text{SPEAKS} = \{(a0, l0), (a1, l1)\}$

We adopt the convention of listing the elements of tuples in these relations in order of decreasing obliqueness, putting the subject last. Using this order for the arguments slightly simplifies the definition of the satisfaction relation, as will become clear.

Now consider any particular $s \in S$. We will show that for any such s, $\mathcal{M} \models_s \phi$. If s is some r-sequence

$$s = (\overset{\textit{ref}}{\overset{\downarrow}{a_0}},\ \overset{\textit{dist}}{\overset{\downarrow}{l_1}},\ \overset{\textit{foc}}{\overset{\downarrow}{l_1}}, \ldots)$$

The first three elements of this r-sequence are $ref(s) = a0$, $dist(s) = l1$ and $foc(s) = l1$. To show that our example is verified by this and every other r-sequence, the contents of the registers and the first positions in the sequence are irrelevant. Any other r-sequence will verify the sentence ϕ too.

To establish $\mathcal{M} \models_s \phi$, we simply step through the truth definition given above. First we see that

$$\mathcal{M} \models_s \phi \text{ iff } \quad \text{for every } a \in \textsc{language}, \quad \mathcal{M} \models_{\overset{dist}{\underset{a}{\downarrow}}\,s} [_{foc}\texttt{some linguist}[_t \blacktriangleleft_{foc} \blacktriangleleft_{dist}\texttt{speaks}]].$$

Notice that this step involves considering extending of the original r-sequence s to r-sequences as in which the $dist$ register has also been reset to a. Using the truth definition again, we see that

$$\mathcal{M} \models_s \phi \text{ iff for every } a \in \textsc{language}, \text{ and for some } l \in \textsc{linguist}, \quad \mathcal{M} \models_{\overset{foc}{\underset{l}{\downarrow}}\,\overset{dist}{\underset{a}{\downarrow}}\,s} [_t \blacktriangleleft_{foc} \blacktriangleleft_{dist}\texttt{speaks}].$$

The next two steps involve extending the r-sequences of interest without resetting any registers. Rather, they extend the r-sequence by copying the contents of the registers. First we extend with the contents of the *foc* register, then with the contents of the *dist* register:

$$\mathcal{M} \models_s \phi \text{ iff for every } a \in \textsc{language}, \text{ and for some } l \in \textsc{linguist}, \quad \mathcal{M} \models_{l\,\overset{foc}{\underset{l}{\downarrow}}\,\overset{dist}{\underset{a}{\downarrow}}\,s} [\blacktriangleleft_{dist}\texttt{speaks}].$$

$$\mathcal{M} \models_s \phi \text{ iff for every } a \in \textsc{language}, \text{ and for some } l \in \textsc{linguist}, \quad \mathcal{M} \models_{a\,l\,\overset{foc}{\underset{l}{\downarrow}}\,\overset{dist}{\underset{a}{\downarrow}}\,s} \texttt{speaks}.$$

Now we can see that the satisfaction relation does hold, since for every $a \in$ LANGUAGE there is some $l \in$ LINGUIST such that $allas \in V(\texttt{speaks})$. This is the case because for every $a \in$ LANGUAGE there is some $l \in$ LINGUIST, $(a, l) \in$ SPEAKS.

In this simple semantic theory, we see that the scope of a quantifier is determined by the syntactic position of its interpretable structure. We have constituents $\blacktriangleleft_{dist}$, $\blacktriangleleft_{foc}$, and $\blacktriangleleft_{ref}$, which are "bound" in some sense, but they are unlike variables in certain other respects. There are not infinitely many of them. Notice also that a clause in which no arguments have moved is perfectly interpretable. That is, there is no need to move just in order to produce a variable that can be bound by a quantifier. Furthermore, the positions of quantifiers is severely restricted by the syntax. Since we only have two decreasing quantifiers in the fragment, `no` and `less than one`, and since we know from

Proposition 2 that when these quantifiers occur in object position their interpretable structures never move above the interpretable structure of the subject, it follows that decreasing objects will never have an inverse scope reading.

