Quantifiers*

Michael Glanzberg

University of California, Davis

Quantified terms are terms of generality. They are also provide some of our prime examples of the phenomenon of scope. The distinction between singular and general terms, as well as the ways that general terms enter into scope relations, are certainly fundamental to our understanding of language. Yet when we turn to natural language, we encounter a huge and apparently messy collection of general terms; not just *every* and *some*, but *most*, *few*, *between five and ten*, and many others. Natural-language sentences also display a complex range of scope phenomena; unlike first-order logic, which clearly and simply demarcates scope in its notation.

In spite of all this complexity, the study of quantification in natural language has made remarkable progress. Starting with a seminal trio of papers from the early 1980s, Barwise and Cooper (1981), Higginbotham and May (1981), and Keenan and Stavi (1986), quantification in natural language has been investigated extensively by philosophers, logicians, and linguists. The result has been an elegant and far-reaching theory. This chapter will present a survey of some of the important components of this theory. Section I will present the core of the theory of generalized quantifiers. This theory explores the range of expressions of generality in natural language, and studies some of their logical properties. Section II will turn to issues of how quantifiers enter into scope relations. Here there is less unanimity than in the theory of generalized quantifiers. Two basic approaches, representative of the main theories in the literature, will be sketched and compared. Finally, section III will turn briefly to the general question of what a quantifier is.

*Thanks to the members of the Syntax Project at the University of Toronto, and to Ernie Lepore, for comments on earlier drafts.

I Generality in Natural Language

The first of our topics is the notion of quantified expressions as expressions of generality. We have already observed that natural languages present us with a wide range of such expressions. We thus confront a number of questions, both foundational and descriptive: what are the semantics of expressions of generality, what sorts of basic semantic properties do they have, and what expressions of generality appear in natural language?

One of the accomplishments of research over the last 25 years is to give interesting answers to these questions. Though many problems remain open, a great deal about the basic semantic properties of natural-language quantifiers is known. This is encapsulated in what is often called *generalized quantifier theory*. This section will be devoted to the core of this theory. It should be noted at the outset that generalized quantifier theory is a large and well-developed topic, and there is too much in it to cover in any exhaustive way. There are, fortunately, two very good more specialized surveys to which interested readers may turn for more details and more references: Keenan and Westerståhl (1997) and Westerståhl (1989).

I.1 Denotations for Quantifier Expressions?

Consider two sentences:

(1) a. Bill weighs 180 lbs.

b. Everything weighs 180 lbs.

The beginning of a story about the semantics of (1a) is easy to see. The subject expression *Bill* picks out an individual, and the predicate *weighs 180 lbs.* predicates some property of that individual. The sentence is true if and only if the individual has the property.

But what of (1b)? The property of weighing 180 lbs. remains the same, but what is it being predicated of? Is there some *denotation* for the expression *everything*? More generally, we might ask what contribution *everything* makes to the truth conditions of (1b). Can we identify some entity, the *semantic value* of *everything*, which captures this contribution? (I shall use the terms *denotation* and *semantic value* interchangeably.)

It is fairly obvious that no individual can be the denotation of an expression of generality like *everything*. That would be a strange individual indeed, both some particular individual and at the same time 'everything'. But it might seem appealing to make the semantic value of such an expression something like a property. For instance, we might propose that the contribution of *everything* to (1b) is the property of being among everything.

There are a number of problems with this idea. One might raise metaphysical concerns about properties, or about whether properties can be the denotations of terms the way individuals can be the denotations of names (hence, the more neutral term *semantic value* might be more apt). But there are also some more immediate semantic problems which make this proposal fail. First, it leaves mysterious how the truth conditions of a sentence like (1b) could be determined. If both the subject *everything* and the predicate *weighs 180 lbs.* contribute properties, we lack an account of how to combine them to determine a truth value.

We might attempt to solve this problem, but it looks like we would simply get the wrong results for some cases. Here is an idea: suppose we say a sentence like (1b) is true if the things which fall under the property given by *everything* also fall under the property given by *weighs 180 lbs.* This seems to work for (1b). But the same idea would get the wrong answers for:

(2) Nothing weighs 180 lbs.

Presumably our idea would associate with *nothing* the property of being among nothing, i.e. an empty property. But then everything which falls under this property also bears the property of weighing 180 lbs., vacuously. So, our idea predict that (2) is true. This is just wrong. (For more extensive arguments along these lines, see Heim and Kratzer (1998).)

The solution is to treat the semantic values of expressions of generality not as properties of individuals, but as *properties of properties*, i.e. as *second-level properties*. This idea essentially comes from Frege (Frege, 1879, 1891, 1893). (Frege himself would have insisted that quantifiers are what he called *second-level concepts*, but we do not need to worry about Frege's particular notion of concept to make the basic point.) Let us first think of this in the more familiar terms of first-order logic. A sentence like $\forall x F(x)$, according to the Fregean view, tells us that the property of being F is such that everything falls under it. Thus, the contribution of $\forall$ is the second-level property which holds first-level properties under which every individual falls.

We can think of *everything* in (1b) as working the same way. It contributes the second-level property of being a property under which everything falls. The sentence says that the property of weighing 180 lbs. has this feature, which is false. Likewise, we get the right answer for (2). In (2), *nothing* contributes the second-level property of being a property under which nothing falls. The sentence says that the property of weighing 180 lbs. has this feature, which is false.

I.2 Generalized Quantifiers

For our purposes, we do not need to worry in any serious way about the nature of properties. They apply to individuals, and in doing so make a certain kind of contribution to the truth or falsehood of a sentence. To make this vivid, we can represent them by *sets*. This is to ignore the intensional aspects of properties, but they will not be at issue here. For our purposes, treating properties as sets is a harmless theoretical simplification.[1]

If we represent properties by sets, then second-level properties are sets of sets. This allows us to put the fundamental observation of section I.1 as a thesis about the semantic values of quantifier expressions:

(3) The semantic values of quantifier expressions are sets of sets.

This thesis, though it will be refined in some ways as we progress, is the core of the theory of quantifiers we will develop.

We need a little more detail to make this thesis precise. We will generally start with some background *universe of discourse* M. The semantic value of a predicate is then thought of as a *subset* of M (which we think of a representing something like a property). A quantified expression like *everything* or $\forall$ has as semantic value a set of subsets of M. *Everything* has as value the set of

[1] I am generally assuming that semantic values are sets, and that they are *extensional*. Much of what follows is independent of these assumptions, though there are a number of applications in the literature for which it is crucial that predicate semantic values have *cardinalities*.

subsets of M which include all of M, i.e. are the entire universe. Likewise *something* or $\exists$ has as value the set of subsets of M which are non-empty.

Once we see quantifiers as sets of sets, we can quickly observe that being non-empty and being the entire universe are merely two among many. Set theory provides many such sets of sets, and some of them prove of interest in logic. So, for instance, relative to a fixed universe M, we can define:

(4) a. $(\mathbf{Q}_R)_M = \{X \subseteq M \mid |X| > |M \setminus X|\}$

b. $(\mathbf{Q}_\alpha)_M = \{X \subseteq M \mid |X| \geq \aleph_\alpha\}$

($|X|$ is the cardinality of a set X. In many cases, where we have some set which is to be thought of as the semantic value of an expression, I will put the set in **bold**; so $(\mathbf{Q}_R)_M$ interprets Q_R relative to a universe M. As I mentioned above, I shall use 'semantic value' and 'denotation' interchangeably.)

Sets of sets like those defined in (4) are often called *generalized quantifiers* or *Mostowski quantifiers*, in honor of their first extensive study by Mostowski (1957). Mostowski quantifiers can be added to the usual first-order logic. $Q_\alpha xF(x)$ says that the extension of F has cardinality $\geq \aleph_\alpha$. $(\mathbf{Q}_R)_M$ is the Rescher quantifier (Rescher, 1962). For a finite universe M, $(Q_R)xF(x)$ says that the extension of F is more than half the size of M. Mostowski quantifiers thus allow us to supplement our usual first-order logic to express more than $\forall$ and $\exists$. The basic idea of quantifier expressions denoting sets of sets allows us to also express such properties as being of a certain cardinality, and being more than half.

One fairly technical distinction needs to be made before we close this subsection. We defined Mostowski quantifiers for a fixed universe M. These are what are usually called *local* generalized quantifiers. *Global* generalized quantifiers are simply functions from sets M to local generalized quantifiers on M. So, for instance, for each M, $(\mathbf{Q}_R)_M$ is the local Rescher quantifier on M. $\mathbf{Q}_R$, the global Rescher quantifier, is the function which takes M to $(\mathbf{Q}_R)_M$. For the most part, we will ignore this rather technical distinction, but it will matter in a few important places.

I.3 Generalized Quantifiers in Natural Language

Though the kind of generalization of ∀ and ∃ given by Mostowski quantifiers is a major step, it is not enough to accurately explain natural language quantifiers. For instance, in a way the Rescher quantifier Q_R expresses *most*, but not the way natural language does. Consider:

(5) a. Most students attended the party.

b. Most birds fly.

c. Most people have ten fingers.

These do not do what Q_R does. Q_R compares the size of a predicate extension with the size of the entire universe. These, on the other hand, compare the size of one subset of the universe with another. The first, for instance, says that the set of students who came to the party is larger than the set of students who did not come to the party.

