7 Quantification and Grammar

7.1 The problem of quantifiers in object position

Almost all instances of quantifying DPs that we have looked at so far were in subject position. And there was a good reason for it. Compare (1a) to (1b):

(1) (a) $[_{DP}$ every linguist$]$ $[_{VP}$ offended John$]$.
    (b) John $[_{VP}$ offended $[_{DP}$ every linguist$]]$.

(1a) is true just in case the set of linguists is included in the set of those who offended John. We have seen how to arrive at the correct truth-conditions in a compositional way. The determiner denotation relates two sets. The first set (the restrictor set) is provided by the common noun "linguist", the second by the VP "offended John". But what if "every linguist" occurs in object position as in (1b)? Shouldn't we assume that "every" still denotes a relation between sets? But then, which two sets? The restrictor set is the set of linguists, and it is provided by the common noun as before. The second set should be the set of all those who were offended by John. But this set is not denoted by any constituent in (1b). This is, in a nutshell, the problem of quantifiers in object position.

The dilemma becomes more dramatic if we consider sentences with multiple quantifier phrases:

(2) Some publisher offended every linguist.

(2) has two readings. On one reading, the claim is that there is at least one publisher who offended every linguist. The other reading is compatible with a situation where every linguist was offended by a possibly different publisher. Set theory lets us express the two readings:

(2′) (a) $\{x : x \text{ is a publisher}\} \cap \{x : \{y : y \text{ is a linguist}\} \subseteq \{z : x \text{ offended } z\}\} \neq \emptyset$
     (b) $\{x : x \text{ is a linguist}\} \subseteq \{x : \{y : y \text{ is a publisher}\} \cap \{z : z \text{ offended } x\} \neq \emptyset\}$.

But how can we compute such statements in a compositional way from plausible syntactic structures?

From the perspective of our type-theoretic framework, the problem of quantifiers in object position presents itself as a type mismatch. What happens if we try to interpret (1b), for example? Recall that ⟦**every**⟧ is of type <<e,t>,<<e,t>,t>>, and ⟦**linguist**⟧ of type <e,t>. These combine by Functional Application (FA) to yield a denotation of type <<e,t>,t> for the quantifier phrase. ⟦**offend**⟧ is of type <e,<e,t>>. Unfortunately, denotations of this type do not combine with those of <<e,t>,t>, either as function and argument or as argument and function. So FA yields no value for the VP, nor does any other principle apply. We are stuck.

We mentioned in the last chapter that the relational theory of quantification that we have adopted here is the oldest known theory of quantification, dating back at least to Aristotle. The problem of quantifiers in object position is almost as old. Medieval scholars tried to solve it, but failed, and so did many logicians and mathematicians in more modern times. A solution was eventually found by Frege. Frege discovered the notation of quantifiers and variables, and thereby "resolved, for the first time in the whole history of logic, the problem which had foiled the most penetrating minds that had given their attention to the subject."[1]

Modern linguistic theories fall into different camps, depending on their approach to the problem of quantifiers in object position.[2] There are those who assume in the spirit of Frege that sentences are constructed in stages, and that at some stage, the argument positions of predicates might be occupied by traces or pronouns that are related to quantifier phrases via a syntactic relationship. The relationship is movement (Quantifier Lowering or Quantifier Raising) in Generative Semantics and in Chomsky's Extended Standard Theory and its offspring, and the operation of "Quantifying in" in Montague Grammar. Other semanticists avoid displacement of quantifier phrases, and try to interpret all arguments of predicates *in situ*. The displacement of quantifier phrases may be simulated in the semantics by storing their denotation in a so-called Cooper Store,[3] or the flexibility of type theory may be used to overcome type mismatches. Variable-free versions of predicate logic (Combinatory Logic[4]) led to variable-free versions of natural language semantics, as in the work of Szabolcsi, Steedman, Cresswell, and Jacobson.[5] In what follows, we have picked an example of an *in situ* approach and a particular instantiation of a movement approach for further discussion and comparison.

7.2 Repairing the type mismatch *in situ*

We will now consider a way of overcoming the problem of quantifier phrases in object position by leaving the quantifier phrase in place.

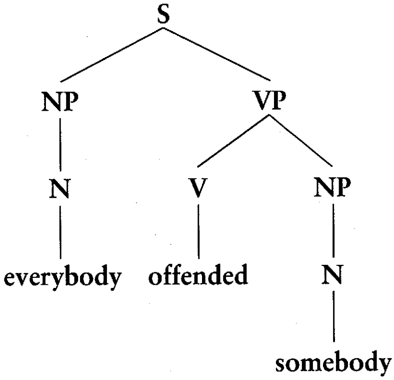
7.2.1 *An example of a "flexible types" approach*

On the "flexible types" approach,[6] we try to solve the problem of quantifier phrases in object position by optionally changing the semantic type of the quantifier phrase or of the verb. We will illustrate the first possibility here. You are invited to try out the second on your own.[7]

Let quantifier phrases be multiply ambiguous. For 1-place and 2-place predicates as arguments we have the following two entries[8] (extending this approach to any n-place predicate is straightforward – see exercise below):

$[\![\textbf{everybody}_1]\!] = \lambda f \in D_{<e,t>} \,.\, \text{for all persons } x \in D,\ f(x) = 1.$
$[\![\textbf{everybody}_2]\!] = \lambda f \in D_{<e,<e,t>>} \,.\, [\lambda x \in D \,.\, \text{for all persons } y \in D,\ f(y)(x) = 1].$
$[\![\textbf{somebody}_1]\!] = \lambda f \in D_{<e,t>} \,.\, \text{there is some person } x \in D \text{ such that } f(x) = 1.$
$[\![\textbf{somebody}_2]\!] = \lambda f \in D_{<e,<e,t>>} \,.\, [\lambda x \in D \,.\, \text{there is some person } x \in D \text{ such that } f(y)(x) = 1].$

For an example, consider the phrase structure tree (1):

(1)


The truth-conditions of (1) can be calculated as follows:

$[\![[_S\ \textbf{everybody}_1\ [_{VP}\ \textbf{offended somebody}_2]]]\!] = 1$
iff
$[\![\textbf{everybody}_1]\!]([\![[_{VP}\ \textbf{offended somebody}_2]]\!]) = 1$
iff
for all persons x, $[\![[_{VP}\ \textbf{offended somebody}_2]]\!](x) = 1$
iff

for all persons x, $[\![$**somebody$_2$**$]\!]([\![$**offended**$]\!])(x) = 1$
  iff
for all persons x, there is some person y, such that $[\![$**offended**$]\!](y)(x) = 1$
  iff
for all persons x, there is some person y, such that x offended y.

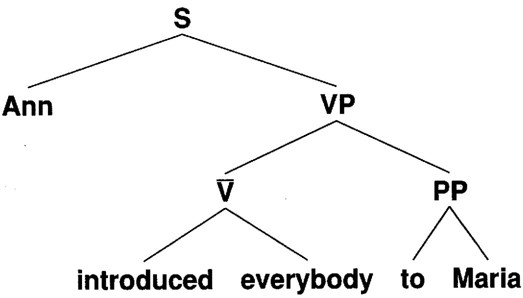
This proposal implies that all English quantifier phrases are multiply ambiguous. They all have multiple syntactic representations. Since the ambiguity is systematic, it's a benign case of ambiguity. On this proposal, the syntax of English doesn't have to specify that, say, **everybody$_1$** can only occur in subject position, and **everybody$_2$** can only occur in object position. As far as the syntax goes, any quantifier phrase with any subscript can occur in any position. If a quantifier phrase appears in the wrong position, its mother node is not interpretable, and the structure is automatically ruled out.

Exercise 1

Specify the semantic rule for **nobody$_2$**, and calculate the truth-conditions for the phrase structure tree corresponding to **somebody$_1$ greets nobody$_2$**.

Exercise 2

Design a new entry for "everybody" (**everybody$_3$**) that makes it possible to interpret phrase structure trees of the following kind:



Up to now in this section, we have limited ourselves to quantifying DPs that consist of one word, like "everybody", "something". But of course we must deal with the full range of quantifying DPs of the form [Det NP]. These are not listed in the lexicon, so if they are ambiguous, their ambiguity must be traceable to a lexical ambiguity in one of their constituents. The natural place to locate this ambiguity is in the determiner.