The puzzle about this is that the computation of syntactic structure, sketched in the previous section, determines quantifier scope and yet it makes no reference to the semantic values of the determiners. This should be puzzling, as we can see from the fact that, as far as the syntax is concerned, there is no reason not to expect a decreasing quantifier to have the feature `+dist` or `+ref`. The fact that there is no such quantifier in human languages would then be just an accident, but this is implausible. To avoid this impasse, the theory needs one additional ingredient.

2.4 Inference

The decreasing quantifiers can be characterized by their inferential behavior. Stating our inference rules over the structures in , (G), and showing just the interpretable structures, a quantifier `D` is decreasing if the following inference is valid:

$$\frac{\begin{array}{c}\texttt{[D N1 V]}\\ \texttt{[}_{dist}\texttt{every N2 [}\blacktriangleleft_{dist}\texttt{ some N1 is]]}\end{array}}{\texttt{[D N2 V]}}$$

This observation suggests a solution to the puzzle with which we began. As noted in the introduction, we can assume that the very features of a determiner which determine scoping options also constrain the inferential role of the structures containing the determiner. Let's sketch how this might work.

Beginning with the easiest case, we can assume that we have the following inference rule for <u>any</u> determiner `D`,

$$\frac{\begin{array}{c}\texttt{[}_{neg}\texttt{ D N1 V]}\\ \texttt{[}_{dist}\texttt{every N2 [}\blacktriangleleft_{dist}\texttt{ some N1 is]]}\end{array}}{\texttt{[}_{neg}\texttt{D N2 V]}}$$

That is, any determiner which appears in the specifier position of `neg` will be one that licenses this inference. Any such determiner is decreasing. Once this rule is given, it is no longer an accident that negative quantifiers scope the way they do. A link between decreasingness and scope is established, in purely formal terms.

Notice that we do not say, if a determiner is decreasing then it may occur in the specifier of `neg`. That is not quite true, in the first place, since counting determiners may also be decreasing even though they cannot occur in the specifier of `neg`. But more importantly, this claim reverses the order of explanation. On the present account, we do not assume that the speaker has some "grasp" of the semantic value of the determiner first, and then decides where to put it in the syntactic structure. Rather, the speaker uses the determiner in a certain way, in the syntax according to the requirements specified in its features, and in inference. Constraints on its semantic value then follow.

The situation here is artificially simple, but we have secured the basic point, that the very features which determine syntactic properties can also determine inferential and semantic properties in theories like this one. Notice for example that although the only counting determiner in our fragment is decreasing, we could have had counting determiners like `more than one`, which are not decreasing. So in this case we will not be able to say that every counting determiner will license the characteristic inference of decreasing quantifiers. Rather, the particular kind of quantifier that appears in the counting quantifier positions will matter. This is just a case where formal distinctions must be drawn within the categories of items that can occupy the syntactic positions.

3 SOME ALTERNATIVES AND ELABORATIONS

3.1 Alternative explanations of scoping restrictions

There are many alternative approaches to the computation of quantifier scope. Most are easily assigned to one of the following categories:

Syntactic restructuring with variables: The approach offered here falls in this category, together with previous approaches to "quantifier raising" (QR). Other examples of this sort of approach are provided by Higginbotham and May (1981), Fiengo and May (1995), and others.

Syntactic restructuring without variables: Keenan (1987) shows how, if we allow the subject and verb to form an interpretable constituent which is, in turn, interpreted with the object, we can compute inverse scope without the use of bound variables.

Semantic restructuring with variables: It is also possible to compute the normal and inverse scope readings from a single syntactic structure by, in effect, storing and raising the embedded object quantifier in the computation of logical form. Examples of this sort of approach are provided by Cooper (1983), Pereira (1990), Dalrymple et al. (1994) and others.

Semantic restructuring without variables: It is possible to avoid variables on this approach too. Instead of raising the object quantifiers to bind a variable, the object quantifiers can be interpreted as combining with the verb to yield a functions from quantifiers to propositions, as in Nam (1991) and others.