In (5), we see quantifiers comparing one set to another, relating the denotation of one predicate with the denotation of a second predicate. We see a fundamentally *binary* structure. This binary structure is quite widespread in natural language. We see, for instance:

(6) a. Few students attended the party.

b. Both students attended the party.

c. Enough students attended the party.

Each of these involves an expression of generality (*few, both, enough*) relating two predicates (*students, attended the party*).

We also see the same binary pattern of expression of generality relating two predicates in many more constructions, as as:

(7) a. Between five and ten students attended the party.

b. At least ten students attended the party.

c. All but five students attended the party.

d. More male than female students attended the party.

e. John's mother attended the party.

f. More of John's than Mary's friends attended the party.

In fact, though we treated *everything* and*something* as like the unary ∀ and ∃ in section I.1, the English *every* and *some* really display this binary structure as well:

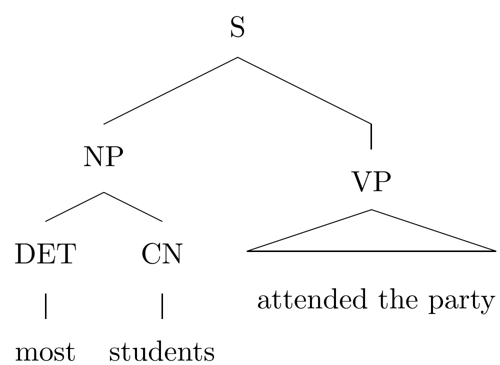
(8) a. Every student attended the party.

b. Some student attended the party.

(These examples are modeled on the much more extensive list in Keenan and Stavi (1986).)

The binary pattern in natural-language expressions of generality is no accident. It reflects a fundamental feature of the syntax of natural languages. Simplifying somewhat, we can observe that sentences break down into combinations of noun phrases (NPs) and verb phrases (VPs). Noun phrases also break down, into combinations of *determiners* (DETs) and common nouns (CNs) (or more complex construction with adjectival modifiers like *small brown dog*). Quantifier expressions of the sorts we see in (6–8) occupy the determiner positions in subject noun phrases. The basic structure we see in all those examples follows the pattern:

(9)



This structure is not only a matter of syntax. It is *semantically significant*. Examples like (5) show that we need to see the CN position as semantically significant to capture the meaning of expressions like *most*.[2]

[2]There are a number of syntactic issues I am putting aside here. See any current syntax text, or the handbook discussions of Bernstein (2001) and Longobardi (2001). For some interesting cross-linguistic work, see Baker (2003), Matthewson (2001), and the papers in Bach *et al.* (1995).

To do this, we need a modest extension of the idea of a Mostowskian generalized quantifier. That idea took Frege's suggestion that quantifiers are second-level properties and formalized it as the idea that quantifiers are sets of sets. To capture the binary structure of natural-language expressions of generality, we need to work not with sets of sets, but with *relations* between sets. In (9), we see that the semantic value of the determiner *most* should relate the value of the CN *students* and the VP *attended the party*. As we are assuming CNs like *students* and VPs like *attended the party* have sets as their semantic values, the determiner *most* must have a relation between sets as its semantic value. This is our next thesis:

(10) The semantic values of many quantifier expressions (determiners) in natural languages are relations between sets.

This is often called the *relational theory of determiner denotations*.

The relational theory of determiner denotations allows us to explicitly define a wide range of natural-language quantifiers. As with Mostowski quantifiers, we we start with a universe M. We now define relations between subsets of M. For instance, for each M and $X, Y \subseteq M$:

(11) a. $\mathbf{every}_M(X, Y) \longleftrightarrow X \subseteq Y$

b. $\mathbf{most}_M(X, Y) \longleftrightarrow |X \cap Y| > |X \setminus Y|$

c. $\mathbf{neither}_M(X, Y) \longleftrightarrow |X| = 2 \wedge X \cap Y = \emptyset$

d. $\mathbf{at\ least\ 10}_M(X, Y) \longleftrightarrow |X \cap Y| \geq 10$

Similar definitions can be given for other quantifiers, including those in (6) and (7).

It will be useful to have some notation to keep track of whether we are talking about relational quantifiers like those in (11), or unary ones like those in (4). A Mostowski quantifier, which takes one set input, is classified as type $\langle 1 \rangle$. The quantifiers we have just looked at are classified as type $\langle 1, 1 \rangle$, taking two set inputs. The number 1 signifies that each input is a set (so the quantifier is *monadic*). As in section I.2, technically we want to distinguish *local* from *global* quantifiers. So our official definitions are:

(12) a. A (local) type $\langle 1, 1 \rangle$ quantifier on M is a relation $\mathbf{Q}_M(X, Y)$ on sets $X, Y \subseteq M$.

b. A (global) type $\langle 1,1\rangle$ quantifier is a function from universes M to local quantifiers $\mathbf{Q}_M$.

As before the difference between local and global quantifiers will matter in a few places, but not many.

We thus see that natural-language *determiners* can be interpreted as type $\langle 1,1\rangle$ quantifiers. Full NPs (combining a determiner with a CN, like *most students*) can be understood as these quantifiers with one argument fixed, which are then type $\langle 1\rangle$ quantifiers.[3]

I.4 Restricted Quantifiers

Type $\langle 1,1\rangle$ quantifiers appear to be *restricted* quantifiers. Whereas $\forall$ and $\exists$, and other type $\langle 1\rangle$ quantifiers, range over the entire universe, a quantifier like **most** seems to range over its first input, corresponding to the CN position in a noun phrase. In (6), for instance, we think of **most** as ranging over the set **students**. Intuitively, this means that the truth or falsehood of *Most students attended the party* should depend only on what happens in the set **students**, and nothing else about the universe of discourse.

It does turn out that natural language quantifiers display important features of restricted quantification. However, the reason is more complex than the mere presence of an extra input position corresponding to a CN. It is entirely possible to define type $\langle 1,1\rangle$ quantifiers which are not restricted. For instance:

(13) $\mathbf{more}_M{}^{\langle 1,1\rangle}(X,Y) \longleftrightarrow |X| > |Y|$

This does not behave as if its domain is restricted to X, in cases where Y and X do not overlap. So, $\mathbf{more}_M{}^{\langle 1,1\rangle}(\mathbf{animals},\mathbf{humans})$ holds if there are more animals than humans, which has as much to do with the number of non-humans as humans. (This is a perfectly good type $\langle 1,1\rangle$ quantifier, but as we will see in a moment, it may not correspond to any natural language expression.)

The core feature which makes natural language quantifiers behave like restricted quantifiers is exhibited by the following pattern:

[3] Terminology varies on whether determiners or full NPs are called 'quantifiers'; for instance, Barwise and Cooper (1981) reserve the term 'quantifier' for NP denotations, i.e. type $\langle 1\rangle$ quantifiers.

(14) a. i. Every student attended the party.

ii. Every student is a student who attended the party.

b. i. Few students attended the party.

ii. Few students are students who attended the party.

c. i. Most students attended the party.

ii. Most students are students who attended the party.

In each of these, (i) and (ii) are equivalent. The corresponding feature for $\mathbf{more}_M{}^{\langle 1,1\rangle}$ would be $|X| > |Y| \longleftrightarrow |X| > |X \cap Y|$, which is easily falsified.

The pattern we see in (14) but not in (13) is called *conservativity*:[4]

(15) (CONS) For each $X, Y \subseteq M$, $\mathbf{Q}_M(X,Y) \longleftrightarrow \mathbf{Q}_M(X, X \cap Y)$.

Conservativity expresses the idea of *restrictedness*. For instance, in (14c), it tells us that the truth of *Most students attended the party* depends only on the member of the set **students**.

One of the striking facts about natural languages, observed in Barwise and Cooper (1981) and Keenan and Stavi (1986), is that all natural-language determiner denotations satisfy CONS. It appears that all natural-language quantification is restricted quantification. This is not a conceptual or a logical matter. Examples like (13) clearly violate CONS; hence, there are perfectly intelligible non-conservative quantifiers. Rather, it appears to be an empirical fact about human languages that though logically speaking they could have non-conservative determiner denotations, they do not. We thus have a proposed *linguistic universal*: a non-trivial empirical restrictions on possible natural languages.

As an empirical claim, one of the substantial issues about conservativity is whether it really does hold universally. Much of the discussion has focused on a number of potential counter-examples. Some of them remain controversial, but the consensus in the literature is that the universal holds. Let me give a couple of examples. Why is $\mathbf{more}_M{}^{\langle 1,1\rangle}$ not a counter-example? Because this quantifier does not appear to be the denotation of a natural-language determiner. It might have

[4] This same property was called the 'lives on' property by Barwise and Cooper (1981) and 'intersectivity' by Higginbotham and May (1981). I believe the terminology 'conservativity' is due to Keenan and Stavi (1986).

seemed to be the denotation of *more*, but this is not so. The determiner *more* appears to be a *two-place* determiner, figuring in constructions like:

(16) More students than professors attended the party.

More than is conservative. (Quantifiers taking more than two arguments have been investigated by Beghelli (1994) and Keenan and Moss (1984) (see Keenan and Westerståhl (1997) for additional discussion).

Another much-discussed case is *only*. It may appear to be an easy example of the failure of conservativity. Consider:

(17) Only dogs bark.