For instance, the DP "a linguist" ought to have an alternate meaning of type <<e,et>,et>[9] so that we can use it in object position in place of "somebody" in (2). If we keep the familiar meaning for **linguist** of type <e,t>, this means that the determiner must be of type <et,<<e,et>,et>> in this case. Specifically, we need the following entry:

(2) $[\![a_2]\!] = \lambda f \in D_{<e,t>} \; . \; [\lambda g \in D_{<e,<e,t>>} \; .$
$[\lambda x \in D \; . \; \text{for some } y \in D, f(y) = 1 \text{ and } g(y)(x) = 1]]$

Similarly for other determiners.

If every English determiner is multiply ambiguous in this way, then we would obviously be missing a generalization if we simply listed each reading for each determiner in the lexicon. Speakers of English presumably need not learn each reading separately: once they know what a determiner means in a subject DP, they can infer its meaning in object position without further evidence. So they must know some general rule by which the alternate readings of an arbitrary determiner are predictable from its basic type <et,<et,t>> meaning. In other words, there has to be a *lexical rule* like the following:

(3) For every lexical item δ_1 with a meaning of type <et,<et,t>>, there is a (homophonous and syntactically identical) item δ_2 with the following meaning of type <et,<<e,et>,<e,t>>:
$[\![\delta_2]\!] = \lambda f \in D_{<e,t>} \; . \; [\lambda g \in D_{<e,<e,t>>} \; . \; [\lambda x \in D \; . \; [\![\delta_1]\!](f)(\lambda z \in D \; . \; g(z)(x))]]$.

(3) automatically provides an alternate meaning for any arbitrary determiner. The only determiner entries we need to list individually are those of the simplest type <et,<et,t>>.[10]

7.2.2 *Excursion: flexible types for connectives*

Flexible types were first proposed not for quantifiers but for connectives like "and" and "or".[11] These seem to be able to coordinate phrases of a variety of different syntactic categories, with meanings of a corresponding variety of semantic types. For example:

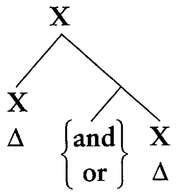
(4) $[_S[_S$ John stays at home] and $[_S$ Mary works]].

(5) Ann will be $[_{PP}[_{PP}$ in the garden] or $[_{PP}$ on the porch]].

(6) Bill $[_V[_V$ writes] and $[_V$ reads]] Portuguese.

(7) $[_{DP}[_{DP}$ A few books] or $[_{DP}$ a lot of articles]] will be read.

Suppose the structure of the coordinate phrase in each example is as follows:[12]

(8)  where X = S, PP, V, or DP

Suppose further that **and** and **or** have their familiar meanings from propositional logic or, more accurately, appropriate Schönfinkelizations thereof:

(9)
$$[\![\textbf{and}]\!] = \begin{bmatrix} 1 \rightarrow \begin{bmatrix} 1 \rightarrow 1 \\ 0 \rightarrow 0 \end{bmatrix} \\ 0 \rightarrow \begin{bmatrix} 1 \rightarrow 0 \\ 0 \rightarrow 0 \end{bmatrix} \end{bmatrix} \qquad [\![\textbf{or}]\!] = \begin{bmatrix} 1 \rightarrow \begin{bmatrix} 1 \rightarrow 1 \\ 0 \rightarrow 1 \end{bmatrix} \\ 0 \rightarrow \begin{bmatrix} 1 \rightarrow 1 \\ 0 \rightarrow 0 \end{bmatrix} \end{bmatrix}$$

Sentence (4), where X = S, is then straightforwardly interpretable. But (5) and (6) are not, because the denotations of PP and V in these examples are of type <e,t> and <e,<e,t>> respectively, therefore unsuitable to combine with $[\![\textbf{and}]\!]$ or $[\![\textbf{or}]\!]$.

A natural response is to posit a systematic ambiguity: There is not just one word "and", but a family of related words $\textbf{and}_1$, $\textbf{and}_2$, $\textbf{and}_3$, etcetera (and likewise for "or"). The most basic members of each family, $\textbf{and}_1$ and $\textbf{or}_1$, are of type <t,<t,t>> and have the meanings defined in (9). The next ones are of type <et,<et,et>>; for instance, $\textbf{or}_2$ is interpreted as follows.

(10) $[\![\textbf{or}_2]\!] = \lambda f \in D_{<e,t>} \,.\, [\lambda g \in D_{<e,t>} \,.\, [\lambda x \in D \,.\, [\![\textbf{or}_1]\!](f(x))(g(x))]]$.

This is the homonym of "or" that we must employ in (5) to make the sentence interpretable. Try it out to verify that it predicts the appropriate truth-conditions! Homonyms of yet more complicated types are needed for (6) and (7).

Exercise 1

Define appropriate meanings for $\textbf{and}_3$ and $\textbf{or}_4$ to be used in (6) and (7).

Exercise 2

What we have said here about the 2-place connectives "and" and "or" carries over to some extent to the 1-place connective negation. But the picture is less clear here because the word **not** has a much more limited syntactic distribution. Among the following examples parallel to (4)–(7) above, only (iv) is grammatical.

(i) $^*[_S$ Not $[_S$ John stays at home]].
*Not does John stay at home.

(ii) ?Ann will be $[_{PP}$ not $[_{PP}$ on the porch]].

(iii) *Bill $[_V$ not $[_V$ reads]] Portuguese.

(iv) $[_{DP}$ Not $[_{DP}$ everything]] will be read.

The most common position for **not** in English is none of the above, but rather as in (v).

(v) John doesn't stay at home.

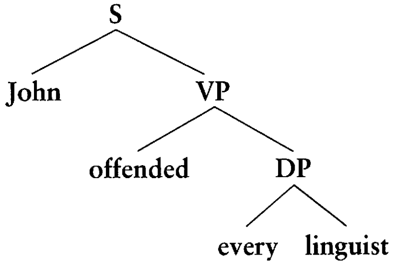
What denotations for **not** do we need to interpret (iv) and (v)?

The point of this excursion into the semantics of "and" and "or" was to lend plausibility to the view that flexible types are a systematic phenomenon in natural language, and not just an *ad hoc* device we need for the treatment of quantifiers.

7.3 Repairing the type mismatch by movement

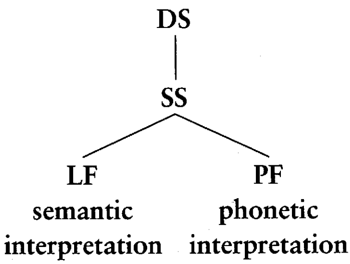
We have briefly looked at one way of assigning intuitively correct interpretations to VPs which contain quantificational objects. On that approach, the objects were left in place. We will now pursue in more depth an approach which maintains our original assumption that the determiner is unambiguous and can only be combined with two 1-place predicates. Since the overt syntactic structure of "John offended every linguist"

(1)



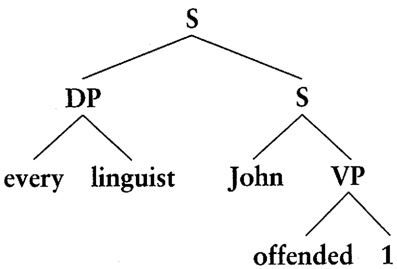
does not contain two such predicates, it follows that this structure cannot be the input to the semantic component. Rather, this sentence must have another structural description under which "every" combines with two constituents each having a denotation of type <e,t>. Such a structure can be created by *moving* the DP "every linguist".

To have a concrete proposal to work with, suppose here (and in the chapters to come) that we have a model of grammar like the inverted Y model of the Revised Extended Standard Theory and Government Binding Theory:[13]



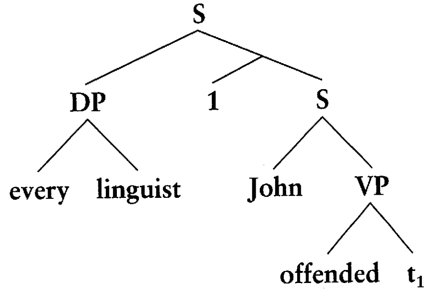
According to this model, semantic interpretation applies to representations on a level of Logical Form (LF), which is transformationally derived from S(urface) Structure (SS). The DP "every linguist" in (1) will move out of its VP and adjoin to S in the derivation from SS to LF. This movement operation, then, might feed semantic interpretation, but not necessarily phonetic realization, and might therefore be invisible. Like all movement operations, it leaves a trace. So the structure created by this movement has at least the following ingredients:

(2)



What we see in (2) is not quite enough yet to make an interpretable structure. Traces, on our assumptions from chapter 5, must bear an index to be interpretable. And since in a complete sentence every variable must be bound, the index on the trace has to be matched by an index on a variable binder somewhere. We propose that (3) below is a more complete and accurate representation of the structure created by moving "every linguist".