The present approach differs from previous QR-based approaches in various respects: the classes of quantifiers distinguished by Beghelli and Stowell (1996) are distinctive, as are the assumptions about the structural positions available for these constituents. The representation of previous positions of arguments is also distinctive both syntactically (there are finitely many types of symbols $\blacktriangleleft_x$, not infinitely many variables), and semantically (the symbols $\blacktriangleleft_x$ are interpreted as propositions in a propositional calculus, not as terms in a predicate calculus). In a certain sense, there are no "variables," though the elements $\blacktriangleleft_x$ play a very similar role.

The basic properties of the present approach which do the work in solving the puzzle though are just these: the quantifiers fall into various syntactic categories with distinctive properties, and this variety also plays a role in determining inferential role and semantic value. So the semantic restructuring accounts face two difficulties. In the first place, there are empirical considerations of the sort adduced by Beghelli and Stowell (1996), Szabolcsi (1996) and other work in the same tradition. If it is true that the different classes of quantifiers are syntactically distinguished, with their surface positions related to different positions in linguistic structure by restricted movement relations of just the sort found elsewhere in syntax, then a syntactic approach to scoping phenomena is clearly preferable. Semantic approaches face a second kind of problem when we try to elaborate them to solve the puzzle of §1.

Let's briefly consider a recent proposal from Dalrymple et al. (1994) as a representative example.[6] This work assumes that a single syntactic structure (an "f-structure") like (12) is mapped into either of two alternative logical forms (13) or (14):

(12)

$$f{:}\begin{bmatrix} \textsc{pred} & \text{`speak'} \\ \textsc{tense} & \textsc{pres} \\ \textsc{subj} & g{:}\begin{bmatrix} \textsc{spec} & \text{`every'} \\ \textsc{pred} & \text{`linguist'} \end{bmatrix} \\ \textsc{obj} & h{:}\begin{bmatrix} \textsc{spec} & \text{`some'} \\ \textsc{pred} & \text{`language'} \end{bmatrix} \end{bmatrix}$$

(13) $f_\sigma \rightsquigarrow_t \mathit{every}(w, \mathit{linguist}(w), \mathit{some}(z, \mathit{language}(z), \mathit{speak}(w, z)))$

(14) $f_\sigma \rightsquigarrow_t \mathit{some}(z, \mathit{language}(z), \mathit{every}(w, \mathit{linguist}(w), \mathit{speak}(w, z)))$

The deduction of either logical form is made possible by providing lexical entries for such predicates that allow its arguments to be bound in either order. The

[6]The features of the Dalrymple et al. (1994) proposal that I discuss here are very basic, shared by many other approaches. The way the linear logic eliminates the need for "quantifier storage," the way certain logical forms are properly made unavailable – these interesting and distinctive features of the Dalrymple et al. (1994) proposal are not considered here.

predicate `speak` is associated with the higher order linear logic form (15) in a deductive system in which this form is mutually derivable with (16):

(15) $\forall X.\ g_\sigma \rightsquigarrow X \multimap (\forall Y.\ h_\sigma \rightsquigarrow Y \multimap f_\sigma \rightsquigarrow \mathit{speak}(X, Y))$

(16) $\forall Y.\ h_\sigma \rightsquigarrow Y \multimap (\forall X.\ g_\sigma \rightsquigarrow X \multimap f_\sigma \rightsquigarrow \mathit{speak}(X, Y))$

Without going into the details, by binding the inner formula first, using the first of these forms leads to normal scope, while using the latter leads to inverse scope. Notice that the linear logic is used in the deduction of the logical forms, but the language of the logical forms themselves, shown in (13) and (14), does not contain linear connectives like $\multimap$ or the higher order variables X, Y.