A natural reading of this sentence makes it true if and only if the set of barking things is included in the set of dogs. This suggests a highly simplified semantics for *only*:

(18) $\mathbf{only}_M(X, Y) \longleftrightarrow Y \subseteq X$

This is simplified in many ways, but it makes the failure of conservativity vivid. $Y \subseteq X \longleftrightarrow (Y \cap X) \subseteq X$ only holds when $Y \subseteq X$. Hence, any false sentence suffices to show that conservativity fails.

Even so, there is good reason to think that *only* is not a determiner. It appears outside of noun phrases, as in:

(19) John only talked to Susan.

It also appears in places we do not see determiners in English noun phrases:

(20) a. Only the Provost/John talked to Susan.

b. Only between five and ten students came to the party.

We have good reason to think that *only* is not a counter-example to conservativity because it is not a determiner.[5]

[5] For more on *only*, see Herburger (2000) and Rooth (1985, 1996). Related to expressions like *only* are adverbs of quantification, such as *always* and *never*. For discussion of these, see Lewis (1975) and von Fintel (1994).

It appears that all natural-language determiner denotations are conservative, and so the linguistic universal of conservativity holds. A moment ago, I identified conservativity as the reason natural-language quantification appears to be restricted quantification. However, there is a minor complication to this claim, due to differences between local and global quantifiers. (This is one of those points where this technical distinction does matter.)

Conservativity tells us that for a given M and $X, Y \subseteq M$, whether $\mathbf{Q}_M(X, Y)$ holds depends only on X. But this does not guarantee that some change in M which has no effect on X cannot matter. Intuitively, for a restricted quantifier, we expect that it cannot. Intuitively, we think that the only thing that can matter to a restricted quantifier is X, period. This is a property of *global* quantifiers. It tells us that as far as a global restricted quantifier $\mathbf{Q}$ is concerned, $\mathbf{Q}_M(X, Y)$ is just the same as $\mathbf{Q}_X(X, Y)$. This stronger notion of restrictedness is given by the principle:

(21) (UNIV) For each M and $X, Y \subseteq M$, $\mathbf{Q}_M(X, Y) \longleftrightarrow \mathbf{Q}_X(X, X \cap Y)$.

('UNIV' for 'universe-restricting'. Note the subscript on the right-hand side is X.)

The difference between CONS and UNIV is relatively small, but not entirely trivial. It was observed by van Benthem (1983, 1986) that UNIV is equivalent to CONS together with the property EXT (for 'extension'):

(22) (EXT) For each $X, Y \subseteq M \subseteq M'$, $\mathbf{Q}_M(X, Y) \longleftrightarrow \mathbf{Q}_{M'}(X, Y)$.

As observed by Westerståhl (1985*b*, 1989) EXT, expresses the idea that quantifiers do not change their meanings on different domains. This, plus CONS, captures the strong intuitive idea of restrictedness.

A moment ago I glossed the proposed universal of conservativity as one that told us that all natural-language quantification is restricted. In light of our observation that restrictedness is really expressed by UNIV, and that CONS might leave out EXT, we should also ask if it is a linguistic universal that all natural-language determiner denotations satisfy EXT (and hence UNIV) as well.

It appears that they do. As with CONS, logic easily provides us with quantifiers that violate EXT. One example given by Westerståhl (1985*b*) is:

(23) $\mathbf{many}^*{}_M(X, Y) \longleftrightarrow |X \cap Y| > 1/3 \cdot |M|$

As with CONS, there appear to be reasons to reject this as a genuine counter-example, are there appear to be reasons to deny that **many*** is the denotation of a natural-language determiner. One reason is that *many* appears to be context-dependent, in that what counts as many is heavily influenced by context. Depending on how this sort of context-dependence is handled, it may be argued that *many* has a very different sort of meaning than **many***. If it does, we have no reason to think that *many* violates EXT or CONS. Of course, we still need to see how to interpret *many* properly. This remains a controversial issue, and I shall not pursue it in any more detail. See Westerståhl (1985*b*) for extensive discussion.[6]

Though there remains some controversy, especially in cases like *many*, the proposed linguistic universal that all natural-language determiner denotations satisfy CONS and EXT enjoys a great deal of support. It thus appears plausible that all natural-language quantification really is restricted quantification.

In introductory logic classes, we are shown how to build certain restricted quantifiers out of unrestricted ones. *Every student attended the party* can be analyzed as $\forall x(\mathit{student}(x) \longrightarrow \mathit{attended\ the\ party}(x))$. This shows us how to define the $\langle 1, 1\rangle$ restricted quantifier $\mathbf{every}_M(X, Y)$ in terms of the type $\langle 1\rangle$ unrestricted quantifier $\forall$. We have now seen that natural-language determiners denote type $\langle 1, 1\rangle$ quantifiers, and they are restricted quantifiers. This raises the question of whether they can all be defined in terms of type $\langle 1\rangle$ quantifiers.

The answer is they cannot. It is a somewhat technical matter in logic, but it is known that $\mathbf{most}_M$ defined in (11) cannot be defined by any combination of type $\langle 1\rangle$ quantifiers. (There is a modest complication here, involving issues to be discussed in section I.6. I will return to this briefly in section I.8.)

[6]The context-dependence proposed for determiners like *many* is in the meaning of the determiner, not in the restriction of its domain. For discussions of how context restricts the domains of quantifiers, see Cappelen and Lepore (2002), Stanley and Szabó (2000) (with comments by Bach (2000) and Neale (2000)), von Fintel (1994), and Westerståhl (1985*a*). I am skipping over the issue, related to paradoxes, of whether all quantifiers, including such apparently unrestricted ones as *everything*, wind up with some non-trivial contextual domain restriction. This is discussed in Glanzberg (2004) and Williamson (2004).

I.5 How Many Quantifiers Are There?

The simple answer to this question is a lot. If we take a universe M of size n, there are 2^{4^n} type $\langle 1,1\rangle$ (local) quantifiers on M.

Conservativity does more than capture (most of) our intuitive idea of restricted quantification. It also have a significant effect on how many quantifiers there are, and more generally, what the space of quantifiers is like. First of all, there are fewer conservative quantifiers: there are 2^{3^n} type $\langle 1,1\rangle$ quantifiers satisfying CONS on a universe of size n (cf. van Benthem, 1984)).

Perhaps more importantly, the space of conservative quantifiers is much more orderly than its size might make it seem. Conservative quantifiers are all built up in stages. We start with a small collection of basic determiner denotations. In particular, we can start with just $\mathbf{every}_M$ and $\mathbf{some}_M$ (as type $\langle 1,1\rangle$ quantifiers). We then build more quantifiers by a couple of systematic procedures. One is to combine quantifiers we already have by operations of *Boolean combination.* This gives us quantifiers like $\mathbf{all\ or\ some}_M$. We also build more quantifiers by further restricting the domains of quantifiers we already have. This will allow us to build $\mathbf{some\ yellow}_M$. More generally, if we have built $\mathbf{Q}_M(X,Y)$, we may then build $\mathbf{Q}_M(X\cap C,Y)$ for $C\subseteq M$. This amounts to closure under (intersective) adjectival restriction in an NP. Call this closure under *predicate restriction.*

One of the striking features of the space of conservative quantifiers is that it includes *exactly* the quantifiers that we can build this way. This is the *conservativity theorem* due initially to Keenan and Stavi (1986), further investigated by Keenan (1993) and van Benthem (1983, 1986). Let us give it a more precise statement. Let M be a fixed *finite* universe. Call the collection of collection of conservative type $\langle 1,1\rangle$ quantifiers on M by $CONS_M$. Call the collection of quantifiers we build up from our base set $D-GEN$. More formally, $D-GEN$ is the set of quantifiers on M containing $\mathbf{every}_M$ and $\mathbf{some}_M$ and *closed* under Boolean combination and predicate restriction. The conservativity theorem tells us:

(24) $CONS_M = D-GEN_M$

(This is a *local* theorem. The proof carries out different constructions for different size M.)

It is an appealing speculation that this might explain why the linguistic universal of conservativity holds. Natural languages might build up their stock of quantifiers in much the way $D - GEN_M$ is built up. Whether this explanation holds good or not, it does point out that the space of conservative quantifiers is not 'too big'. For any finite universe M and any given quantifier in $CONS_M$, we can follow the proof of the conservativity theorem to build a natural language expression which denotes it (granted, one that can be quite long and syntactically complex). This is the *Finite Effability Theorem* of Keenan and Stavi (1986):

(25) For a finite M, each element of CON_M is expressed by a determiner of English.

Thus, the conservativity property makes for a much more tractable space of determiner denotations, built up in a systematic way which is closely tied to constructions we can carry out in natural language.

I.6 Logicality

We began this section with the idea that quantifiers are expressions of generality. Though we have seen a wide range of determiner denotations which fall within $CONS$ and EXT, we have yet to give any statement of what makes them general. Intuitively, expressions like *most students* do not pick out any particular individual, but pick out 'most of the students, whomever they may be'. This contrasts, for instance, with proper names or demonstratives, which pick out a particular individual, not just whichever individuals meet some conditions.