(3)



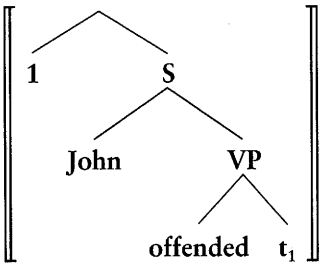
The indexed trace here is straightforwardly interpreted by our Traces and Pronouns Rule as a variable. The adjoined index right below the moved phrase is supposed to be the variable binder. This requires a slight generalization of our Predicate Abstraction Rule, which currently only covers variable binders of the form "such_i" "wh_i", "who_i" etcetera. The fact that the rule had to mention particular lexical items was undesirable to begin with, since it went against the spirit of type-driven interpretation that does not permit composition principles that mention lexical items. When you look at the Predicate Abstraction Rule a bit more closely, however, you will notice that all it needs to see is the index on the relative pronouns, not their particular shapes. Let's therefore adopt the following revision of the Predicate Abstraction Rule, with the understanding that "such" and the relative pronouns are to count as semantically vacuous items.

(4) *Predicate Abstraction Rule* (PA)
Let α be a branching node with daughters β and γ, where β dominates only a numerical index i. Then, for any variable assignment a, $[\![\alpha]\!]^a = \lambda x \in D\ .\ [\![\gamma]\!]^{a^{x/i}}$.

The interpretation of structure (3) is now straightforward. Inside the VP, the transitive V of type <e,<e,t>> composes by FA with the trace of type e, yielding a VP meaning of type <e,t>. This composes (again by FA) with the subject's meaning, here of type e, to yield a type t meaning for the lower S. Concretely, the interpretation obtained for this lower S-node is this:

For any a: $[\![\textbf{John offended t}_1]\!]^a = 1$ iff John offended a(1).

At the next higher node, PA applies, and yields the following meaning of type <e,t>.



$= \lambda x \in D$. John offended x.

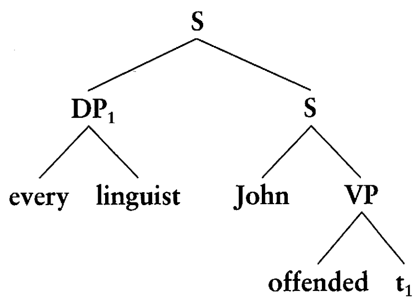
This is a suitable argument for the quantifying DP's meaning of type <<e,t>,t>, so FA can apply to the top S-node, and we obtain:

$[\![(3)]\!] = 1$ iff for every x such that x is a linguist, John offended x.

We have obtained the correct result, and we have been able to derive it here without resorting to a type-shifted homonym of "every". We used exactly the denotation for "every linguist" that this DP has in subject position. There was no type mismatch at any point in the tree, thanks to the movement operation that applied to the object. This operation effected two crucial changes: it provided the transitive verb with an argument of type e, and the moved quantifier phrase with an argument of type <e,t>.

So far, so good. But the structure in (3) is not exactly what syntacticians imagine to be the output of movement.[14] When we introduced the idea of moving the object in (2), you probably expected a representation like (5) rather than (3).

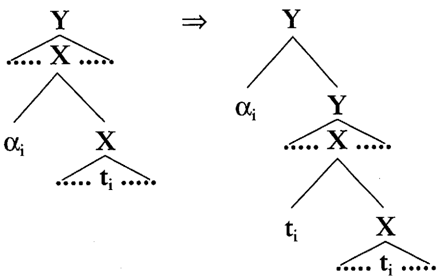
(5)



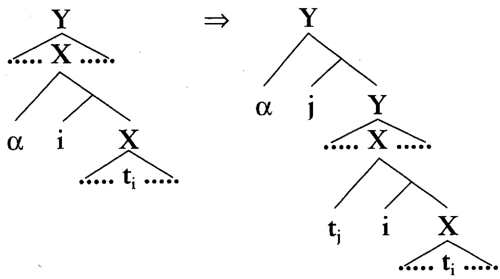
The difference between (3) and (5) is that in (5) the higher index forms a constituent with the moved phrase, whereas in (3) it forms a constituent with the moved phrase's scope (= sister). Is this a substantive difference or just an insignificant variation of notation?

From the point of view of semantics, the difference is clearly substantive (or else we wouldn't have bothered to depart from familiar custom): the constituency in (5) would not have been interpretable by means of our existing inventory of composition principles (or minor revisions thereof).[15] How about from the point of view of syntax? There it is less obvious that it makes a real difference whether we assume (3) or (5). Certain principles that refer to co-indexing will have to be trivially reformulated, but otherwise, how could we possibly tell the difference? One apparent difference in predictions concerns the question of what happens when a moved phrase moves further (as in successive cyclic movement). (5) leads us to expect that the whole unit including the index moves, as in (6), whereas (3) would seem to imply that a new binder–trace relationship is created (possibly marked by a new index $j \neq i$), as in (7).

(6)



(7)



These surely *look* different, but it is harder than one might think to come up with empirical evidence that would bear on the choice. To our knowledge, the issue has not been investigated, and there are no obvious reasons why (3) (and (7)) wouldn't be just as suitable for the purposes of syntactic theory as (5) (and (6)). So we will assume henceforth that whenever we find representations like (5) in the syntactic literature, we can simply treat them as abbreviations for representations like (3).

7.4 Excursion: quantifiers in natural language and predicate logic

The movement analysis that we just looked at makes it possible for us to maintain that a quantifier word such as **every** or **some** *always* combines with two 1-place predicates to make a sentence. For instance, the sentence **every cat is hungry** is interpreted by combining the denotations of **every**, **cat**, and **is hungry**. And the denotation of the sentence **Ann fed every cat** is computed by combining the denotations of **every**, **cat**, and a predicate abstract of the form **[i [Ann fed** $\mathbf{t_i}$**]]**. In this respect, English quantifiers are quite unlike the quantifiers ∀ and ∃ of standard predicate logic[16] (henceforth PL). The latter seem to have the syntax of sentence operators. For instance, a sentence like **∃x [hungry(x)]** is built by prefixing **∃x** to a sentence **hungry(x)**.[17]

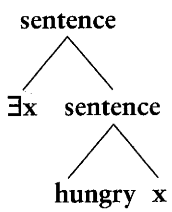
Thus we see two salient differences between English determiners and PL quantifiers. One difference concerns the *types* of arguments they combine with: English determiners combine with (1-place) *predicates*, PL quantifiers with *sentences*. The other difference has to do with the *number* of arguments each needs in order to make a complete sentence: English determiners need *two* arguments, PL quantifiers only *one*.

As we will see in this section, the first of these differences is relatively superficial, and can easily be removed by a minor departure from the standard syntactic and semantic treatment of PL. But the second difference is more substantive, and points to an important property of natural languages that is not shared by PL.

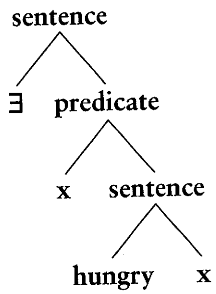
7.4.1 Separating quantifiers from variable binding

As we have mentioned, the syntax of PL is customarily set up in such a way that a quantified expression like **∃x [hungry(x)]** is built up from a sentence **hungry(x)** by adding a prefix **∃x**. But this is not the only possible way of doing it. One could equally well think of **∃x [hungry(x)]** as built up from the bare quantifier **∃** and an expression **x [hungry(x)]**.

Customary parse:



Alternative parse:



Semantically, one would then want to treat the constituent **x [hungry(x)]** as a predicate abstract: It would denote the set {x : x is hungry}.[18] The appropriate semantic rule would be analogous to our Predicate Abstraction Rule.

This way, a bare quantifier such as ∃ or ∀ can be seen to combine with one predicate to form a sentence. It is thus a *1-place second-order predicate*. In other words, a bare quantifier expresses a property of sets of individuals. For example, ∃ expresses the property of being a non-empty set, and ∀ the property of including all elements of D_e. Semantic rules to this effect would be as follows.

Let α be any predicate. Then
(i) ∃α is true iff α denotes a non-empty set;
(ii) ∀α is true iff α denotes D.