This is a "semantic restructuring" account because the mechanisms used in the deduction of the alternative scopes – a linear logic in this case – are not the ones used in the deduction of the syntactic structure. How could such an account be elaborated to disallow the inverse scope reading when the object is a decreasing quantifier? Since one of the main points of the Dalrymple et al. (1994) approach is apparently to make either argument of the predicate equally available, it appears that a significant change will be required. Any account is going to have to reconstruct some sort of asymmetry between the two arguments of the verb, and then have some corresponding difference between the lexical entries for the various sorts of quantifiers so that derivations of the unwanted scopes are unavailable. Supposing that this could be done, there is another problem, which is to indicate a plausible link between the restrictions on the scoping options and the inferential roles of the quantifiers. This is likely to be difficult too, because in accounts like this one, the mechanisms which govern the calculation of the logical form do not appear in the logical form itself. They are eliminated. As noted above, the language used in the deduction of the logical forms is distinguished from the language of the logical forms themselves. Whatever restrictions there are on quantifier scoping are presumably realized in the former, whereas inference patterns are presumably defined over the latter. In the fragment discussed above, on the other hand, there is only one language, , (G). It is syntactic form, the interpreted form, and the language of reasoning, all at once. In this language, it is easy to tie the grammatical features responsible for quantifier scoping to the features that license certain inference patterns.

3.2 Alternative explanations of the semantic generalizations

Barwise and Cooper (1981) note that "processing" a quantified statement `D N Pred` need not involve considering the generalized quantifier `D N`, which in a simple extensional treatment is a function from properties to truth values (or a set of properties). Such a function (or set) can be quite large, but rather than considering the whole function, it suffices, when the quantifier is monotone, to select a witness and check whether this witness stands in an appropriate relation to the property denoted by `Pred`. When the function is increasing, we check to see that every member of the witness is a member of the denotation of the predicate. Similar ideas are proposed by Ben-Shalom (1993) and Szabolcsi (1996).

These are clearly not accounts of what people do when they use the language, at least not in any direct sense. A witness or the generator of a principal filter can be very large too, infinite and undecidable, without any corresponding difficulty in the language users' use of the phrase in inference. Humans do not, of course, actually "check" each member of these sets for membership in another set, performing some computational step for each element of the set the way a semantic automaton might. Rather, we must draw our conclusions by deductive steps defined over representations of more abstract relations among whole sets, representations that are in at least this respect like our syntactic structures with their interpretable elements. Then, no surprise, the size of the sets being reasoned about does not generally have any bearing on the complexity of the reasoning.

So if the witness and generator based verification procedures are metaphorical, involving verification steps that are never really performed by the human language speaker, can we explain why these have seemed relevant in semantic theories for human languages? Yes. Inference methods can be regarded as justified by certain relations among the verification procedures associated with the sentences involved. For example, the direct verification, by checking all elements of the witness sets, of `every human speaks a language`, includes as a proper part all of the steps that would be required in the direct verification of `every linguist speaks a language`. That is why the inference of the latter from the former is justified. With such an intimate connection between these notions, it is not in the least surprising that insights about verification conditions are relevant to inferential roles, which are in turn tied to formal, syntactic properties of linguistic expressions in the computational model of the speaker-hearer.

4 CONCLUSIONS

There is a philosophical tradition which rejects the idea that pure thought is coded into language and then expressed, as if we could think the very thoughts we do without knowing anything of any human language, pure semantics. This view is rejected by Dummett (1993), for example,

> A view that might claim to represent common sense is that the primary function of language is to be used as an instrument of communication, and that, when so used, it operates as a code for thought. On this view, it is only because we happen to lack a faculty for directly transmitting thoughts from mind to mind that we are compelled to encode them in sounds or marks on paper ...
>
> The idea of a language as a code became untenable because a concept's coming to mind was not, by itself, an intelligible description of a mental event: thought *requires* a vehicle.

The conception of language and reasoning offered in this paper, according to which at least much of what we call reasoning is reasoning with the structures of our native language, fits with this rejection of the code conception of thought. We do not need to assume that there are two steps in reasoning: the formulation of pure thoughts, and then the bringing together of words to express them. A certain kind of bringing together of words, spoken or not, is thinking.