One way to articulate the notion of generality is that it requires the truth of a sentence to be independent of exactly which individuals are involved in interpreting a given quantifier. This can be captured formally by the constraint of *permutation invariance*:

(26) (PERM) Let π be a permutation of M (i.e. a bijection from M to itself). Then $\mathbf{Q}_M(X, Y) \longleftrightarrow \mathbf{Q}_M(\pi[X], \pi[Y])$.

PERM guarantees that changing the individuals we are talking about does not change the truth of what we are saying, so long as the individuals satisfy the right properties.

Technically speaking, PERM is a local condition. It works with a fixed universe M. A global version can be stated:

(27) (ISOM) For any M and M', if $\iota\colon M \to M'$ is a bijection, then $\mathbf{Q}_M(X,Y) \longleftrightarrow \mathbf{Q}_M{}'(\iota[X],\iota[Y])$.

ISOM states the property of *isomorphism invariance*, which captures the idea of changing the individuals we are talking about, not just within a universe M, but across different universes. The mathematical literature on quantifiers commonly assumes ISOM, and it is built into the definitions of quantifiers in Lindström (1966) and Mostowski (1957).[7]

Though ISOM is the standard condition in the literature, and technically somewhat stronger than PERM, the difference between the two conditions is not that great. Westerståhl (1985*b*, 1989) observed that if we assume EXT, the domain of quantification ceases to matter, and ISOM and PERM are equivalent.

Following van Benthem (1983, 1986), one sometimes sees quantifiers satisfying CONS, EXT, and ISOM called *logical quantifiers*. There is a rich and extensive mathematical theory of the logical quantifiers. For an introduction, see van Benthem (1986) or Westerståhl (1989).

ISOM (or PERM) does appear to capture the idea that quantifiers are *general*, and so not about any objects in particular. It is a further question whether this makes them genuinely *logical constants*, as the label 'logical quantifier' suggests. The idea that some sort of permutation-invariance is a key feature of logical notions has been proposed by Mautner (1946) and Tarski (1986). A vigorous defense of the logicality of ISOM quantifiers is given in Sher (1991).

I.7 Quantifiers and Noun Phrases

We have seen that, noting a few controversial potential exceptions, natural-language determiner denotations satisfy CONS and EXT. Intuitively, we might also want to say that the expressions

[7]The condition is called 'ISOM', as ι induces an isomorphism between the structures $\mathfrak{M} = \langle M, X, Y\rangle$ and $\mathfrak{M}' = \langle M', \iota[X], \iota[Y]\rangle$. In essence, as Lindström (1966) observed, a type $\langle 1, 1\rangle$ generalized quantifier is a class of structures of the form $\langle M, X, Y\rangle$; if it satisfies ISOM, we have a class of structures closed under isomorphism.

we identify as *quantifiers* also satisfy ISOM (or PERM). It is a tempting generalization that natural-language quantifiers are the *logical quantifiers*.

However, there are some clear cases treated by generalized quantifier theory which do not satisfy ISOM, and so are not logical quantifiers. We have already seen one. The possessive construction *John's* in (7) violates ISOM. So do some syntactically complex constructions like *every ___ except John* when treated as determiners.

Perhaps a more pressing case is that of proper names. We *can* treat proper names as generalized quantifiers. Suppose *John* denotes an individual **j**. We can build a type $\langle 1 \rangle$ generalized quantifier to interpret the NP *John* following Montague (1973). Let $\mathbf{John}_M = \{X \subseteq M \mid \mathbf{j} \in X\}$. This is a quantifier violating ISOM.

There are two ways to respond to these cases. One is to give up on ISOM as a feature of quantifiers in natural language. This leaves the generalization that determiners denote type $\langle 1, 1 \rangle$ quantifiers satisfying CONS and EXT, but not necessarily ISOM. These determiners build type $\langle 1 \rangle$ quantifiers satisfying CONS and EXT when combined with a CN denotation, so we might make the further generalization that all NPs denote type $\langle 1 \rangle$ generalized quantifiers, once we have given up on ISOM.

Another response is to keep the generalization that all natural-language quantifiers satisfy ISOM, and attempt to explain away the apparent violations. (If we count constructions like *every ___ except John* as determiners, we should specify only quantifiers denoted by syntactically simple determiners.) In the type $\langle 1 \rangle$ case, we can easily observe that though it is possible to treat *John* as a generalize quantifier, it can also be treated as simply denoting an individual. There are good reasons to take this simpler route (cf. Partee, 1986). (Indeed, much of the philosophical literature on names would not even consider any other option!) Thus, an apparently non-ISOM quantifier in natural language may not be a quantifier at all. Likewise, in the type $\langle 1, 1 \rangle$ case, we might find analyses of possessive constructions which do not treat them as syntactically on par with simple determiners, or do not treat them as determiners at all. (See Barker (1995) for an extensive discussion of the syntax and semantics of possessives.)

If we offer this second response, we can defend a strong hypothesis: quantifiers in natural language are the denotations of determiners (or perhaps the syntactically simple determiners), and they are logical generalized quantifiers satisfying CONS, EXT, and ISOM. In light of non-ISOM examples like proper names, this hypothesis predicts an important difference between genuine quantified noun phrases, built up out of determiners denoting ISOM quantifiers, and other noun phrases like proper names or possessive constructions.

If this strong hypothesis is correct, there are real differences between quantified NPs and other NPs. We could provide further support for the hypothesis by finding ways in which quantified NPs behave differently from other NPs. The more differences we can see in the ways quantified and non-quantified NPs behave, the more reason we have to accept an analysis which makes them fundamentally different.

In fact, there are ways in which quantified and non-quantified NPs behave differently. One way is brought out by what are called *weak crossover* cases. Compare:

(28) a. * His$_i$ mother loves every boy$_i$.

b. His$_i$ mother loves Mary's Brother$_i$.

c. His$_i$ mother loves John$_i$.

(The subscripts here are to indicate that the desired reading has *his* bound by or coreferring with the subsequent expression it is co-indexed with.) A number of authors have noted that we get unacceptability in weak crossover environments with ISOM quantified noun phrases, but not with non-ISOM or non-quantified ones. We thus have a difference in behavior between quantified and non-quantified NPs, and so he have evidence for the strong hypothesis (cf. Higginbotham and May, 1981; Larson and Segal, 1995; Lasnik and Stowell, 1991). (Readers of the logic literature should be aware that regardless of their status in natural language, most logicians take generalized quantifiers to satisfy ISOM by definition.)

I.8 Glimpses Beyond

We now have seen the beginnings of generalized quantifier theory, but only the beginnings. The surveys of Keenan and Westerståhl (1997) and Westerståhl (1989) discuss a number of extensions of the theory, and applications of generalized quantifier theory in linguistics.

Among the results they discuss is one that shows that the quantifier **most** defined in (11) cannot be defined by any combination of (ISOM) type $\langle 1 \rangle$ quantifiers. This shows that we really do need at least type $\langle 1, 1 \rangle$ quantifiers (cf. Väänänen, 1997). They also investigate the delicate issue of whether we need to go beyond $\langle 1, 1 \rangle$. We saw that *more* should be interpreted as taking *three* arguments. Whether we will also need to consider what are called *polyadic* quantifiers, which take *relations* rather than sets as inputs, remains an active area of research (cf. Hella *et al.*, 1996; Higginbotham and May, 1981; Keenan, 1992; May, 1989; Moltmann, 1996; van Benthem, 1989; Westerståhl, 1994).

II Quantification and Scope

The relational theory of determiner denotations, which we examined all too briefly in section I, explains some of the important properties of the semantic values of determiners. But it does not do very much to explain how determiners interact with the rest of semantics. As an example of where quantifiers fit into semantic theory, I shall present some ideas about how quantifiers take scope in natural language. In an example like *Every student likes some professor*, for instance, it is clear that the sentence can be read as having *every student* take scope over *some professor*, or vice versa. The theory of generalized quantifiers by itself does not explain how this can happen. Indeed, as we will see, the theory of generalized quantifies by itself already runs into trouble explaining how the parts of a sentence like this can combine. Seeing how they can, and how they can in ways that allow for multiple scope readings, will show us something about how quantifiers work.

Perhaps more so than the theory of generalized quantifiers, this area remains controversial. There are a number of good textbook presentations of the basic material, including Heim and

Kratzer (1998) and Larson and Segal (1995). (I follow the former quite closely here.) But there is also some significant disagreements in the literature. To illustrate this disagreement, I shall discuss two representative examples of approaches to quantifier scope. I shall need some machinery to do so, which is built up in sections II.1–II.4. The actual discussion of scope is in section II.5.

II.1 Quantifiers and Semantic Types

The account of generalized quantifiers as relations between sets pays no attention to the order in which a quantifier's arguments are 'processed'. For studying the properties of determiners, this has proved a useful idealization. But if we are to consider how quantifiers interact with the rest of semantics, we will need to be more careful about how they combine with other semantic values.

A glance at the sentence structure in (9) tells us that the compositional semantics of determiners should first have the determiner's value combine with the value of the CN, resulting in an NP semantic value. It is the NP value which combines with the VP value to determine the value of the sentence. We should first build the value of *most students*, and then see how that combines with the value of *attended the party*.

To capture this, it will be useful to reformulate our description of a quantifier somewhat. Generally, we will turn out attention from sets, and sets of sets, to *functions*. Recall that a set of elements of M can be thought of as a function from M to *truth values*. The members of the set are the elements on which the function returns the value true. A set of sets (i.e. a type $\langle 1 \rangle$ quantifier) can be thought of as a function which takes functions (giving sets) as inputs and outputs truth values.