This reinterpretation of the syntax and semantics of PL does not alter anything substantial. We still generate all the same PL sentences (except with slightly different structures), and they get exactly the same truth-conditions as before. It does, however, eliminate the first of our two differences between English determiners and PL quantifiers: both now take *predicates* rather than sentences as arguments.

7.4.2 *1-place and 2-place quantifiers*

There remains the difference in the *number* of arguments: an English quantificational determiner requires *two* predicates to form a complete sentence, a PL quantifier needs only *one*. *This* difference is actually quite an obstacle to a mechanical translation of English into PL, and it has a lot to do with the most common mistakes that people make in a beginning logic class. Let's reflect a little on what we do when we are asked to symbolize English sentences in PL.

Consider two simple English sentences and their PL translations:

(1) (a) **Some cat is gray.**
(b) **∃x [cat(x) & gray(x)]**

(2) (a) **Every cat is gray.**
(b) $\forall x$ [cat(x) $\rightarrow$ gray(x)]

The transition from (a) to (b) involves an evident distortion of syntactic structure, and even the insertion of new lexical items (the connectives & and →). Why are such changes necessary?

One way of putting the point is as follows. The PL quantifiers ∃ and ∀ don't really symbolize the English *determiners* **some** and **every**; rather, they correspond to certain *complete DPs* of English: namely, the DPs **some individual** and **every individual**.[19] So when we translate English sentences into PL, we must first paraphrase them in such a way that all instances of **some** and **every** combine only with the noun **individual**. Combinations of the form D + noun for any other noun must be paraphrased away. For instance, we can get rid of **some cat** in (1a) by paraphrasing (1a) as (1c), and we can eliminate **every cat** from (2a) by paraphrasing it as (2c) or (equivalently (2c′)[20]).

(1) (c) **Some individual is [both a cat and gray].**

(2) (c) **Every individual is [gray if a cat].**
(c′) **Every individual is [either not a cat or gray].**

If it weren't for the existence of paraphrases of this kind, we wouldn't be able to express the meanings of (1a) and (2a) in PL at all.

In a way it is a lucky coincidence that such paraphrases exist. It depends on the particular lexical meanings of **some** and **every**. The meaning of English **some**, for instance, ensures that, for two arbitrary predicates α and β, **some** α β is equivalent to **some individual [α and β]** (and not, for example, to **some individual [[not α] or β]**, except for certain specific choices of α and β). Likewise, it is a fact about the particular meaning of English **every** that **every** α β is always equivalent to **every individual [[not α] or β]** (but normally has very different truth-conditions from **every individual [α and β]**).

A question that suggests itself at this point is the following: Do *all* English determiners support systematic equivalences of this kind? More precisely, if δ is an arbitrary determiner of English, is there going to be some way of forming out of two predicates α and β some complex predicate F(α, β) by means of **and** and **not**,[21] in such a way that (for arbitrary α and β) [δ α] β is equivalent to [δ **individual(s)**] F(α, β)?

It turns out that the answer to this question is "no". The standard counterexample is the determiner **most**. It has been proved that there is no way to construct a predicate F(α, β) out of α and β by conjunction and negation so that **most** α β is always equivalent to **most individuals** F(α, β). We will not attempt to prove this general claim here – you will have to take our word for it.[22] But the following exercise should at least give you some intuitive appreciation of it.

Exercise

Suppose we add a quantifier **M** to PL and give it the following interpretation:

M α is true iff α is true of more individuals than it is false of.

(We presuppose here the alternative parse introduced in 7.4.1.) An appropriate English gloss for **Mx [F(x)]** would be "most individuals are F".

Now consider the following two proposed PL symbolizations for the English sentence **Most cats are gray**.

(i) **Mx [cat(x) & gray(x)]**
(ii) **Mx [cat(x) → gray(x)]**

Neither of them is right. Explain why not. For each symbolization, describe a possible state of affairs where its truth-value deviates from the intuitive truth-value of the English sentence.

What exactly is the significance of the fact that natural languages have determiners like **most**? One thing that it implies is that English **most** sentences cannot be translated into PL. This in itself, however, would not be so interesting if it could be traced to a mere limitation in the *lexicon* of PL. As it turns out, there are other English quantifiers that cannot be translated into PL: for instance, **finitely many**. As logicians have proved, an English statement of the form

(3) [**finitely many** α] β

cannot be symbolized in PL. But in this case, the problem is not that the English determiner **finitely many** takes two arguments rather than just one. (3) *is* equivalent to (4).

(4) **finitely many individuals** [α **and** β]

(For instance, **Finitely many natural numbers are smaller than 100** has exactly the same truth-conditions as **Finitely many individuals are natural numbers and smaller than 100.**) In this case, the problem is simply that PL has so few quantifiers: namely, just $\exists$ and $\forall$. This is, so to speak, a mere limitation in the *lexicon* of PL, and it could be relieved, without any change to the syntax and compositional semantics of PL, by adding further quantifier symbols. For example, we might add to $\exists$ and $\forall$ such symbols as $\sim\exists$, $\exists!$, $\exists^{\infty}$, and $\sim\exists^{\infty}$, and define their semantics as follows:

(5) Let α be a 1-place predicate. Then:
- (i) $\sim\exists\ \alpha$ is true iff α denotes the empty set;
- (ii) $\exists!\alpha$ is true iff α denotes a singleton set;
- (iii) $\exists^{\infty}\ \alpha$ is true iff α denotes an infinite set;
- (iv) $\sim\exists^{\infty}\ \alpha$ is true iff α denotes a finite set.

By enriching the lexicon of PL in this way, we could make expressible such English statements as (3). But we still couldn't express **most**. The problem with **most** runs deeper. It is an irreducibly *2-place* quantifier, and adding more 1-place quantifiers to PL will therefore not help.

7.5 Choosing between quantifier movement and *in situ* interpretation: three standard arguments

We began this chapter by observing a problem with quantifiers in object position. Given the semantic type for quantificational DPs that we had come up with in chapter 6, we predicted that they were interpretable only when they were sisters to a phrase of type <e,t>. This prediction was falsified by the distribution of quantificational DPs in English – or at least, it was falsified by their surface distribution. We presented two different responses to this problem. One response took for granted that the surface distribution of quantifiers essentially reflected their distribution at the level at which semantic interpretation takes place, and it consisted of positing a systematic *type ambiguity* in the lexical meanings of determiners. The other response was to stick by the prediction and draw appropriate conclusions about the *syntactic* behavior of quantifiers. For the time being, the choice between them seems to be open. There is a trade-off between positing lexical type-shifting rules and positing movement. Pending independent motivation for either one or the other, the choice seems to be just a matter of taste.

But maybe if we broaden our scope of linguistic phenomena, we will find that there *is* some independent evidence for one choice over the other? Indeed, this has been claimed to be the case. Specifically, it has been argued that the movement approach has a decided advantage in dealing with three phenomena: scope ambiguity, antecedent-contained deletion, and bound-variable anaphora.

We will present some versions of these standard arguments in this section. But we should note at the outset that their ultimate force is very difficult to assess. For one thing, proponents of the *in situ* approach have developed ever more sophisticated answers to these arguments, and the most recent theories that they have developed differ from the movement-based theories in so many respects at once that comparison becomes a very global and difficult matter. And an even more important difficulty is that the issue is not really an all-or-nothing choice. It is entirely conceivable that there exist both quantifier movement and mechanisms for *in situ* interpretation (of nonsubject quantifiers), and that the grammars of natural languages employ one option in some constructions and the other in others. In fact, the majority of scholars who have done influential work in this domain have (explicitly or implicitly) taken such mixed positions of one sort or another.[23]

So we could not possibly purport here to give decisive evidence in favor of a pure movement approach. The best we can hope for is to give you some preliminary idea of what sorts of considerations are pertinent, and what price you have to pay for a particular decision.