This view can also be contrasted with views according to which we have many languages: at least, the one we speak and the one we reason with. The objections to this sort of account, which we find in certain "semantic" accounts of quantifier scoping, have a different character. Here, it is argued that such views cannot provide the simplest explanations of semantic generalizations about syntax.

It is sometimes held that the meaning of an expression is uniquely determined by its inferential role. No such thing has been assumed here. But the connection between inferential role and meaning is especially clear in the case of quantifiers and logical constants. From this perspective, it is no surprise to find that the same formal structure that determines the scoping options of a quantifier – whatever, exactly, that structure is – can determine the inferential role of the quantifier. Consequently, semantic generalizations about the scoping options of quantifiers are not surprising, and we can make sense of how such generalizations can be true in a computational model of the language user.

References

Alechina, Natasha. 1995. *Modal Quantifiers.* PhD thesis, University of Amsterdam.

Barwise, Jon, and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4:159–219.

Beghelli, Filippo, and Tim Stowell. 1996. Distributivity and negation. In Anna Szabolcsi (Ed.), *Ways of Scope Taking.* Boston: Kluwer.

Ben-Shalom, Dorit. 1993. Object wide scope and semantic trees. In *Procs. of the Workshop on Semantics and Linguistic Theory, SALT III.*

Ben-Shalom, Dorit. 1996. *Semantic Trees.* PhD thesis, UCLA. Available at http://128.97.8.34/.

Brody, Michael. 1995. *Lexico-Logical Form: A Radically Minimalist Theory.* Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1995a. Language and nature. *Mind* 104:1–61.

Chomsky, Noam. 1995b. *The Minimalist Program.* Cambridge, Massachusetts: MIT Press.

Cooper, Robin. 1983. *Quantification and Syntactic Theory.* Dordrecht: Reidel.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat. 1994. A deductive account of quantification in LFG. In M. Kanazawa, C.J. Piñón, and H. de Swart (Eds.), *Quantifiers, Deduction and Context.* Stanford, California: Center for the Study of Language and Information.

Dummett, Michael. 1993. *The Seas of Language.* Oxford: Clarendon Press.

Fiengo, Robert, and Robert May. 1995. *Indices and Identity.* Cambridge, Massachusetts: MIT Press.

Goldblatt, Robert. 1992. *Logics of time and computation.* Stanford: CSLI/Stanford University.

Higginbotham, James, and Robert May. 1981. Questions, quantifiers and crossing. *The Linguistic Review* 1:41–79.

Keenan, Edward L. 1987. Unreducible n-ary quantifiers in natural language. In P. Gardenfors (Ed.), *Generalized Quantifiers.* Dordrecht: Reidel.

Keenan, Edward L., and Edward P. Stabler. 1994. Abstract syntax. In *Configurations*, Université du Québec à Montreal. Forthcoming. Available at http://128.97.8.34/.

Koopman, Hilda. 1994. Licensing heads. In David Lightfoot and Norbert Hornstein (Eds.), *Verb Movement*, 261–296. New York: Cambridge University Press.

Liu, Feng-hsi. 1990. *Scope Dependency in English and Chinese.* PhD thesis, Department of Linguistics, UCLA.

Nam, Seungho. 1991. Scope interpretation in nonconstituent coordination. In *Proceedings of the West Coast Conference on Formal Linguistics, WCCFL 10*, 337–348.

Pereira, Fernando C.N. 1990. Categorial semantics and scoping. *Computational Linguistics* 16:1–10.

Sportiche, Dominique. 1995. Sketch of a reductionist approach to syntactic variation and dependencies. In Hector Campos and Paula Kempchinsky (Eds.), *Evolution and Revolution in Linguistic Theory*. Washington: Georgetown University Press.

Stabler, Edward P. 1996. *Acquiring and parsing languages with movement*. Oxford: Basil Blackwell. Forthcoming. Draft available at http://128.97.8.34/.

Szabolcsi, Anna. 1996. Strategies for scope-taking. In Anna Szabolcsi (Ed.), *Ways of Scope Taking*. Boston: Kluwer.