It will be useful to have some notation to keep track of the inputs and outputs of functions. One way to do this is to use *type theory*. Type theory is a highly general theory of functions. In order to try to avoid confusion between types in the sense of quantifier types and this type theory, I shall sometimes call the latter *semantic type theory*.

Semantic type theory starts with two basic types: t is the type (set) of truth values, which we may take to have two elements $\top$ and $\bot$; e is the type (set) of individuals, which we may take to be some fixed universe M. The theory then builds up functions out of these. The type (e, t) is the type of functions from individuals to truth values, i.e. it is a notation for $\wp(M)$, the set of subsets of M. A quantifier-type $\langle 1 \rangle$ quantifier (a set of sets) is a function of type $((e,t),t)$, taking as input functions representing sets, and having truth values as outputs. Generally, for any two types a and b, (a,b) is the type of functions from a to b.[8]

Using the apparatus of semantic types, we can put our definition of quantifier-type $\langle 1,1 \rangle$ quantifiers in terms of functions. Definition (12) makes a type $\langle 1,1 \rangle$ quantifier $\mathbf{Q}_M$ a relation between sets. We might think of this as a function on two arguments X and Y. But our semantic type theory only has functions of one argument. To handle functions of multiple arguments, we simply process the arguments in sequence.[9] We first input X, and output the function $\mathbf{Q}_M(X)$. This is a function from Y to truth values, which has output $\top$ iff $\mathbf{Q}_M(X,Y)$ is true. Our notation helps make this clear. A quantifier-type $\langle 1,1 \rangle$ quantifier is of semantic-type $((e,t),((e,t),t))$. It takes as input a set (element of type (e,t)), and returns a function of type $((e,t,),t)$. This is a function which takes another set as input, and outputs a truth value. (From now on, we will work with a fixed universe M, giving type e, and only consider local quantifiers on M.)

Semantic type theory gives us a useful notation for keeping track of complex functions. It also gives us a useful way to keep track of the kinds (the types) of semantic values various expressions should have. We will continue with our assumption that the values of VPs and CNs are sets of individuals, i.e. are of type (e,t). We will also continue with the extensional perspective, which gives sentences semantic values of type t. (This is of course, an idealization.) We will also assume that non-quantified NPs are of type e, in accord with the strong hypothesis of section I.7 supposed. As we have just seen, quantified NPs have semantic values of type $((e,t),t)$. Determiners have values of type $((e,t),((e,t),t))$. (I shall often abuse notation and say that e.g. determiners *are* of type $((e,t),((e,t),t))$.)

This analysis of determiner denotations is essentially the relational one of section I, except that it takes into account the order in which inputs are processed. For the most part, I shall treat

[8] I am writing semantic types with round brackets, such as (a,b). Much of the literature writes semantic types with angle brackets, but these are already being used for quantifier types.

[9] This is what is sometimes called 'Currying' a binary relation, in honor of the logician Haskell B. Curry.

semantic type theory simply as a notational device. Most of what we will do with semantic type theory can be done without it as well. (There is one point at which this will not be the case, in section II.4.)

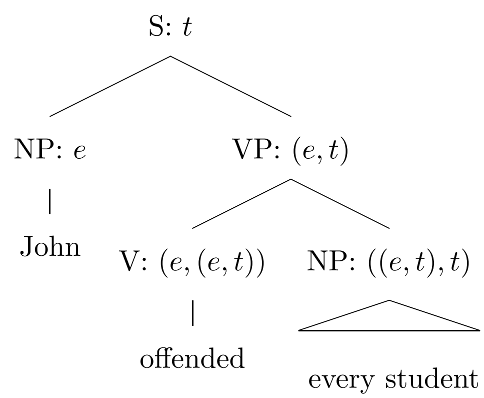
II.2 Quantifiers in Object Position

Our semantic analysis starts with the idea that determiners are of type $((e,t),((e,t),t))$, CNs are of type (e,t), and VPs are of type (e,t). Describing these semantic values in terms of semantic types also allows us to explain how they combine according to the structure of a sentence, to yield the semantic value of the sentence (of type t). For instance, in sentences like (9), the DET value takes as argument the CN value, and yields a quantified NP value, of type $((e,t),t)$. This takes as input the VP value, and the result is of type t, i.e. a truth value, as desired.

If we look at little more widely, however, we run into problems of composition. Transitive verbs with quantifiers in object position provide one sort of problem. A transitive verb will be of type $(e,(e,t))$, taking two type e arguments (in sequence). But consider an example like:

(29) a. John offended every student.

b.



The entries for the VP simply do not match. *Offended* is of type $(e,(e,t))$. But the quantified NP *every student* is of type $((e,t),t)$. Neither can be the argument for the other. If, as the basic type-theoretic perspective supposes, semantic composition is composition of function and argument, we have no way to combine them. The notation of semantic types makes this problem vivid, but it is not special to semantic type theory. One way or another, the quantified NP *every student* should denote something like a second-level property, set of sets, or elements of type $((e,t),t)$, while the V offended should denote a two-place first-level property, or element of type $(e,(e,t))$. The problem is we have no way to combine these denotations.

The theory of generalized quantifiers, as a theory of determiner denotations, does not help us to solve this problem.[10] Instead, some more apparatus is needed, either in the semantics or in the syntax. There are two basic approaches to solving this problem. One involves significant claims about *logical form*. The other makes some corresponding claims about *semantic types*.

II.3 Logical Form and Variable Binding

One approach to the problem of quantifiers in object position, perhaps the dominant one, is to posit underlying logical forms for sentences which are in some ways closer to the ones used in the standard formalisms of logic.

The problem of quantifiers in object position does not arise in first-order logic. It does not because Frege in effect solved it. In first-order logic, we would represent (29) as:

(30) $\forall x(\mathit{student}(x) \longrightarrow \mathit{offended}(\mathit{John}, x)$.

The solution implicit here has nothing to do with unrestricted versus restricted quantifiers. We could do just as well if we could produce a structure that looks something like:

(31) Every student_x (John offended x).

What solves the problem is the apparatus of quantifiers and variables. We put a variable in the predicate, and *bind* it with the quantifier. In terms of the structure of (29), the idea is to replace the quantified NP *every student* in the VP with a variable of type e. This variable would function as the argument of the type $(e,(e,t))$ verb, and also be *bound* by the quantifier from outside the VP. This is in effect what we see in (31).

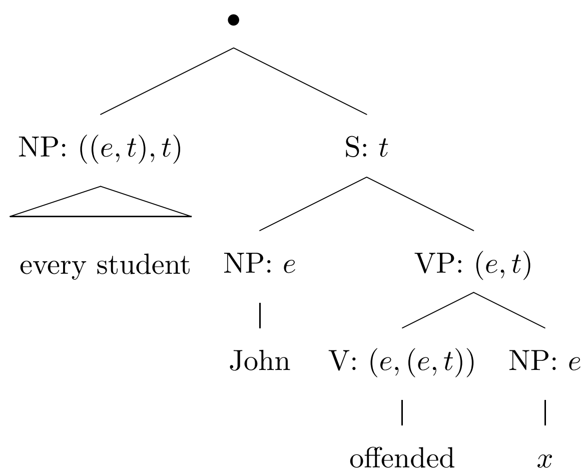
To explain how this can work in our framework of semantic types, we need to look a little further at how variables work. Let x be a variable of type e. If x is free, we can treat it like the

[10]There is one drastic generalized quantifier theory option we might take, which would be to appeal to polyadic quantifiers of the sort hinted at in section I.8, following Keenan (1992).

pronoun *it*. It has its value fixed by context, but otherwise acts like a referring expression. It is like any other expression of type e, except for needing context to fix its value.

Because of this, an overly simple implementation of the idea in (31) does not work. We might propose simply to replace *every student* in the VP with a variable x of type e, and write the quantified NP *every student* all the way to the left. This would give something like:

(32)



But we still have a mismatch of types, and the structure cannot be interpreted. The variable x is simply an expression of type e. It does combine with the V *offended*. Running up the tree, all looks well up to the S node, which is of type t as it should be. But then we have a problem. This cannot combine with the NP node of type $((e, t), t)$.

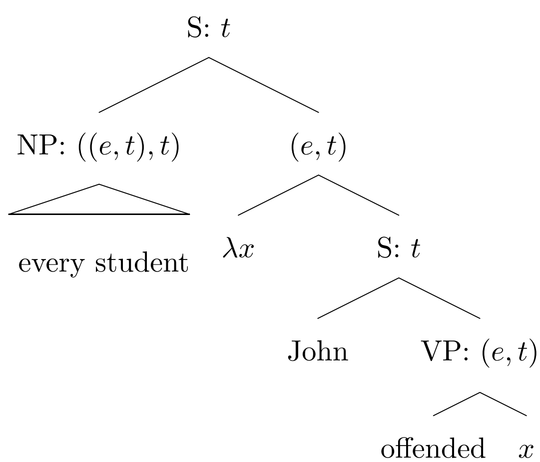
What we left out of this overly simple implementation is what is supposed to be shown by the subscript in *every student*$_x$. To get the structure we had in mind in (31), we need to cash out the idea that *every student*$_x$ really *binds* x in the VP. Insofar as x is just another expression of type e, we have no explanation of how it might be bound by a quantifier. Writing the subscript on the quantifier is just notation: we need to explain the idea this notation is supposed to show us. We need some explanation of how binding works.