7.5.1 *Scope ambiguity and "inverse" scope*

If all quantifiers are interpreted in their surface positions, then a given surface structure with two or more quantifiers in it can receive only one reading, unless we admit ever more complicated types, going far beyond the flexible types options we have considered so far.[24] Take sentence (1):

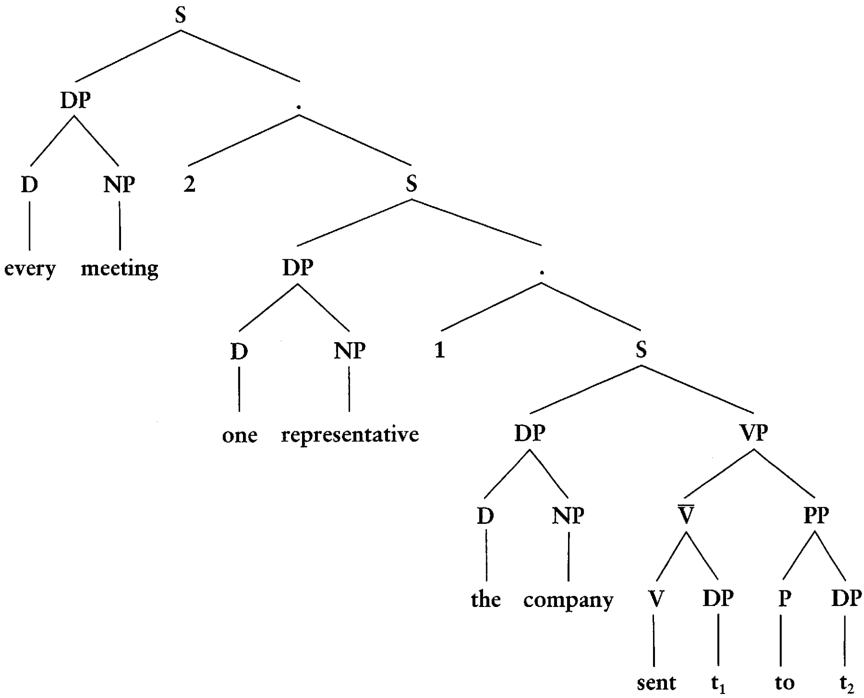
(1) Somebody offended everybody.

(1) has two readings, not just one. It can mean that there is somebody who offended everybody. Or else it can mean that for everybody there is somebody that s/he offended. On either of the proposals for *in situ* interpretation that we discussed above, we predict only the reading where there is somebody who offended everybody.

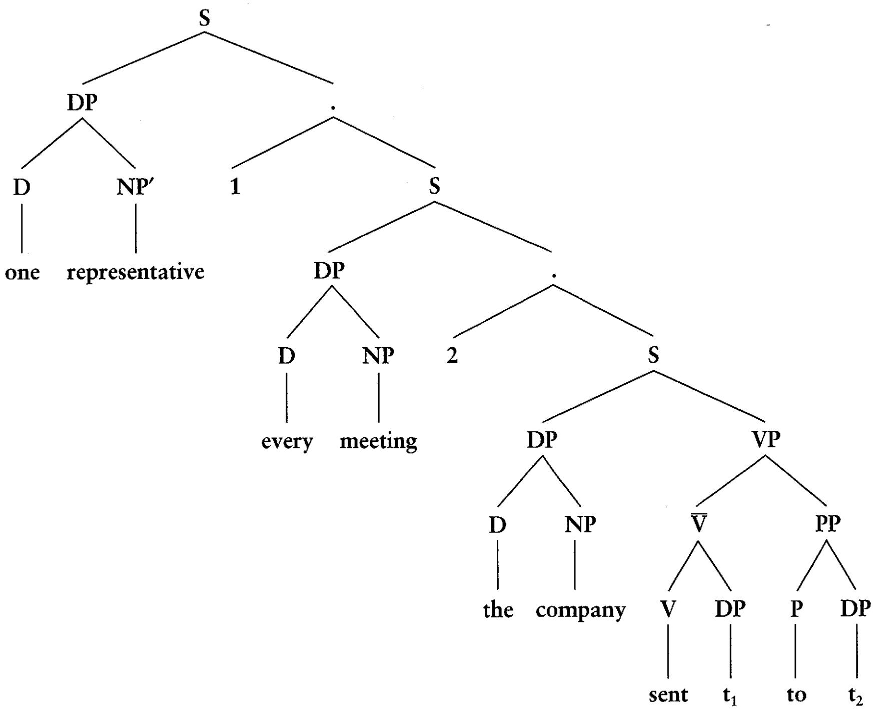
Once we move quantifiers, however, it is trivial to derive several distinct and truth-conditionally non-equivalent LFs from a given SS. For instance, we can derive at least the LFs (3) and (4) from the SS of sentence (2).

(2) The company sent one representative to every meeting.

(3)


S
DP .
D NP 2 S
every meeting DP .
D NP 1 S
one representative DP VP
D NP V̄ PP
the company V DP P DP
sent t1 to t2

(4)


S
DP .
D NP′ 1 S
one representative DP .
D NP 2 S
every meeting DP VP
D NP V̄ PP
the company V DP P DP
sent t1 to t2

If we apply our semantic rules to these two structures, they assign them distinct truth-conditions. We can prove this, for example, by considering the following state of affairs:

(5) There is exactly one company, c.
There are exactly two representatives, r_1 and r_2.
There are exactly three meetings, m_1, m_2, and m_3.
c sent r_1 to m_1, r_2 to both m_2 and m_3, and nobody else to anything else.

We can show that one of the LFs above is true in this scenario and the other one is false. More specifically:

(6) *Claim 1*: Given the facts in (5), $[\![(3)]\!] = 1$.
Claim 2: Given the facts in (5), $[\![(4)]\!] = 0$.

Here is a proof of claim 2:

(i) Work out the extension of the larger predicate abstract in (4):[25]
Let's abbreviate that predicate abstract by "α":
$\alpha :=$ **1[every meeting 2[the company sent t_1 to t_2]]**
Now let $x \in D$ be an arbitrary individual. Then:
$[\![\alpha]\!](x) = 1$
iff
$[\![$**[every meeting 2[the company sent t_1 to t_2]]**$]\!]^{\varnothing^{x/1}} = 1$
iff
$[\![$**every meeting**$]\!]([\![$**2[the company sent t_1 to t_2]**$]\!]^{[1\to x]}) = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: $[\![$**2[the company sent t_1 to t_2]**$]\!]^{[1\to x]}(y) = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: $[\![$**the company sent t_1 to t_2**$]\!]^{[1\to x]^{y/2}} = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: $[\![$**the company sent t_1 to t_2**$]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]} = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: $[\![$**sent t_1 to t_2**$]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]}(c) = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: $[\![$**sent**$]\!]([\![$**t_1**$]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]})([\![$**$t_2$**$]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]})(c) = 1$
iff
for every $y \in \{m_1, m_2, m_3\}$: c sent $([\![t_1]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]})$ to $([\![t_2]\!]^{\left[\begin{smallmatrix}1\to x\\2\to y\end{smallmatrix}\right]})$
iff
for every $y \in \{m_1, m_2, m_3\}$: c sent x to y
iff
c sent x to m_1, m_2, and m_3.

According to (5), no individual x satisfies this condition, so we have determined:
For no $x \in D$: $[\![\alpha]\!](x) = 1$.

(ii) It follows that there is no $x \in D$ such that $[\![\textbf{representative}]\!](x) = 1$ and $[\![\alpha]\!](x) = 1$.
According to the lexical entry of $[\![\textbf{one}]\!]$[26], this implies that
$[\![\textbf{one}]\!]([\![\textbf{representative}]\!])([\![\alpha]\!]) = 0$.
But $[\![\textbf{one}]\!]([\![\textbf{representative}]\!])([\![\alpha]\!]) = [\![(4)]\!]$, so we have proved our claim 2.

Exercise

Prove claim 1 of (6). Annotate each step in your proof with references to all the rules and definitions that you are applying.

May[27] argued that the case for quantifier movement is even stronger when we consider not just examples like (2), but also examples like (7).

(7) One apple in every basket is rotten.

May's point about example (7) is that its *most natural* reading (perhaps even its only reading) cannot be generated by an *in situ* approach, but is straightforwardly predicted by quantifier movement.

Consider what we get on the *in situ* analysis with flexible types: "in" has the same type of meaning as a transitive verb, <e,et>. So "every" must have its type <et,<<e,et>,et>> meaning here. Thus we get:

$[\![\textbf{in every basket}]\!] = \lambda x$. for every basket y, x is in y

We proceed to the next node up by Predicate Modification and get:

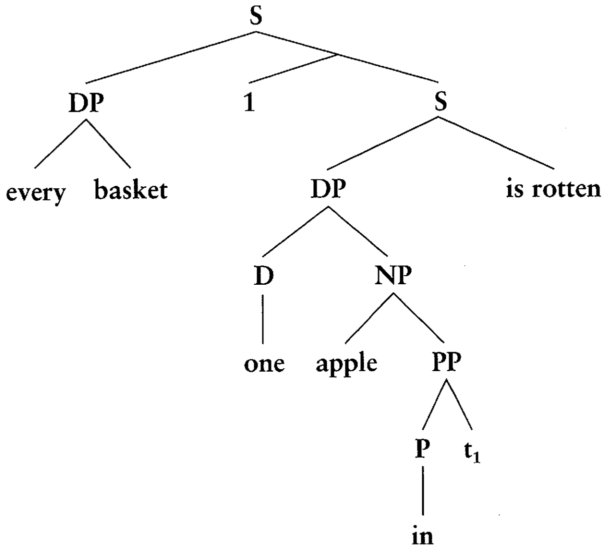
$[\![\textbf{apple in every basket}]\!] = \lambda x$. x is an apple and for every basket y, x is in y

Finally we apply the meanings of "one" and "rotten" and derive the following truth-condition for (7):

$[\![(7)]\!] = 1$ iff
there is at least one $x \in D$ such that x is an apple and x is in every basket and x is rotten.

This is definitely not the salient meaning of (7).[28] What (7) rather seems to mean is that every basket contains one rotten apple.[29] By moving the quantifier phrase "every basket", we can easily generate this meaning:

(7′)



Exercise

Calculate the truth-conditions of (7′).

7.5.2 *Antecedent-contained deletion*

Another phenomenon that is a potential problem for an *in situ* interpretation of quantifier phrases is so-called Antecedent-Contained VP Deletion, illustrated by the following example.

(8) I read every novel that you did.

Antecedent-Contained VP Deletion is an instance of VP deletion.[30] Here are some more examples of VP deletion.

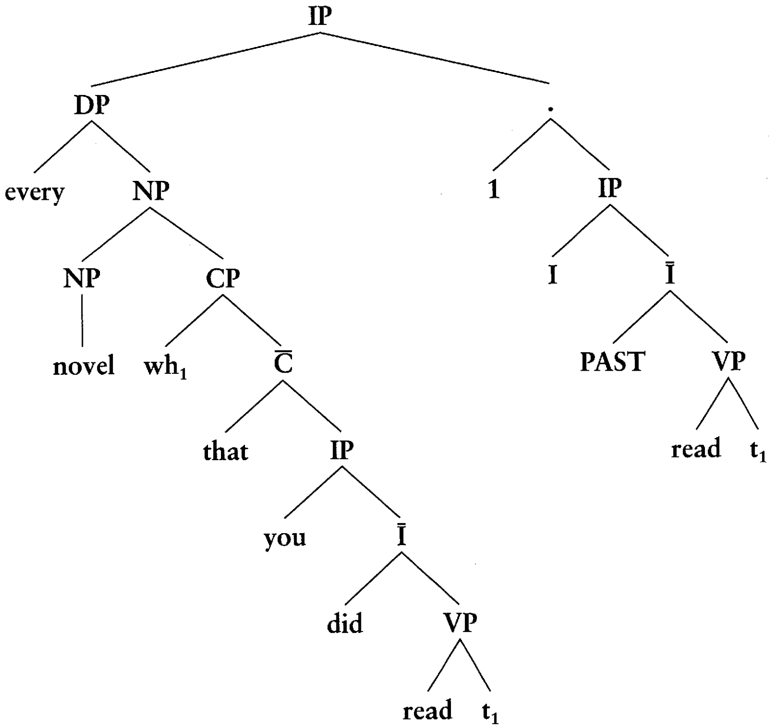
(9) I read *War and Peace* before you did.

(10) I went to Tanglewood even though I wasn't supposed to.

(11) You may very well put this experience behind you, but you shouldn't think that you really have to.

Suppose that in the construction of those sentences, a VP is deleted in the derivation from SS to PF. This VP, then, is not pronounced, but is available for semantic interpretation. The deletion is licensed by a preceding VP that has the same shape. With sentences (9)–(11), it is easy to see what the licensing VP would be. In (9), it's "read *War and Peace*". In (10), it's "go to Tanglewood", and in (11), it's "put this experience behind you". But now go back to (8). What has to be deleted in (8) is a VP consisting of the verb **read** and a trace that is bound by the (index introduced by) the relative pronoun. The surface structure of (8) does not seem to provide an antecedent VP that looks the same. If the object quantifier phrase in (8) is allowed to move out, however, we obtain the right antecedent VP. The LF for (8) will now look at follows:[31]

(8′)



So in examples involving Antecedent-Contained VP Deletion, we seem to need movement of object DPs anyway. So there are reasons to assume such movement is in principle available, and we might as well use it to resolve the type mismatch in simple sentences with quantified objects as well. An additional mechanism of type-shifting is, at best, redundant.[32]

7.5.3 *Quantifiers that bind pronouns*

Consider the following sentences:

(12) Mary blamed herself.

(13) No woman blamed herself.

(14) Every woman blamed herself.

Sentences (12)–(14) contain instances of reflexive pronouns. Reflexive pronouns are necessarily anaphoric. If a pronoun is used anaphorically, its value is determined by its antecedent. If the antecedent is a proper name, then a pronoun that is anaphorically related to it might simply inherit the proper name's referent as its semantic value. But what if the antecedent is a quantifier phrase? (13) is not synonymous with (13′), and (14) is not synonymous with (14′). Hence we don't seem to be able to claim that reflexives always inherit the denotation of their antecedent.

(13′) No woman blamed no woman.

(14′) Every woman blamed every woman.

Reflexives are not the only pronouns that can be anaphorically related to quantifier phrases. Pronouns like **he, she, it** have such uses as well. This is shown by the following examples:

(15) *No man* noticed the snake next to *him.*

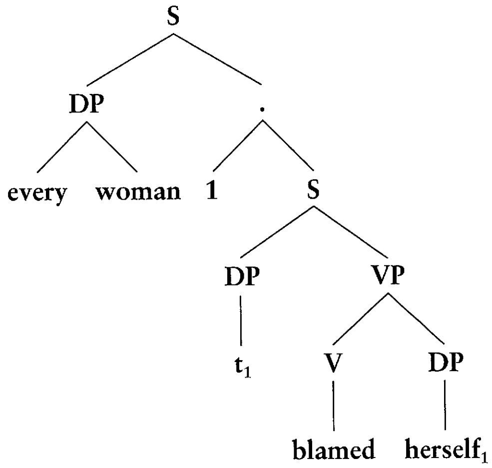
(16) We showed *every woman* a newspaper article with a picture of *her.*

Again, it would not do to say that these pronouns simply inherit the denotations of their antecedents. (15) does not mean the same as "No man noticed the snake next to no man".

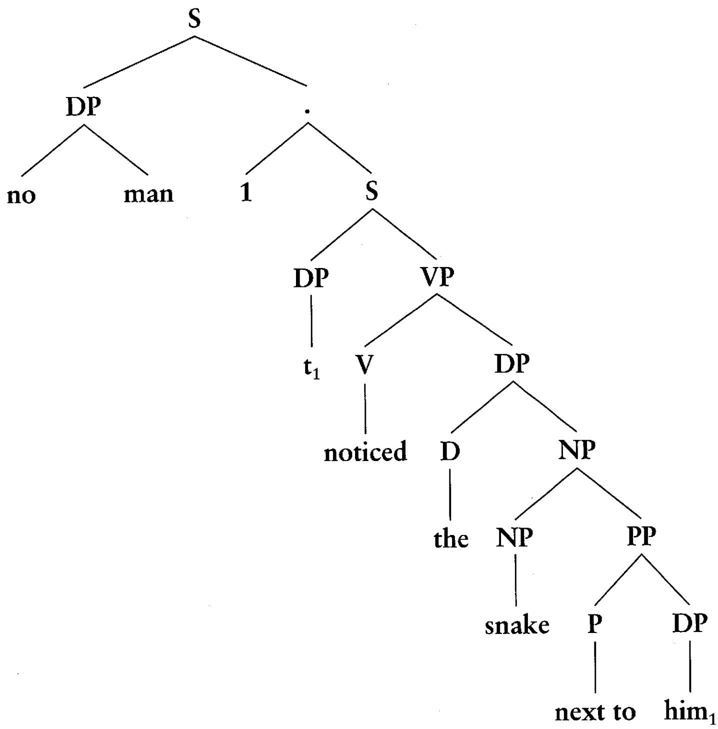
So how *should* we interpret these reflexives and pronouns? It seems that they behave as bound variables. We have successfully treated some other cases of bound-variable pronouns in the chapter on relative clauses. Can't we just combine our treatment of pronouns from that chapter with the present treatment of quantifier phrases? Let's try.