In the type-theoretic setting, binding is done by the apparatus of λ-*abstraction*. λ is the operation that creates *functions* in the framework of semantic types. Consider the semantic value **offended** x of the S node in (32). This is of type t because x is treated as anther type e expression, which contributed its value to **offended** x and then is done. We want it not to contribute its value there, but rather to mark an *input* place, resulting is a function which takes an input into the x place, and gives an appropriate output. This is the function $\lambda x.$**John offended** x. This function an element of type (e,t), i.e. a function which takes a type e input in the x position, and outputs a type t value.

λ *binds* a variable position, resulting in a function. Building a function by binding a variable with a λ is usually called λ-*abstraction*. (For more discussion of the mathematics of λs, see Gamut (1991) or Hindley and Seldin (1986).) In full generality, if β is an element of type b and y is a *variable* of type a, then $\lambda y.\beta$ is an element of type (a,b). λ-abstraction allows us to build functions, and so allows us to construct elements of complex types like (a,b).

To get something that works like (31), we need to add λ-abstraction. With it, we can resolve the mismatch between types we see in (29) along the following lines:

(33)



Adding the variable in VP produces an element **John offended** x of type t. λ-abstraction then yields the desired element $\lambda x.$**John offended** x of type (e,t). This can now properly combine with the denotation of the quantified noun phrase.[11]

The use of λ-abstraction in (33) explains what we intuitively represented by the subscript x on *every student*$_x$ in (31). We wanted to make clear that the quantified NP *every student* binds the x position. This is explicitly done by the λ-node in (33). More fully, the λ-note binds the x position, in such a way as to make an input for the quantified NP of the right sort.

[11] Technically, we should say that we add syntactic elements which are interpreted as variables and λs. See Büring (2004) and Heim and Kratzer (1998) for more discussion of the syntax and semantics of these particular structures.

The role of λ-abstraction highlights a point about generalized quantifier theory. Generalized quantifier theory as discussed in section I is not a theory of variable binding. Describing relations between sets does not explain how they figure into variable binding. On the approach I am sketching here, variable binding is done by λ-abstraction, which produces semantic values of appropriate type to be inputs into generalized quantifiers. There are other ways to treat variable binding, but the moral is that generalized quantifier theory does not do this job.

The structure of (33) represents a very rough proposal for the *logical form* of (29); the fully worked out version is that of Heim and Kratzer (1998). This is a significant proposal. The claim is not merely that a formalism like (31) makes the logical dependencies of a sentence clear. Rather, it is that the semantic interpretation of a sentence of natural language is derived from a structure like (33). Thus, logical form is posited as a genuine level of linguistic representation. This is a substantial empirical claim. For more thorough discussion of this notion of logical form, see "Logical Form and LF" in this volume.[12]

It should be noted that once we have forms looking like (31), it is possible to treat binding in a more Tarskian way, without relying on the apparatus of λ-abstraction and types. As I mentioned a moment ago, some account of binding is needed, but there are versions not using λs. One example is the more Davidsonian treatment of Larson and Segal (1995). There are some general methodological questions about the use of higher types in semantics, but the basic idea of treating quantifiers in object position by way of a substantial level of logical form is not particularly sensitive to them.[13]

[12] Following May (1977, 1985), many linguists think of logical form as the result of *movement processes* which move quantifiers from their *in situ* positions to positions more or less like the ones in (33). A survey of ideas about logical form in syntactic theory is given in Huang (1995).

[13] Lepore (1983) and Pietroski (2002) offer critiques of type-based semantics from a broadly Davidsonian viewpoint. Another view of logical form and its role in semantics, more explicitly Davidsonian than the one I am sketching here, is presented in Higginbotham (1985).

II.4 Type Shifting

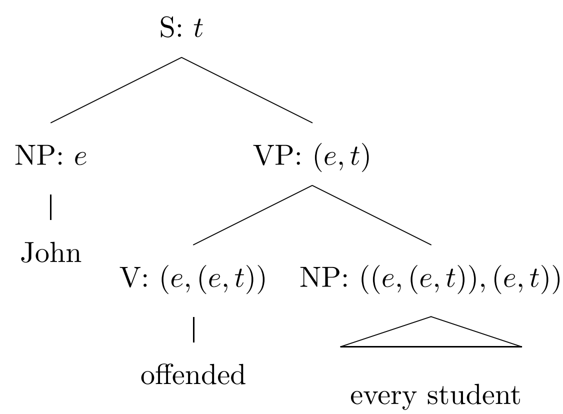
This section is somewhat more technically demanding than the rest of the paper. Readers wanting to avoid long λ-terms might want to skip to section II.5, which can be read without this one.

The approach to resolving the problem of quantifiers in object position I briefly sketched in section II.3 relies on some substantial ideas about logical form. It posits underlying logical forms which look substantially different from the surface forms of sentences, as we saw in (33). There is another way to handle quantifiers in object position, and more generally, to think about issues of binding. Rather than positing a distinct level of *logical form*, the other approach posits more complex modes of *composition* in the semantics.

In this section, I shall very briefly indicate some of the ideas that go into this other approach. This is not to offer any kind of objection to the logical-form-based approach, nor to suggest which approach is right. It is only to show that formally speaking, there are other options.

Suppose we change the type of a quantified NP from $((e,t),t)$ to $((e,(e,t)),(e,t))$. Then we can interpret (29) directly:

(34)



The values of the V and NP compose by the NP value taking the V value as an argument.

How can we change something's type? In this case, the transformation from $((e,t),t)$ to $((e,(e,t)),(e,t))$ is more natural than it might seem. It is an instance of what is known as the *Geach Rule* (cf. Geach, 1972):

(35) $(b,c) \Longrightarrow ((a,b),(a,c))$

This can be thought of as introducing an additional mode of composition, over and above function application. It is essentially *function composition*:

(36) a. i. $(a, b) + (b, c) \Longrightarrow (a, c)$

ii. $\alpha_{(a,b)} + \beta_{(b,c)} \Longrightarrow (\beta \circ \alpha)_{(a,c)}$

b. i. $(e, (e, t)) + ((e, t), t) \Longrightarrow (e, t)$

ii. $\gamma_{(e,(e,t))} + \delta_{((e,t),t)} \Longrightarrow (\delta \circ \gamma)_{(e,t)}$

(36) displays the scheme of function composition, according to which we apply one function α followed by another β. (36b) shows the specific case of (36a) in which we are interested.

(35) adds an operation of function composition by adding a type-shifting operator. It can be spelled out by:

(37) $Geach_a(\beta_{(b,c)}) = (\lambda X_{(a,b)} \lambda y_a[\beta_{(b,c)}(X_{(a,b)}(y_a))])_{((a,b),(a,c))}$

For $\mathbf{Q}_{((e,t),t)}$ of type $((e, t), t)$, $Geach_e(\mathbf{Q}_{((e,t),t)}) = \lambda\nu_{(e,(e,t))}\lambda x_e[\mathbf{Q}_{((e,t),t)}(\nu_{(e,(e,t))}(x_e)))]$ So, for instance ($Geach_e$(**every student**))(**offended**) = **every student** $\circ$ **offended**. This is now of the right type to combine with **John**. Thus, applying the Geach rule resolves the problem of quantifiers in object position.

The operator *Geach* carries out λ-abstraction, as we see in (37). Thus again in this framework, the essential function of having a quantifier interact with the right position in a VP in the right way is done by λ-abstraction. This is a beginnings of a theory of binding which does not invoke logical forms different from the surface forms of sentences. For more development along these lines, see Barker (forthcoming), Hendriks (1993), Jacobson (1999), and Steedman (2000), as well as the earlier Cooper (1983).[14]

The basic idea of the type-shifting approach exemplified here is to think of expressions as *polymorphic*. They inhabit multiple types at once. We think of expressions as entered into the lexicon with their minimal type, which can then be *shifted* by type-shifting rules, like the Geach

[14]Much of this literature works in the framework of categorial grammar, and attempts to develop 'variable-free' accounts of binding phenomena. The background mathematics for this work is combinatory logic, which is a close cousin of the λ-calculus I have employed here. See Hindley and Seldin (1986) for extensive comparisons.

rule. This makes expressions in a way ambiguous. (See Partee (1986), Partee and Rooth (1983), and the extensive discussion in van Benthem (1991).)

Whereas the logical form approach made relatively minor use of type theory, the type-shifting approach leans very heavily on it. Type-shifting approaches do not posit additional levels of linguistic representation, over and above the more or less overt surface structure of the sentence, but they do make use of some powerful mathematics. It is a significant question, both empirical and methodological, which approach is right.

II.5 Scope Relations

The problem of quantifiers in object position barely hints at the complexity of the semantics of quantification. To give a slightly richer example, I shall finally turn to some aspects of quantifier scope relations.