On the quantifier movement approach, the matter is straightforward. Although subject quantifiers are not forced to move in order to avoid a type mismatch, there is no reason why they shouldn't be *allowed* to move. Suppose we exercise this option, and also choose to co-index the pronoun with the trace left by the moved quantifier. This leads to the following representations.

(17)



(18)



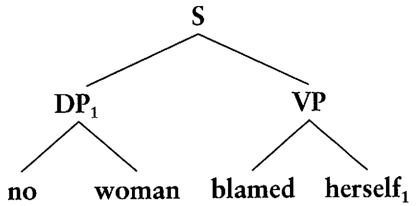
The rules for interpreting these structures are already all in place. Look up, in particular, our Traces and Pronouns Rule. Although we did not have reflexive pronouns in mind when we first wrote that rule, our formulation just mentions a "pronoun"; so it will apply equally to "herself$_1$" in (17) and "him$_1$" in (18) – provided only that the lexicon classifies both as pronouns.

Exercise

Calculate the predicted truth-conditions for one of the structures (17), (18).

On a pure *in situ* approach to quantifiers, it is less obvious how to derive the appropriate meanings. Suppose the pronoun is co-indexed with its antecedent, but that antecedent is an unmoved DP:

(19)



Does (19) express the correct truth-conditions?

The interpretation of the VP-node in (19) is straightforward. Given that we are interpreting "herself$_1$" as a variable, by the Traces and Pronouns Rule, it receives the following assignment-dependent meaning:

For any a: $[\![\textbf{blamed herself}_1]\!]^a = \lambda x\ .\ x$ blamed a(1).

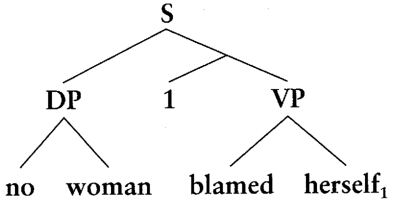
But how do we continue up to the S-node? If we simply compose the VP meaning with the meaning of the DP "no woman", ignoring the index on the latter, we wind up with this:

For any a: $[\![(19)]\!]^a = 1$ iff no woman blamed a(1).

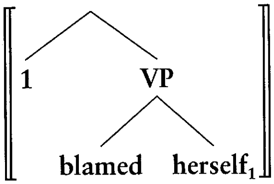
This cannot be what we want, since it does not give us an assignment-independent truth-value for (19).

Somehow, we must interpret (19) in such a way that the variable 1 gets bound. Where is the variable binder that does this? The only candidate is the index on the DP "no woman". In light of our earlier discussion, let's try to think of (19) as an abbreviation for (19′).

(19′)



Is this an interpretable structure? We get without problems to the (unlabeled) node above VP. According to our current formulation of PA, this denotes the following function:

 $= \lambda x\ .\ [\![\textbf{blamed herself}_1]\!]^{[1\to x]}$

This turns out to be a function of type <e,<e,t>>. (*Exercise*: Calculate this function, and find an English word or phrase that actually denotes it.) So it is not a suitable argument for the basic type <et,t> meaning of "no woman", but forces us to employ the alternate type <<e,et>,et>. But then the whole sentence in (19′) receives a meaning of type <e,t> instead of a truth-value! (*Exercise*: Calculate the predicted meaning of (19′), and find an English word or phrase that actually denotes it.)

So reading (19) as short for (19′) did not work in this case. What other options do we have? Well, one thing we can certainly do is take (19) at face value and introduce a new composition rule specifically for structures of this form. Here is a proposal that yields the correct predictions for our example:

(20) If α has the daughters β_i and γ, where $[\![\beta]\!]^a \in D_{<et,t>}$ for all a, then, for any assignment a: $[\![\alpha]\!]^a = [\![\beta]\!]^a(\lambda x\ .\ [\![\gamma]\!]^{a^{x/i}}(x))$

Applying this to (19′), we obtain:

$[\![19']\!] = 1$
iff (by (20))
$[\![\textbf{no woman}]\!](\lambda x\ .\ [\![\textbf{blamed herself}_1]\!]^{a^{x/i}}(x)) = 1$
iff (by meaning of "no woman")
there is no woman y such that $[\lambda x\ .\ [\![\textbf{blamed herself}_1]\!]^{a^{x/i}}(x)](y) = 1$

iff (by definition of λ-notation)
there is no woman y such that $[\![\textbf{blamed herself}_1]\!]^{a^{y/1}}(y) = 1$
iff (by meaning of VP, as calculated earlier)
there is no woman y such that y blamed $a^{y/1}(1)$
iff (by definition of assignment modification)
there is no woman y such that y blamed y.

This works, but the addition of a new composition rule is costly, and, *ceteris paribus*, we would prefer an alternative that avoids it. Moving the quantified subject seems to be just such an alternative. By doing that, we were able to interpret the examples with bound-variable pronouns with no more semantic rules than we had already motivated before.

Exercise

It might be suspected that the problem we have just outlined for the *in situ* approach is really an artifact of an inadequate view of pronouns. Wouldn't things be easier if we didn't think of "herself" as a variable in the first place? Then we wouldn't need to worry about getting it bound. In this exercise, you get to explore a non-variable semantics of reflexives which at first sight is quite attractive.[33]

Maybe a reflexive is a special sort of second-order predicate, as in the following lexical entry:

$[\![\textbf{herself}]\!] = \lambda f \in D_{<e,<e,t>>} . \lambda x \in D . f(x)(x)$

This meaning is of type <<e,et>,et> so as to be able to take the <e,<et>> type meaning of a transitive verb like **blame** as its argument and yield an <et> type VP meaning as the value.

(a) Show how sentence (13) above is interpreted on this proposal.
(b) What does the proposal predict for the following example?

No woman bought a book about herself.

Draw a tree for this example; then work through it under the flexible types theory that was presented above, using suitable readings for the type-ambiguous determiners "no" and "a".

Notes

1 M. Dummett, *Frege. Philosophy of Language*, 2nd edn (Cambridge, Mass., Harvard University Press, 1981), p. 8. Frege's *Begriffsschrift* was published in 1879.
2 Useful handbook articles are A. von Stechow, "Syntax and Semantics," in A. von Stechow and D. Wunderlich (eds), *Semantik/Semantics. An International Handbook of Contemporary Research* (Berlin and New York, de Gruyter, 1991), pp. 90–148; J. van Eijck: "Quantification," in ibid., pp. 459–87; P. Jacobson, "The Syntax/Semantics Interface in Categorial Grammar," in S. Lappin (ed.), *The Handbook of Contemporary Semantic Theory* (Oxford, Basil Blackwell, 1996), pp. 89–116.
3 R. Cooper, "Montague's Semantic Theory and Transformational Syntax" (Ph.D. dissertation, University of Massachusetts, Amherst, 1975), *idem*, *Quantification and Syntactic Theory* (Dordrecht, Reidel, 1983).
4 M. Schönfinkel, "Über die Bausteine der mathematischen Logik," *Mathematische Annalen*, 92 (1954), pp. 305–16; H. Curry and R. Feys, *Combinatory Logic* (Amsterdam, North-Holland, 1958).
5 A. Szabolcsi, "Bound Variables in Syntax (Are There Any?)," in *Proceedings of the Sixth Amsterdam Colloquium* (University of Amsterdam, Institute for Language, Logic and Information, 1987), pp. 331–51; M. Steedman, "Combinators and Grammars," in R. T. Oehrle, E. W. Bach, and D. Wheeler (eds), *Categorial Grammars and Natural Language Structures* (Dordrecht, Reidel, 1988), pp. 417–42; M. J. Cresswell, *Entities and Indices* (Dordrecht, Kluwer, 1990); P. Jacobson, "Antecedent-Contained Deletion without Logical Form," in C. Barker and D. Dowty (eds), *Proceedings of the Second Conference on Semantics and Linguistic Theory*, Columbus, Ohio State University, 1992, Ohio State University Working Papers in Linguistics, 40, pp. 193–214; P. Jacobson, "Bach–Peters Sentences in a Variable-Free Semantics," in P. Dekker and M. Stokhof (eds), *Proceedings of the Eighth Amsterdam Colloquium* (University of Amsterdam, Institute for Logic, Language, and Computation, 1992), pp. 283–302; *idem*, "Flexible Categorial Grammar: Questions and Prospects," in R. Levine (ed.), *Formal Grammar: Theory and Implementation* (Oxford, Oxford University Press, 1992), pp. 168–242.
6 Some references: H. Hendriks, "Type Change in Semantics: The Scope of Quantification and Coordination," in E. Klein and J. van Benthem (eds), *Categories, Polymorphism and Unification* (Edinburgh, Centre for Cognitive Science, and Amsterdam, ITLI, University of Amsterdam, 1987), pp. 96–119; *idem*, *Studied Flexibility*, ILLC Dissertation Series (Amsterdam, ILLC, 1993); B. H. Partee and M. Rooth, "Generalized Conjunction and Type Ambiguity," in R. Bäuerle, C. Schwarze, and A. von Stechow (eds), *Meaning, Use and Interpretation of Language* (Berlin, de Gruyter, 1983), pp. 361–83.
7 That is the one which Richard Montague used in his paper "On the Proper Treatment of Quantification in Ordinary English" ("PTQ"), in R. H. Thomason (ed.), *Formal Philosophy. Selected Papers by Richard Montague* (New Haven, Yale University Press, 1974), pp. 247–70. Abstracting away from matters irrelevant to the present issue, Montague's lexical entries for transitive verbs specify denotations of type <<<e,t>,t>,<e,t>>. *Exercise*: Define a meaning of this type for the verb "offend", and show that your proposal correctly predicts the truth-conditions of sentence (1).
8 The subscripts on the quantifier phrases help distinguish between homonyms of different semantic types. They shouldn't be confused with the indices we use for the interpretation of pronouns and traces.