One important feature of quantifiers in natural language is that they can generate scope ambiguities. Recall, as every student of first-order logic learns, *Everyone likes someone* has two first-order representations:

(38) Everyone likes someone.

a. $\forall x \exists y L(x, y)$

b. $\exists y \forall x L(x, y)$

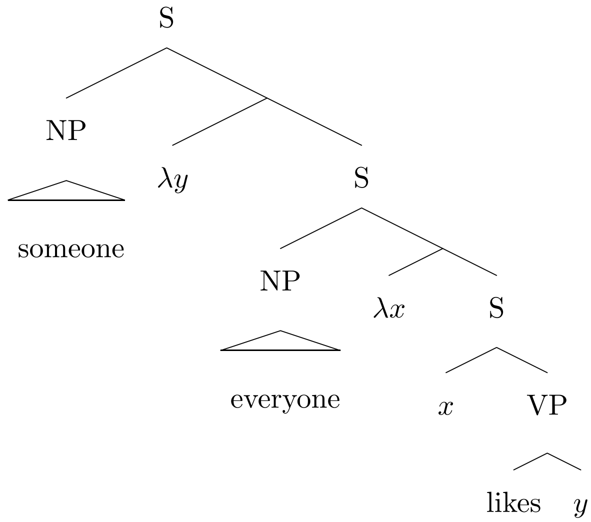
The second is usually called the *inverse scope reading*, as it inverts the surface order of the quantifiers. Another, more complicated inverse scope example is that of inverse linking (May, 1977):

(39) Someone from every city despises it.

May observed that in this sort of case, the inverse scope reading is the only natural one (or perhaps the only one available).

The logical form approach has no fundamental problem with the existence of inverse scope readings. Basically, the logical form approach treats quantifier scope much the way it is treated in first-order logic, modified to employ generalized quantifiers and the account of binding outlined in section II.3. Direct and inverse scope readings are simply the result of different mappings of a sentence to logical forms, corresponding to different orders in which the quantifiers are 'moved' from their *in situ* positions to positions further to the left and higher in the tree. For instance, the inverse scope reading of (38) is given by:

(40)



If we adopt the logical form theory, quantifier scoping is taken care of by the same apparatus which handled quantifiers in object position.[15]

This is an elegant result, and part of a battery of arguments often marshaled to show the existence of a level of logical form (cf. May, 1985). Scope ambiguity is explained by holding that in fact sentences like (38) have *two* distinct logical forms—two distinct linguistic structures. At logical form, scope ambiguity is structural ambiguity.

Type-shifting approaches have to do more work to handle inverse scope. The Geach rule described in section II.4 is not sufficient. One approach to scope via type shifting is to introduce two type-shifting operators which raise the types of the arguments of a transitive verb from e to $((e,t),t)$, allowing the verb to combine with two quantifiers. The *order* in which these operators

[15]The syntax of scope is a rich area of linguistics. The basics can be found in many syntax books. For a recent survey, see Szabolcsi (2001).

Though many logical form theories take the syntax of logical form to determine scope, May (1985, 1989) considers a theory in which it does not completely do so.

are applied determines the scope relations between the quantifiers, much as the order in which the quantifiers are moved does on the logical form approach. Hendriks (1993) shows that these operators can be derived from a single type-shifting principle, but I will leave the rather technical details to him.[16]

Both approaches thus *can* handle inverse scope (though I have suppressed more detail in the type-shifting approach). Which one is right is a substantial question, both methodological and empirical. We face general questions about the apparatus of type shifting and linguistic levels like logical form. We also face empirical issues about which theories can explain the full range of data related to scope and binding. Perhaps the preponderance of current research (at least, research close to syntax) takes place in some version of the logical form approach, but see Jacobson (2002) for a spirited defense of the type-shifting approach.

Though both approaches can handle basic scope inversion cases like (38), the phenomena related to scope in natural language are in fact quite complex. I shall close this section by mentioning a few of the many issues that a full theory of quantifier scope must face.

Though in many cases quantifiers can enter into arbitrary scope relations, there are some well-know situations where they cannot. For instance, quantifiers cannot scope out of relative clauses. Consider (Rodman, 1976):

(41) Guinevere has a bone that is in every corner of the house.

This cannot be given the (more plausible) interpretation in which *every corner of the house* has wide scope. This fact is often cited as evidence in support of logical form theories, which seek to explain it by general syntactic principles, but see Hendriks (1993) for a discussion in type-shifting terms.

Different languages display different scope interactions. Aoun and Li (1993) note sentences which are ambiguous in English but not in Chinese, including the simple:

[16] There are systems which produce inverse scope readings with type-shifting operations more closely related to the Geach rule, like the elegant Lambek calculus with permutation of van Benthem (1991). Unfortunately, this system over-generates scope ambiguities, predicting one in *John loves Paris*, as Hendriks (1993) shows. A more refined theory along van Benthem's lines is given a textbook presentation in Carpenter (1997).

(42) Every man loves a woman.

(The example is credited to Huang.) It is also known that not all quantifiers exhibit the same scope potentials, even in one language. Beghelli and Stowell (1997) and Szabolcsi (1997) note that inverse scope readings do not appear to be available in:

(43) a. Three referees read few abstract.

b. Every man read more than three books.

Aoun and Li (1993) and Beghelli and Stowell (1997) and Szabolcsi (1997) use this data to support their own developments of the logical form approach (cf. Takahashi, 2003). There are also much-discussed difficult issues about the scope of *the* and *a*. See Heim (1991) and van Eijck and Kamp (1997) for surveys.

III What Is a Quantifier?

Can we now say what quantifiers are? Perhaps. Generalized quantifier theory, and the relational theory of determiner denotations which goes with it, offer an answer. The strong hypothesis we considered in section I.7 holds that natural-language quantifiers are logical generalized quantifiers, satisfying the constraints CONS, EXT, and ISOM. These are expressed by determiners, which combine with CNs to build quantified noun phrases. A somewhat weaker hypothesis holds that natural language quantifiers need not be ISOM, but must be CONS and EXT. Section I.7 offered some reasons to prefer the stronger hypothesis.

In a way, this tells us what quantifiers *are* in remarkably specific terms. But the moral of section II is that it does not tell us all that much about how quantifiers *work*. The examples there show us that to understand *quantification* in natural language is to understand more than what quantifiers are; it is also understand significant aspects of semantics, and the ways semantics interact with syntax. Being a quantifier is a property with significant semantic and grammatical implications.

References

Aoun, J. and Li, Y.-H. A. (1993). *Syntax of Scope* (Cambridge: MIT Press).

Bach, E., Jelinek, E., Kratzer, A., and Partee, B. H. (eds.) (1995). *Quantification in Natural Languages* (Dordrecht: Kluwer).

Bach, K. (2000). 'Quantification, Qualification, and Context: A Reply to Stanley and Szabó', *Mind and Language*, 15: 262–283.

Baker, M. C. (2003). *Lexical Categories* (Cambridge: Cambridge University Press).

Barker, C. (1995). *Possessive Descriptions* (Stanford: CSLI Publications).

——— (forthcoming). 'Remarks on Jacobson 1999: Crossover as a Local Constraint', *Linguistics and Philosophy*.

Barwise, J. and Cooper, R. (1981). 'Generalized Quantifiers and Natural Language', *Linguistics and Philosophy*, 4: 159–219.

Beghelli, F. (1994). 'Structured Quantifiers', in M. Kanazawa and C. J. Piñón (eds.), *Dynamics, Polarity, and Quantification* (Stanford: CSLI Publications), 119–145.

Beghelli, F. and Stowell, T. (1997). 'Distributivity and Negation: The Syntax of *Each* and *Every*', in A. Szabolcsi (ed.), *Ways of Scope Taking* (Dordrecht: Kluwer), 71–107.

Bernstein, J. B. (2001). 'The DP Hypothesis: Identifying Clausal Properties in the Nominal Domain', in M. Baltin and C. Collins (eds.), *Handbook of Contemporary Syntactic Theory* (Oxford: Blackwell), 536–561.

Büring, D. (2004). *Binding Theory* (Cambridge: Cambridge University Press).

Cappelen, H. and Lepore, E. (2002). 'Insensitive Quantifiers', in J. Keim Campbell, M. O'Rourke, and D. Shier (eds.), *Meaning and Truth: Investigations in Philosophical Semantics* (New York: Seven Bridges Press), 197–213.

Carpenter, B. (1997). *Type-Logical Semantics* (Cambridge: MIT Press).

Cooper, R. (1983). *Quantification and Syntactic Theory* (Dordrecht: Reidel).

Frege, G. (1879). *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* (Halle: Nebert). References are to the translation as "Begriffsschrift, a Formal Language, Modeled upon that of Arithmetic, for Pure Thought" by S. Bauer-Mengelberg in van Heijenoort (1967).

——— (1891). *Function und Begriff* (Jena: Pohle). References are to the translation as "Function and Concept" by P. Geach in Frege (1984).

——— (1893). *Grundgesetze der Arithmetik*, vol. I (Jena: Pohle). References are to the partial translation as "The Basic Laws of Arithmetic" by M. Furth, University of California Press, Berkeley, 1964.

——— (1984). *Collected Papers on Mathematics, Logic, and Philosophy* (Oxford: Basil Blackwell). Edited by B. McGuinness.

Gamut, L. T. F. (1991). *Logic, Language, and Meaning*, vol. 2 (Chicago: University of Chicago Press). 'Gamut' is a pseudonym for J. van Benthem, J. Groenendijk, D. de Jongh, M. Stokhof, and H. Verkuyl.