9 Here and below, we sometimes leave out the innermost angled brackets and commas in complicated type expressions. For instance, "<<e,et>,et>" abbreviates "<<e,<e,t>>,<e,t>>". This notational practice improves readability without introducing any ambiguity.

10 Actually, this is not quite right: (3) as formulated generates only variants of type <et,<<e,et>,<e,t>>, not those of even more complex types which we would need, e.g., in the innermost object position of a 3-place verb like "introduce". It is possible, however, to formulate a more general version of (3) which recursively defines meanings for a whole infinite supply of alternate determiners of all the types that could possibly be required. If you are interested, you can consult, for instance, Mats Rooth's 1985 UMass Ph.D. thesis, "Association with Focus."

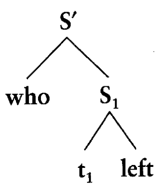
11 For an extensive discussion, see E. L. Keenan and L. Faltz, *Boolean Semantics* (Dordrecht, Reidel, 1983).

12 We assume binary branching because our system of composition rules is not equipped to deal with ternary branching nodes. If instead we extended the system so that it could deal with ternary branching, this should not substantially affect the point about type flexibility that we are interested in at the moment.

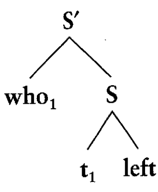
13 N. Chomsky, "Conditions on Rules of Grammar," *Linguistic Analysis*, 2 (1976), pp. 303–51; *idem*, *Lectures on Government and Binding* (Dordrecht, Foris, 1981). Most of what we have to say about quantifier movement applies to more recent instantiations of the movement theory as well. Consult C.-T. J. Huang, "Logical Form," in G. Webelhuth (ed.), *Government and Binding Theory and the Minimalist Program* (Oxford, Basil Blackwell, 1995), pp. 125–75. See also N. Hornstein, *Logical Form. From GB to Minimalism* (Oxford, Basil Blackwell, 1995).

14 At least not in most of the syntactic literature. We are aware of one exception: Edwin Williams and Henk van Riemsdijk, for reasons of their own, have actually proposed syntactic structures more like (3) than like (5), not only for QR but also for *wh*-movement. See, e.g., Edwin Williams, "A Reassignment of the Functions of LF," *Linguistic Inquiry*, 17/2 (1986), pp. 265–99, for an argument why "Who left?" should have structure (i) instead of (ii).

(i)



(ii)



15 The best we could do if forced to interpret (5) is to add a new composition principle which wraps up predicate abstraction and functional application of quantifier to argument all in one operation. *Exercise*: Formulate that principle.

16 See L. T. F. Gamut, *Logic, Language and Meaning*, vol. 1: *Introduction to Logic* (Chicago, University of Chicago Press, 1991), ch. 3, or any other logic text.

17 We are using "sentence" here in the sense of what logicians call a "formula". "Sentences" in the narrower sense of logic are only those formulas that don't contain any variables free. Our non-standard terminology may not be optimal for the purposes of logicians, but it is more natural from a linguist's point of view, and it facilitates the comparison between natural language and logic that interests us here.

18 Or, more accurately, the characteristic function of this set. As usual, we will indulge in set talk.
19 Where the predicate **individual** is to be understood in its most general sense, in which it is true of every element of D.
20 The equivalence of (2c) and (2c′) is a consequence of the semantics of **if**. As you recall from propositional logic, **if** ϕ, ψ is equivalent to [**not** ϕ] **or** ψ.
21 We could also say here: "by means of **and**, **not**, **or**, **if**, and **iff**". But this wouldn't make any difference, because it is known that **or**, **if**, and **iff** are definable in terms of **and** and **not**. (See Gamut, *Logic*, vol. 1, ch. 2, or any other introduction to propositional logic.)
22 There are proofs in the literature, see, e.g., J. Barwise and R. Cooper, "Generalized Quantifiers and Natural Language," *Linguistics and Philosophy*, 4 (1981), pp. 159–219.
23 This includes R. Montague, in Thomason (ed.), *Formal Philosophy*; R. Fiengo and J. Higginbotham, "Opacity in NP," *Linguistic Analysis*, 7 (1981), pp. 395–421; C.-T. J. Huang, "Logical Relations in Chinese and the Theory of Grammar" (Ph.D. dissertation, MIT, 1982); R. May, *Logical Form: Its Structure and Derivation* (Cambridge, Mass., MIT Press, 1985); T. Reinhart, *Interface Strategies*, OTS Working Paper TL-95-002 (Utrecht University, 1995).
24 See Hendriks, "Type Change." Cooper stores are another option; see Cooper, *Quantification*.
25 In some of the steps of this derivation, we have collapsed more than one rule application. If necessary, fill in the intermediate steps.
In the first step of our calculation, we presuppose that $[\![\alpha]\!] = [\![\alpha]\!]^{\varnothing}$. To make sense of this, remember that the empty set $\varnothing$ is a special case of a partial function from $\mathbb{N}$ to D, hence qualifies as an assignment in the technical sense of our definition from chapter 5. $\varnothing$ is the (extremely partial) assignment which has no variable at all in its domain. It is generally the case that a tree will have a well-defined semantic value under the assignment $\varnothing$ just in case it has the same semantic value under all assignments. So "$[\![\phi]\!]$" can always be replaced by "$[\![\phi]\!]^{\varnothing}$".
26 We assume here that **one** has the following entry: $[\![\textbf{one}]\!] = \lambda f \in D_{<e,t>} \,.\, [\lambda g \in D_{<e,t>} \,.$ there is at least one $x \in D$ such that $f(x) = g(x) = 1]$.
27 R. May, "The Grammar of Quantification" (Ph.D. dissertation, MIT, 1977).
28 We can leave open for the moment whether it is a grammatical (though implausible) reading, or not even that. See ch. 8.
29 May dubbed this the "inversely linked" reading, because the scope relations here are in some sense the "inverse" of the surface hierarchy: the superficially more embedded quantifier has the widest scope.
30 VP deletion will be discussed in chapter 9.
31 The account of Antecedent-Contained Deletion we just sketched presupposes that information about the presence of a suitable antecedent VP at LF is available during the derivation from SS to PF. Apparent "communication" between LF and PF is a currently much debated issue that comes up independently in the area of focus interpretation and pronunciation. As for Antecedent-Contained Deletion, some authors assume that DPs are not truly in their VP-internal positions at PF any more. They have already moved up the tree, and occupy a position within a functional projection. See Hornstein, *Logical Form*, for example.
32 This argument is originally due to Ivan Sag, *Deletion and Logical Form*, MIT (Ph.D. dissertation, MIT, 1976; published by Garland, New York, 1980), and

repeated in May, *Logical Form*. But see Jacobson, "Antecedent-Contained Deletion," for an analysis of VP deletion that does maintain the *in situ* interpretation of quantifiers.

33 More sophisticated versions of this analysis have been extended to all pronouns (not just reflexives). Some of them are serious competitors for the quantifier movement solution we adopt in this book, and thus potentially invalidate the objection to *in situ* approaches we have raised here. See especially the works of Pauline Jacobson, cited in n. 5.