Geach, P. (1972). 'A Program for Syntax', in D. Davidson and G. Harman (eds.), *Semantics of Natural Language* (Dordrecht: Reidel), 483–497.

Glanzberg, M. (2004). 'Quantification and Realism', *Philosophy and Phenomenological Research*, 69: 541–572.

Heim, I. (1991). 'Artikel und Definitheit', in A. von Stechow and D. Wunderlich (eds.), *Semantics: An International Handbook of Contemporary Research* (Berlin: de Gruyter), 487–535.

Heim, I. and Kratzer, A. (1998). *Semantics in Generative Grammar* (Oxford: Blackwell).

Hella, L., Luosto, K., and Väänänen, J. (1996). 'The Hierarchy Theorem for Generalized Quantifiers', *Journal of Symbolic Logic*, 61: 802–817.

Hendriks, H. (1993). *Studied Flexibility* (Amsterdam: ILLC Publications).

Herburger, E. (2000). *What Counts* (Cambridge: MIT Press).

Higginbotham, J. (1985). 'On Semantics', *Linguistic Inquiry*, 16: 547–593.

Higginbotham, J. and May, R. (1981). 'Questions, Quantifiers and Crossing', *Linguistics Review*, 1: 41–79.

Hindley, J. R. and Seldin, J. P. (1986). *Introduction to Combinators and λ-Calculus* (Cambridge: Cambridge University Press).

Huang, C.-T. J. (1995). 'Logical Form', in G. Webelhuth (ed.), *Government and Binding Theory and the Minimalist Program* (Oxford: Blackwell), 127–175.

Jacobson, P. (1999). 'Towards a Variable-Free Semantics', *Linguistics and Philosophy*, 22: 117–184.

——— (2002). 'The (Dis)organization of Grammar: 25 Years', *Linguistics and Philosophy*, 25: 601–626.

Keenan, E. L. (1992). 'Beyond the Frege Boundary', *Linguistics and Philosophy*, 15: 199–221.

——— (1993). 'Natural Language, Sortal Reducibility and Generalized Quantifiers', *Journal of Symbolic Logic*, 58: 314–325.

Keenan, E. L. and Moss, L. S. (1984). 'Generalized Quantifiers and the Expressive Power of Natural Language', in J. van Benthem and A. ter Meulen (eds.), *Generalized Quantifiers in Natural Language* (Dordrecht: Foris), 73–124.

Keenan, E. L. and Stavi, J. (1986). 'A Semantic Characterization of Natural Language Determiners', *Linguistics and Philosophy*, 9: 253–326. Versions of this paper were circulated in the early 1980s.

Keenan, E. L. and Westerståhl, D. (1997). 'Generalized Quantifiers in Linguistics and Logic', in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language* (Cambridge: MIT Press), 837–893.

Larson, R. and Segal, G. (1995). *Knowledge of Meaning* (Cambridge: MIT Press).

Lasnik, H. and Stowell, T. (1991). 'Weakest Crossover', *Linguistic Inquiry*, 22: 687–720.

Lepore, E. (1983). 'What Model-Theoretic Semantics Cannot Do', *Synthese*, 54: 167–187.

Lewis, D. (1975). 'Adverbs of Quantification', in E. L. Keenan (ed.), *Formal Semantics of Natural Language* (Cambridge: Cambridge University Press), 3–15.

Lindström, P. (1966). 'First Order Predicate Logic with Generalized Quantifiers', *Theoria*, 32: 186–195.

Longobardi, G. (2001). 'The Structure of DPs: Some Principles, Parameters, and Problems', in M. Baltin and C. Collins (eds.), *Handbook of Contemporary Syntactic Theory* (Oxford: Blackwell), 562–603.

Matthewson, L. (2001). 'Quantification and the Nature of Crosslinguistic Variation', *Natural Language Semantics*, 9: 145–189.

Mautner, F. I. (1946). 'An Extension of Klein's Erlanger Program: Logic as Invariant Theory', *American Journal of Mathematics*, 68: 345–384.

May, R. (1977). *The Grammar of Quantification*, Ph.D. dissertation, MIT.

——— (1985). *Logical Form: Its Structure and Derivation* (Cambridge: MIT Press).

——— (1989). 'Interpreting Logical Form', *Linguistics and Philosophy*, 12: 387–435.

Moltmann, F. (1996). 'Resumptive Quantifiers in Exception Sentences', in M. Kanazawa, C. Piñón, and H. de Swart (eds.), *Quantifiers, Deduction, and Context* (Stanford: CSLI Publications), 139–170.

Montague, R. (1973). 'The Proper Treatment of Quantification in Ordinary English', in J. Hintikka, J. Moravcsik, and P. Suppes (eds.), *Approaches to Natural Language* (Dordrecht: Reidel), 221–242. Reprinted in Montague (1974).

——— (1974). *Formal Philosophy* (New Haven: Yale University Press). Edited by R. Thomason.

Mostowski, A. (1957). 'On a Generalization of Quantifiers', *Fundamenta Mathematicae*, 44: 12–36.

Neale, S. (2000). 'On Being Explicit: Comments on Stanley and Szabó, and on Bach', *Mind and Language*, 15: 284–294.

Partee, B. H. (1986). 'Noun Phrase Interpretation and Type-Shifting Principles', in J. Groenendijk, D. de Jongh, and M. Stokhof (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers* (Dordrecht: Foris), 115–143.

Partee, B. H. and Rooth, M. (1983). 'Generalized Conjunction and Type Ambiguity', in R. Bäuerle, C. Schwarze, and A. von Stechow (eds.), *Meaning, Use and the Interpretation of Language* (Berlin: de Gruyter), 361–393.

Pietroski, P. M. (2002). 'Function and Concatenation', in G. Preyer and G. Peter (eds.), *Logical Form and Language* (Oxford: Oxford University Press), 91–117.

Rescher, N. (1962). 'Plurality-Quantification: Abstract', *Journal of Symbolic Logic*, 27: 373–374.

Rodman, R. (1976). 'Scope Phenomena, 'Movement Transformations,' and Relative Clauses', in B. H. Partee (ed.), *Montague Grammar* (New York: Academic Press), 165–176.

Rooth, M. (1985). *Association with Focus*, Ph.D. dissertation, University of Massachusetts at Amherst.

——— (1996). 'Focus', in S. Lappin (ed.), *Handbook of Contemporary Semantic Theory* (Oxford: Blackwell), 271–297.

Sher, G. (1991). *The Bounds of Logic: A Generalized Viewpoint* (Cambridge: MIT Press).

Stanley, J. and Szabó, Z. G. (2000). 'On Quantifier Domain Restriction', *Mind and Language*, 15: 219–261.

Steedman, M. (2000). *The Syntactic Process* (Cambridge: MIT Press).

Szabolcsi, A. (1997). 'Strategies for Scope Taking', in A. Szabolcsi (ed.), *Ways of Scope Taking* (Dordrecht: Kluwer), 109–154.

——— (2001). 'The Syntax of Scope', in M. Baltin and C. Collins (eds.), *Handbook of Contemporary Syntactic Theory* (Oxford: Blackwell), 607–633.

Takahashi, S. (2003). 'More than Two Quantifiers', *NELS*, 33: 405–424.

Tarski, A. (1986). 'What Are Logical Notions?', *History and Philosophy of Logic*, 7: 143–154. Posthumous publication of a lecture given in 1966, edited by J. Corcoran.

Väänänen, J. A. (1997). 'Unary Quantifiers on Finite Models', *Journal of Logic, Language, and Information*, 6: 275–304.

van Benthem, J. (1983). 'Determiners and Logic', *Linguistics and Philosophy*, 6: 447–478.

——— (1984). 'Questions about Quantifiers', *Journal of Symbolic Logic*, 49: 443–466.

——— (1986). *Essays in Logical Semantics* (Dordrecht: Reidel). Includes revised versions of van Benthem (1983) and van Benthem (1984).

——— (1989). 'Polyadic Quantifiers', *Linguistics and Philosophy*, 12: 437–464.

——— (1991). *Language in Action* (Amsterdam: North-Holland).

van Eijck, J. and Kamp, H. (1997). 'Representing Discourse in Context', in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language* (Cambridge: MIT Press), 179–237.

van Heijenoort, J. (ed.) (1967). *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931* (Cambridge: Harvard University Press).

von Fintel, K. (1994). *Restrictions on Quantifier Domains*, Ph.D. dissertation, University of Massachusetts at Amherst.

Westerståhl, D. (1985*a*). 'Determiners and Context Sets', in J. van Benthem and A. ter Meulen (eds.), *Generalized Quantifiers in Natural Language* (Dordrecht: Foris), 45–71.

——— (1985*b*). 'Logical Constants in Quantifier Languages', *Linguistics and Philosophy*, 8: 387–413.

——— (1989). 'Quantifiers in Formal and Natural Languages', in D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, vol. IV (Dordrecht: Kluwer), 1–131.

——— (1994). 'Iterated Quantifiers', in M. Kanazawa and C. J. Piñón (eds.), *Dynamics, Polarity, and Quantification* (Stanford: CSLI Publications), 173–209.

Williamson, T. (2004). 'Everything', *Philosophical Perspectives*, 17: 415–465.